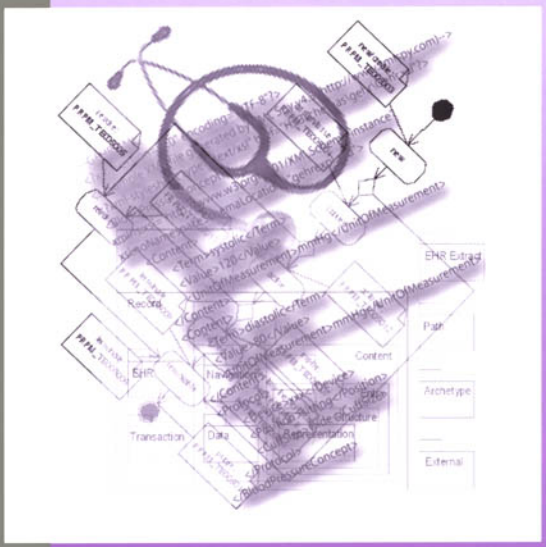
TECHNOLOGY AND INFORMATICS 89

Analysis, Design and Implementation of Secure and Interoperable Distributed Health Information Systems



Bernd Blobel

IOS Press
Ohmsha

ANALYSIS, DESIGN AND IMPLEMENTATION OF SECURE AND INTEROPERABLE DISTRIBUTED HEALTH INFORMATION SYSTEMS

Studies in Health Technology and Informatics

Editors

Volume 89

Earlier published in this series

Vol. 61. R.A. Mortensen (Ed.), ICNP® and Telematic Applications for Nurses in Europe
Vol. 62. J.D. Westwood, H.M. Hoffman, R.A. Robb and D. Stredney (Eds.), Medicine Meets Virtual Reality
Vol. 63. R. Rogers and J. Reardon, Recommendations for International Action
Vol. 64. M. Nerlich and R. Kretschmer (Eds.), The Impact of Telemedicine on Health Care Management
Vol. 65. J. Mantas and A. Hasman (Eds.), Textbook in Health Informatics
Vol. 66. The ISHTAR Consortium (Eds.), Implementing Secure Healthcare Telematics Applications in Europe
Vol. 67. J. Oates and H. Bjerregaard Jensen (Eds.), Building Regional Health Care Networks in Europe
Vol. 68. P. Kokol, B. Zupan, J. Stare, M. Premik and R. Engelbrecht (Eds.), Medical Informatics Europe '99
Vol. 69. F.-A. Allaert, B. Blobel, C.P. Louwerse and E.B. Barber (Eds.), Security Standards for Healthcare Information Systems
Vol. 70. J.D. Westwood, H.M. Hoffman, G.T. Mogel, R.A. Robb and D. Stredney (Eds.), Medicine Meets Virtual Reality 2000
Vol. 71. J.T. Ottesen and M. Danielsen (Eds.), Mathematical Modelling in Medicine
Vol. 72. I. Iakovidis, S. Maglavera and A. Trakatellis (Eds.), User Acceptance of Health Telematics Applications
Vol. 73. W. Sermeus, N. Kearney, J. Kinnunen, L. Goossens and M. Miller (Eds.), WISECARE
Vol. 74. O. Rienhoff, C. Laske, P. van Eecke, P. Wenzlaff and U. Piccolo (Eds.), A Legal Framework for Security in European Health Care Telematics
Vol. 75. G.O. Klein (Ed.), Case Studies of Security Problems and their Solutions
Vol. 76. E.A. Balas, S.A. Boren and G.D. Brown (Eds.), Information Technology Strategies from the United States and the European Union
Vol. 77. A. Hasman, B. Blobel, J. Dudeck, R. Engelbrecht, G. Gell and H.-U. Prokosch (Eds.), Medical Infobahn for Europe
Vol. 78. T. Paiva and T. Penzel (Eds.), European Neurological Network
Vol. 79. A. Marsh, L. Grandinetti and T. Kauranne (Eds.), Advanced Infrastructures for Future Healthcare
Vol. 80. R.G. Bushko, Future of Health Technology
Vol. 81. J.D. Westwood, H.M. Hoffman, G.T. Mogel, D. Stredney and R.A. Robb (Eds.), Medicine Meets Virtual Reality 2001
Vol. 82. Z. Kolitsi (Ed.), Towards a European Framework for Education and Training in Medical Physics and Biomedical Engineering
Vol. 83. B. Heller, M. Löffler, M. Musen and M. Stefanelli (Eds.), Computer-Based Support for Clinical Guidelines and Protocols
Vol. 84. V.L. Patel, R. Rogers and R. Haux (Eds.), MEDINFO 2001
Vol. 85. J.D. Westwood, H.M. Miller Hoffman, R.A. Robb and D. Stredney (Eds.), Medicine Meets Virtual Reality 02/10
Vol. 86. F.H. Roger-France, I. Mertens, M.-C. Closon and J. Hofdijk (Eds.), Case Mix: Global Views, Local Actions
Vol. 87. F. Mennerat (Ed.), Electronic Health Records and Communication for Better Health Care
Vol. 88. A. Tanguy and B. Peuchot (Eds.), Research into Spinal Deformities 3

ISSN: 0926-9630

Analysis, Design and Implementation of Secure and Interoperable Distributed Health Information Systems

Bernd Blobel

Institute of Biometry and Medical Informatics,
Otto-von-Guericke University, Magdeburg, Germany

IOS Press
Ohmsha

Amsterdam • Berlin • Oxford • Tokyo • Washington, DC

ISBN 1 58603 277 1 (IOS Press)
ISBN 4 274 90533 0 C3047 (Ohmsha)
Library of Congress Control Number: 2002109992

Publisher
IOS Press
Nieuwe Hemweg 6B
1013 BG Amsterdam
The Netherlands
fax: +31 20 620 3419
e-mail: order@iospress.nl

Distributor in the UK and Ireland
IOS Press/Lavis Marketing
73 Lime Walk
Headington
Oxford OX3 7AD
England
fax: +44 1865 75 0079

Distributor in the USA and Canada
IOS Press, Inc.
5795-G Burke Centre Parkway
Burke, VA 22015
USA
fax: +1 703 323 3668
e-mail: iosbooks@iospress.com

Distributor in Germany, Austria and Switzerland
IOS Press/LSL.de
Gerichtsweg 28
D-04103 Leipzig
Germany
fax: +49 341 995 4255

Distributor in Japan
Ohmsha, Ltd.
3-1 Kanda Nishiki-cho
Chiyoda-ku, Tokyo 101-8460
Japan
fax: +81 3 3233 2426

PRINTED IN THE NETHERLANDS

Dedicated

to my wife, who facilitated this work by never ending love and patience,

to my commendable colleague Professor Dudeck
and to my co-workers for kind co-operation.

This page intentionally left blank

Foreword

Joachim Dudeck
University of Giessen, Germany

Any type of E-Business including E-Health is growing up changing our world. New generations of Internet-based health information systems intend to meet the challenge of new economy concerning requirements for high efficiency, efficacy, and high quality of care and welfare. Distribution, communication, interoperability, internationalisation, and even globalisation became valid paradigms. Architects and implementers of health information systems work to respond to the requirements and answer countless questions. For making products, solutions became topics of standards developing organisations. The world changed so quickly that it became impossible being updated enough to contribute to the emerging development or at least to use the achievements. As development and knowledge explode, also papers and books became innumerable, describing problems and the state of the art. Many technical books go into depth. Sometimes, papers are talking about challenges and principles without offering appropriate solutions.

This book offers an extended walk trough the domain of health information systems covering all the different aspects from architecture to security. Nevertheless, the broadness of concern is accompanied by a challenging depth in many parts of the presentation. The book addresses medical informaticians, computer scientists, software developers, system architects, system administrators, decision makers, and also users. Each group will find some chapters or sections answering special questions of that audience. To facilitate the flexible use of the book, it has been written in chained chapters, nevertheless offering the opportunity of being interested in, and reading, one chapter only. Therefore, the chapters are completed from introduction up to summary.

Beside practical reports, three paradigms draw the thread through the book: Component-orientation, meta-modelling (UML), and the XML markup language. Following these paradigms, openness, flexibility, scalability, portability, and interoperability of systems should be provided.

Being engaged in many standardisation bodies, the author embraces the healthcare systems turn to the shared care paradigm, analyses the most important architectural approaches for health information systems, introduces the current work on new generation of electronic health records, and presents requirements, standards, and solutions for advanced security services. Thereby, the author covers established systems and upcoming developments for future systems of the next decade as well.

The book benefits from the author's active participation in European projects leading in the field. Being responsible for work items and task forces, he is influencing the direction medical informatics goes into. This is especially true in the domain of health information systems and Electronic health record architecture. But also in the domain of privacy and security for health information systems the author is well accepted internationally. Due to his chairmanship in the European Federation of Medical Informatics (EFMI) Working Groups "Electronic Health Record" and "Security" as well as in HL7 and CORBA Technical Committees and Working Groups, the author presents not only his knowledge but also the knowledge of co-operating international experts.

The theoretical considerations are mostly combined with practical specifications and implementations. This has been done using an acknowledged demonstrator piloting new solutions for health information systems and health networks: the Magdeburg regional Clinical

Cancer Registry and its ONCONET. Sometimes, some more examples for the practical deployment of the results and proposed solutions would be desirable. On the other hand however, the extension of the book sets some limitations.

The inclusion of many important contributions from others has been made. Thereby, many references have to be ignored in such a broad field avoiding a reference list longer than the book itself.

I wish the reader an enjoyable lecture.

Giessen, April 2002

Prof. Dr. Joachim W. Dudeck

Acknowledgement

The author is indebted to thank the European Commission and the Ministry of Education and Science of the German Federal State Saxony-Anhalt for their funding as well as all partners of the projects mentioned for their kind co-operation. In that context, especially the support by Dr Gottfried Dietzel (Bonn, Germany) but also the friendship of, and the promotion by, Prof Joachim Dudeck (Giessen, Germany), Dr Rolf Engelbrecht (Munich, Germany) and Dr Barry Barber (Birmingham, UK) for having introduced the author to international projects of research, development and standardisation need to be mentioned. Without Prof Dudeck's help, the author's engagement in HL7 and OMG would have never happened.

Interactions, discussion rounds, contributions and co-operations in the author's own research group provided the fruitful environment needed. Therefore, the author gives his thanks to Peter Pharow, Volker Spiegel, Kjeld Engel, Rolf Krohn, Dr Martin Holena, but also the other co-workers of the Magdeburg Medical Informatics Department.

Regarding internal support, the author is indebted to thank Peter Pharow for his contribution for polishing the book's language, but also for his help in administering the projects. Volker Spiegel provided many of the implementation specifications presented.

Additionally, both the German HL7 User Group, especially Frank Oemig (Muehlheim) and Dr Kai Heitmann (Cologne), and HL7 USA as well as some International Affiliates, but also the CORBA community gave essential support to the author's effort enabling the comprehensive view and the educational examples presented in the book. Another input arose from colleagues of CEN TC 251 and ISO TC 215, but especially by the EHR groups of CEN and GEHR. In the latter context, especially Thomas Beale (Mooloolah, Queensland, Australia), but also Ken Rubin (Washington D.C., USA), Dr Dipak Kalra and David Lloyd (both London, UK) formed important discussion rounds.

Regarding influencing projects, especially Prof George Stassinopoulos (Athens, Greece) as well as Petra Hoepner (Berlin, Germany) and their teams, both related to the HARP project, but also Dr Birgit Baum-Waidner (Zurich, Switzerland) and Gerrit Bleumer (Hildesheim, Germany) from the ISHTAR project promoted the book's outcome.

The decision support and clinical guideline related stuff got support from Dr Yasser alSafadi (Briarcliff Manor, NY, USA) I'd like to thank for.

Furthermore, I give my gratitude to the CEO of the University Hospital Magdeburg, Mrs. Veronika Raetzel, acknowledging her steady support of our national and international engagement as well as to the former Director of the Institute of Biometry and Medical Informatics, Prof Jürgen Läuter, for the freedom given.

Finally, I'd like to thank my wife for her love and patience enabling time, effort, and engagement to perform all the research and writing.

This list of acknowledged persons must be incomplete, so the author would apologise for everybody who should be mentioned but isn't, however.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

1.1 The Health Systems' Challenge

In all developed countries, the basic conditions of health and welfare are changing caused by social, economic, technological, political and environmental drivers [Garets, 2001].

In the social context,

- the demographic development,
- the citizens' expectation on health, and
- the growing social differentiation of the society;

in the economic context,

- structural deficiencies, accompanied by uncontrolled medical costs;

in the technological context,

- the progress of medicine and bio-medical technology including
- evidence-based medicine,
- the evolving computerised patient record,
- E-health,
- concerted actions on taxonomies for efficiency and quality of services, and
- establishment, improvement and internationalisation of standards;

in the environmental context,

- the globalisation of diseases;

in the political context,

- globalisation,
- improved legislative oversight on healthcare organisations

have to be mentioned.

The demands for health services - at least in Germany and in some other countries - are growing with decreasing social security budgets due to the increasing rate of unemployment at the same time as well as economic problems in general. Under such constraints and challenges, the societies are modifying their healthcare system structure to [Blobel, 1996b; Blobel 1996c; BMG, 1995]:

- decentralisation and specialisation,
- a shared caring concept (see next paragraph),
- extended communication and co-operation between the care providers, but also between providers and funding organisations, e.g. insurance companies, and/or other institutions directly or indirectly involved in healthcare,
- at least a minimum of competitiveness on the basis of corresponding transparency of outcome and flexibility in compliance with ethical principles laid down in fundamentals of the social market economy.

These processes are accompanied by a rapidly extending and improving electronic communication. In Chapter 2.3, such communication is considered in a very generic way realising that information as the formalisation of existing or thought "reality" is always bound to its communication and interpretation. In so far, interoperability means always communication, too.

According to Gartner's vision for healthcare, accountability for payment and compliance with standards and structured data are the main axes within a multi-axial taxonomy for deciding where the health system is moving to. The decision for (social) market economy, standards and structured data are the crucial points for overcoming the challenges mentioned above [Garets, 2001].

By that way, the health system should meet the challenge for high quality and efficient health provision. Following, the consideration is mostly restricted on the domain of healthcare.

1.2 Definition of "Shared Care"

Corresponding to [Ellsässer and Köhler, 1993], *shared care* can be defined as

"a continuous and co-ordinated activity of

- different persons in
- different institutions under
- employment of different methods at
- different times

in order to be able to help patients optimally with respect to their

- medical,
- psychological and
- social being".

The basis of *shared care* is a common view on the common object or – better – subject, the patient. This common view is provided by a common and structured documentation, which has to be comprehensive and consistent. Keeping this in mind, *shared care* is originally based on ideas of the well-known and valiant pioneer of Medical Informatics and Medical Documentation Systems, Dr L.L. Weed, who has inspired the thinking of communication and co-operation in health controlled by the medical problem [Weed, 1970; Weed, 1978]. Nowadays, communication and co-operation in health are performed within departments and organisation, but also crossing regional and in the future even national boundaries. The extension of communication and collaboration mainly depends on the health system's structure of the different organisations, regions, and countries. In that context, essential differences may be found between Germany on the one hand and some other European countries and the United States on the other hand.

The current German health system consists of a federal decentralised structure of *healthcare establishments* (HCE) in self-administration of the bodies involved, which is ruled by a rather strict legislation. Such a structure demonstrates the advantages of a huge variety of services offered and certain flexibility, but also problems in harmonisation of objectives, standards, and solutions as well as in the agreement about common policies overcoming challenges.

Centralising their health systems, countries as the United Kingdom have better fundamentals for introducing *shared care* and promoting communication and co-operation between care providers. Responding to the challenge of *shared care* information systems[1], this advantage is obvious.

Another example for systems promoting *shared care* and supporting *shared care* information systems is the US architecture of *Health Maintenance Organisations* (HMO), that pro-

[1] The relationship between the shared care paradigm and corresponding information systems will be discussed in Chapter 2.

vide *managed care* bringing together, and controlling, the services of hundreds of providers (GPs, hospitals, etc.).

Extended discussions on strengths and weaknesses of centralisation and regulation on the one hand versus decentralisation and free market paradigms on the other are out of scope of the author's intentions for this book.

A discussion of the different architectures of health systems in some countries especially related to its reflection to health information systems (HIS) can be found in [Blobel, 2001]. Continuous and actual information is provided by the *World Market Series* supported by the *World Medial Association* (WMA), the *International Medical Informatics Association* (IMIA), and other main players in the field.

1.3 Objectives of the Book

The challenge for high quality and efficient healthcare within the *shared care* paradigm with its development towards *managed care* can only be met by supporting the health system with adequate information systems. This requirement implies that also such health information systems have to correspond to the *shared care* paradigm, i.e., they have, in particular, to be communicating and interoperable, too. Interoperability faces technical protocols, but also functional protocols, semantic protocols, and functional reference models. Chapter 2 presents a more detailed view on objectives and requirements on distributed, communicating, and co-operating health information systems.

A lot of work has been undertaken to design, specify, and implement such information systems. In that context, different architectural approaches have been developed, sometimes competing and sometime completing each other. The solution needed is impossible to be provided without a careful analysis of requirements including the system users at all levels of the HCE and supporting organisations. In that context, a *shared care* view is indisputable. The different views starting with the real world reflection of structures and services, domain description, system analysis, system design, implementation, and maintenance must be enabled.

Different approaches have been developed and published, describing the architecture based on *entity relationship* (ER) diagrams, as a layered scheme, as a system of objects or as an assembly of components. This monograph makes use of all the different approaches depending of the objectives of the description. Examples for the use of layered approaches in general describing *hospital information and communication systems* (HICS) are given, e.g., in [Winter and Haux, 1995], where a conceptual, a logical, and a physical layer has been introduced. Another example dealing with architecture management in large enterprises can be found in [Hermanns et al., 1999], who defines a four tiers architecture considering the process architecture, the domain terminology architecture, the application architecture, and at the lowest level the system architecture. A third approach based on the layered model architecture has recently been published by Van de Velde [Velde, 2000], however not in a fully consistent and updated way towards globally accepted developments. Another approach developed in the Magdeburg Medical Informatics Department and meanswhile more or less established by different working groups is a component-oriented architecture based on a multi-model design.

Because personal medical information is recorded, stored, processed in, and communicated between, health information systems, such systems require advanced services to guarantee data protection and security to the patient, the Health Professionals, and the organisations involved. The term security represents more than other services legal, ethical, societal, organisational, functional, and also technological issues. Therefore, the book, focussing on the analysis and design of secure health information systems, must look for a comprehen-

sive approach for analysis, design, implementation, and maintenance considering the different aspects mentioned above without gaps and breaches usually occurring.

1.4 This Book's Scope

The book concerns modern architectures for *shared care* information systems. In that context, the technical, architectural, methodological, and functional framework has to be established. An important issue are the essentials to provide newly developed or legacy health information systems for security. Because security-related analysis, design, and implementation of health information systems require co-operation, education, and training of all different user groups involved including the management of HCE, the users, the system administrator, the maintenance staff as well as developers and implementers, these different groups must be enabled for proper contribution. Developing appropriate models with different levels of granularity and abstraction, the book provides methods and advice to support that challenge facilitating the different view and knowledge of the parties by a coherent modelling approach. Reflecting concrete information systems' solutions on the principles stated, the practicability of the approaches is shown promoting analysis and design of the systems by transferring them into harmonised and therefore comparable meta-models. Tracing back requirements and solutions to basic components and functions, the security enhancement of health information systems is eased.

For satifying both practical users and future-oriented developers, the book deals with both available solutions and future concepts broadly introduced after years presumably.

1.5 How to Read the Book

The book covers a wide range of aspects health information systems have to deal with. This concerns the technical, architectural, methodological, and functional framework of such systems, especially mentioning the security aspects often ignore but socially, ethically, and legally pronounced.

The book has been written in chapters in such a way, that the reader can select aspects according to his qualification and interest. For that reason, the chapters are loosely coupled enabling to start and to finish just as the reader likes. This way of writing implicates some repetitions as inevitable.

The book starts with a discussion of status of, and trends for, health information systems (Chapter 2). Exemplified by a modern HICS, communication and co-operation needed to meet the challenges for *shared care* information systems are investigated. Two types of interoperability may be distinguished: *interfacing* and *integration*. Because *interfacing* reflects simple exchange of information not combining functionality by the principals[2] interoperating, for *shared care* information systems providing the advanced services needed the communication type *integration* must be required finally.

Chapter 3 compares the dominant architectural approaches currently used, their ongoing practical enhancements and related health information systems. In that context, the Health Industry Level 7 Communication Standard (HL7), the Object Request Broker Architecture approach for the healthcare domain (CORBAmed) promoted by the Object Management Group (OMG) as well as the European open, interoperable Distributed Healthcare Environment (DHE) based on the Health Information System Architecture (HISA) CEN prestandard are presented and evaluated in detail. This chapter refers to the specifications practically implemented, but also to new and emerging developments of those architectures.

---

[2] Principals might be users, applications, components, objects on the business, logical or physical level (see also Chapter 2)

Latest developments and improvements including fundamental paradigm changes are discussed in Chapter 5.

As the result of the authors recent six year's work, Chapter 4 analyses the component paradigm as an alternative to provide a comprehensive tool-set for analysis and design of security enhanced *shared care* information systems enabling the views of all different user groups involved.

First of all, other currently upcoming architectural solutions are mentioned. Afterwards, components and their properties are defined. Because the Unified Modeling Language (UML) methodology is being used for analysis and further development of the component approach responding to our challenges, an overview is given about this tool-set. Thereby, the domain concept widely applied in Chapter 5 is referred using the component terminology. Based on theories of recursive functions and abstract automatons, the common process model for component system transitions has been investigated. This transition enabling the different granularity, abstraction and therefore view needed, has been derived as keeping the essential properties.

Chapter 5 presents the current activities and resulting models related to Electronic Healthcare Records (EHCR). In that context, the European standards on EHCR and the Australian Good Electronic Health Record (GEHR) running towards the openEHR initiative, but also the US Governmental Computerised Patient Record (GCPR) project as well as HL7's initiatives on Reference Information Model (RIM) and on Clinical Document Architecture (CDA) are discussed in more detail. Based on the generic component model in Chapter 4, the harmonised EHR architecture approach faced by an international team including the author is offered in some detail. Summarising, the chapter shows the future direction in system design, implemenation, use, and maintenace.

Based on the results achieved in Chapter 2, 3, 4, and 5, Chapter 6 presents a set of models to support the systematic analysis and design of security requirements and solutions in health information systems. Regarding the defined, well-distinguishable concepts of application security and communication security, a layered scheme of security concepts, services, mechanisms, algorithms, and data has been developed facilitating the navigation through security requirements and solutions. For implementation, also protocols and products have to be considered. To describe structure, functions, and behaviour of systems, intended or real, the UML methodology is used. Defining domains which reflect the environment, policy and technology, information systems can be kept manageable. Such domains offer help for the establishment and bridging of policies as well as for validation of systems providing additional middleware services or communication only.

Starting with medical scenarios, 6 use case types could be derived enabling the description of any health information system's behaviour. Considering both application and communication security issues, 9 use case types have been specified allowing the description of security requirements and solutions in any system configuration. Specifying organisational and functional roles for Health Professionals (HP), model state transitions of application security services are provided switching to the higher granularity of access control models. To develop the UML diagrams, the Rational Rose toolkit has been deployed. Based on the same methodology, also the security framework for health information systems including authorisation models is derived. The generic access control model approach is elucidated by commonly used access control models referred to in Chapter 7.

Beside the refinement of some application security issues, Chapter 7 deals with further practical aspect of assessment and use of the results achieved in *shared care* information systems. In that context, some legal aspects for development and deployment of such systems are shortly summarised, specific communication services are discussed.

Reflecting the principles, services and mechanisms, the security framework of middleware approaches discussed in Chapter 3 is presented in Chapter 8, also considering the author's own investigations and specifications in the context of CORBAmed security initiatives. As a summary of future directions for enhanced security solutions for component-based, distributed health information systems on the open Internet, the HARP Cross Security Platform (HCSP) elaborated within the European HARP project is shortly presented.

Based on the modelling results elaborated, Chapters 9, 10, and 11 provide specifications and implementation examples dealing with partial and comprehensive solutions for communication and application security services, also considering the security infrastructure and special information system environments.

Chapter 9 discusses principles and solutions of the security infrastructure needed for the design and implementation of security enhanced health information systems. The TrustHealth project results are reflected especially discussing the Magdeburg initiatives within the project framework. Health Professional Cards (HPC) and Trusted Third Party (TTP) services are described in general and for the specific local environment securing an Electronic Health Care Record (EHCR) system in oncology as well as Internet TTP structures. In the context of infrastructural security services, also principles of anonymisation and pseudonymisation are discussed in this chapter

In Chapter 10, the deployment of investigations, developments and modelling to specify and implement security enhanced Electronic Data Interchange (EDI) communication is demonstrated.

Chapter 11 introduces security aspects of the class of chip card based health information systems demonstrated by the DIABCARD example.

Chapter 12 discusses a future-proof approach of a component-based open, portable, secure, and interoperable health information system architecture which is based on the European HARP project achievements. This approach implements the paradigms established in Chapter 4 and improves the advanced EHR architecture presented in Chapter 5. Furthermore, some aspects of decision support, clinical practice guidelines and their real implementation are mirrored using the generic component model approach demonstrating its generic and comprehensive character.

Chapter 13 roughly informs about European projects the Magdeburg Medical Informatics Department was or is involved in. This paper has been stimulated essentially by the work within these projects funded by the European Commission. As projects of the European 4th framework programmes "Telematics Applications Programme" (TAP) and "Information Society Initiative for Standards" (ISIS), DIABCARD, HANSA, ISHTAR, TrustHealth, EUROMED-ETS, and MEDSEC are considered. As projects of the European 5th framework programme "EU Information Society Technologies Programme" (IST), HARP and RESHEN are shortly presented. Furthermore, the German partners in co-operation within the project frameworks are mentioned.

Chapter 14 provides some conclusions and recommendations for using the results presented to improve the current situation of health information systems' security.

The Annexes concluding the monograph present some extracts of documents elaborated in the Magdeburg Medical Informatics Department.

In Annex A and B, normative references and abbreviations directly or indirectly used are mentioned. Annex C describes the regional Clinical Cancer Registry Magdeburg/Saxony-Anhalt as the pilot environment many of the research and development results have been practically implemented for. This annex also illustrates the implementation details for security enhanced EDI communication demonstrated for the HL7 example transferred via secure FTP. Annex D provides an overview on the implemented solution for secure Patient Data Card applications shown for the DIABCARD example. Special attention is given to

the general security of the doctor's workstation used. Both annexes concern technical details of practically implemented solutions a technician or informatician might be interested in. They may be read independently from the book's regular chapters. Therefore, Annexes C and D are handled in a closed way with own references and with some repetitions of terms or figures presented in the book.

Annex E summarises some important legal fundamentals for security enhanced applications in health at European scale explored within the European TrustHealth project.

All the practical implementations deploy the European security infrastructure based on TrustHealth Health Professional Cards and Trusted Third Party services.

2 Paradigm Changes in Health Information Systems

2.1 Healthcare, Health Information Systems and Communication

The organisational and functional structure of any healthcare establishment independent of its level of complexity ranging from single workstations, GP office systems up to hospitals of HMOs consists of different components (be aware of not being confused with the term *component* used in Chapter 4) related to both direct and indirect patient care. In principle such systems can be separated into patient administration, patient care, medical services, and administration logistics. Additional functions could be related to research, education, and training. In some of the HCE the general functions are performed by one or few persons (GP practice) or by a single department without any communication or with internal communication only. In other cases the processes of healthcare are performed by labour-sharing with the need for a more or less extended communication between the partners in co-operation. Figure 2.1 presents one example for a general high level model of the *hospital information and communication system* (HICS) architecture [Velde, 1992]. In general, ward units and medical departments can also be named by patient care and medical services respectively. According to the *shared care* paradigm, the components presented might belong to different HCE (organisations, organisational units, or even single workstations). The communication may be based on paper, might be paperless, i.e. electronic, or a combination of both which will be the practice for a while.

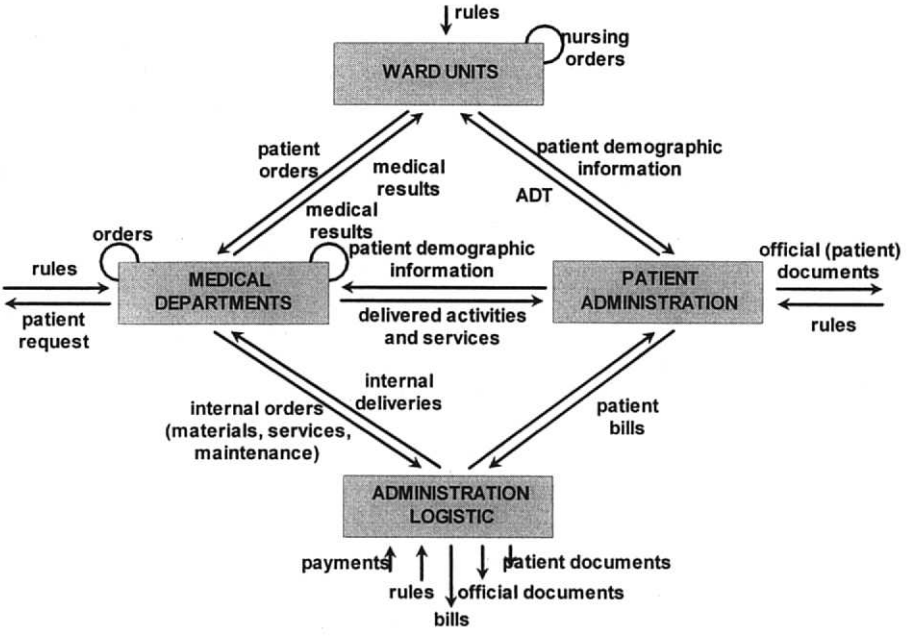


Figure 2.1: General Model of the HICS Architecture (after [Velde, 1992])

2.2 Health Information Systems

The complexity of healthcare processes, the amount of data, and the improvement of technology in health institutions, especially in information technology, lead to an increasing use of information technology to support the care-related processes. This is done in smaller institutions or for early implementations by closed (centralised) solutions. In larger organisations as well as in later implementations, we can find departmental applications and information systems integrated into the care-related processes corresponding to the advanced professional requirements on departmental applications and information systems. The instantiation of the general model of HICS architectures reflecting the Magdeburg University Hospital HICS concept is shown in Figure 2.2 [Blobel, 1996b]. Some logically centralised services must be provided, needed to integrate the decentralised applications or requested as basic service for any component both for consistency and integrity reasons. In that context, a common ID management but also other services managing common functionality are essential. As shown in more details in Chapter 5, these centralised services correspond to a comprehensive EHR. From the three tiers layer architecture's point of view, this integrative part represents the persistent data storage as well as at least partially the common business logic behind an integrated *shared care* environment. Furthermore, a communication infrastructure in hardware, software and organisation has to be established.

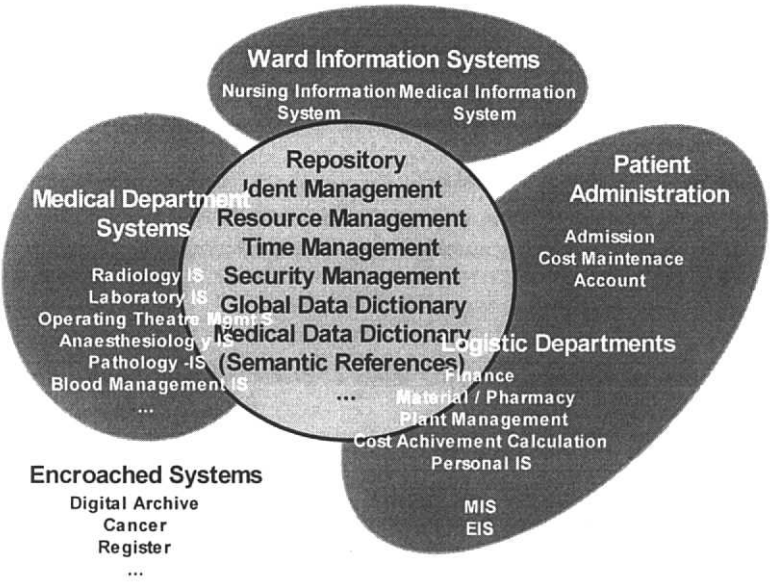


Figure 2.2: Functions and Information Systems in HCE

Generally speaking, the health systems' paradigm shifts from organisation-centred to patient-centred and from facility-oriented to service-oriented business approaches.

Decentralised solutions are accompanied by increased intersystem communications and related security threats and risks. On the other hand, such applications are commonly characterised by more complex functionality, significantly increased scalability, interoperability, and portability.

Summarising, health information systems consist of an infrastructure enabling security, identification, data repository, and management functions, basic services such as record and

retrieval services based on a comprehensive knowledge base (vocabulary, etc.), process and workflow control and management. Domain-specific functions are facilitated by domain-specific knowledge models and procedures based on the aforementioned reference services. This view on health information systems is discussed in more detail in Chapter 5.

2.3 E-Health

With the advent of networks and especially the Internet, a new paradigm has been developed: e-Health. As part of the global Information Society programme as well as the e-Europe Initiative, electronic health aims comprehensive communication and co-operation between entities based on extended networks independent of their physical architecture being wired or wireless. Due to the enhancement of citizens' mobility, mobile computing gets more and more dominant. Distributed collaboration will enable new scenarios and new results. The basic paradigm is resource sharing and co-ordinated problem solving in dynamic, multi-institutional, virtual organisations. The modular structure of the e-Health environment serves enhanced application functionality supported by appropriate visualisation, data mining, knowledge-based integration and advanced queries, grid storage and high-speed networking.

As a crucial part of the e-Health architecture, the Internet will be developed towards Internet 2 and beyond. Commonality is found at meta-model, interface and service architecture level. Social engineering will be at least as important as software engineering. Data will develop as the fundamental driver of systems.

The content and extent of communication as well as the communication infrastructure determine the new threats and risks, define the need for protection, and facilitate new measures for data security.

2.4 Communication in Healthcare

Especially reflecting security issues, communication in healthcare can be characterised by communication content, communication partners, communication infrastructure, and communication services.

2.4.1 Communication Content

In healthcare environment, the following kinds of *communication content* can be distinguished:

- patient-related medical communication (e.g. information about diagnosis and therapy),
- patient-related non-medical communication (e.g. patient account and bill),
- non-patient-related medical communication (e.g. epidemiological results),
- non-medical communication (e.g. materials),
- communication of content with different sensitivity and threats and risks (open information, internal information, sensitive information, secret information) and therefore different needs of protection related to the patients', Health Professionals' and enterprise's point of view.

2.4.2 Communication Partners

Regarding the *communication partners*, the required communication can take place

- within a single healthcare unit,
- between different medical units of a single healthcare institution/organisation,

- between different medical and non-medical units of a single healthcare institution/organisation,
- between different healthcare institutions/organisations,
- between healthcare institution/organisation and non-caring partners in healthcare (e.g. pharmacies, laboratories, other services, institutes, ...),
- between healthcare institution/organisation and non-medical partners in healthcare (insurance companies, funding organisations, ministries, ...),
- between healthcare institution/organisation and institution/organisation from outside of healthcare (e.g. libraries, suppliers, information providers),
- at different legal bases, restrictions in contents and rights to communicate information (see Chapter 7).

2.4.3 Communication Infrastructure

With respect to the *communication infrastructure*, the communication can be transmitted (see Chapter 10)

- via point-to-point connection or
- via networks.

These communications can be performed either by circuit switching or by packet switching. In addition, the communication infrastructure may be private or partly closed (rented line, corporate network) or public.

2.4.4 Communication Services

Regarding *communication services*, simple and advanced services can be distinguished (see Chapter 10). The latter can be general services or application(-domain)-related services. Simple communication services are, e.g.,

- file transfer protocol (ftp),
- remote access (rlogin, RPC, telnet, ...).

As examples for advanced communication services,

- mailing,
- WWW, gopher, WAIS,
- rigid or flexible event-driven messaging (EDI[3] as HL7 (Chapter 3), UN-EDIFACT[4], XML[5], xDT[6]),
- CORBA (Chapter 3), DHE (Chapter 3), ActiveX, JavaBeans and other approaches (Chapter 4) providing interoperability.

The first two are common services; the others are related to application domains. The services could be provided by centralised or decentralised architectures, mentioning architecture-related issues.

In summary, in a "combinatorial way", different contents, communication partners, infrastructure, and services present different communication conditions, following different communication threats as well as risks and request for adequate countermeasures. There are also other threats, regarded for example in SEISMED [SEISMED, 1996; Barber et al.,

---

3 Electronic Data Interchange

4 EDI for Administration, Commerce and Transport

5 Extensible Markup Language, a subset of SGML (Standard Generalized Markup Language)

6 XDT is a XML-like Message Exchange Format standardised in Germany for communication between doctor's office systems as well as between them and the systems of other healthcare providers

1996; Patel and Kantzavelou, 1995]. Nevertheless, the scope of this paper is especially focused on new issues related to communication.

The above mentioned aspects of communication conditions, related threats and risks concern all domains, as commerce, finance, or transport. Nevertheless, healthcare is related to specific circumstances of social and mental behaviour, ethics and others, implying a healthcare-specific consideration of security issues [Brannigan and Beier, 1995; Kluge, 1995a,b].

The communication of patient-related information is accompanied by threats to their integrity and confidentiality but personnel's information or enterprise data can also be sensitive and should be protected in an appropriate manner.

Although they were often designed as closed systems, the information systems in healthcare are meeting open systems regarding the security architecture. The next paragraphs will deal with special conditions and technologies, threats, risks and countermeasures.

2.5 Patient Care and Health Networks

As already mentioned in the introductionary chapter, the only response to the challenge for health systems and its information infrastructure is the *shared care* paradigm. The information-technological support of *shared care* consists, e.g., of specialised components communicating and co-operating as well as an infrastructure facilitating this interoperability. Beside some basic services at the lowest layer of the ISO model for open systems interconnection (operating systems, simple transport protocols, etc.), a set of services located at the medium layers to support the interoperability, and a set of services at the application layer enabling the application functionality intended must be established. All components involved at any level of granularity (see also Chapter 4) can be centralised or decentralised. The management required is related to the same layers: system management at the ground, logical management at the medium layers, and functional management on top. The infrastructure mentioned is implemented in networks defining the term of distribution very generically. The services may be distributed at one single system (different logical units, address space), within compound systems, or around the world. Also *shared care* organisation may be established in such varying realisations ranging from doctors' joint practice and physicians houses up to regional, national, or even international health networks. The management statement is also valid in the latter domain. In both the technical and the organisational view, interoperability is not the same in any case. Therefore, the view on interoperable applications (starting with the system's architecture) must be refined.

2.6 Common Middleware Concepts

As an extension to the ISO-OSI model, communication between open systems can be described in accordance to Figure 2.3 [Blobel, 1996a; Leguit, 1992]. According to the *Leguit* model, integration of applications can be implemented at different levels. The lower level integration type *interfacing* only provides data from the invoked application (← →), sometimes prepared by presentation tools of the invoking application (← • • • →). The integration type *integration* requires interconnection at the object level (↔) providing data and methods of the communicating applications [Blobel, 1996a; Leguit, 1992].

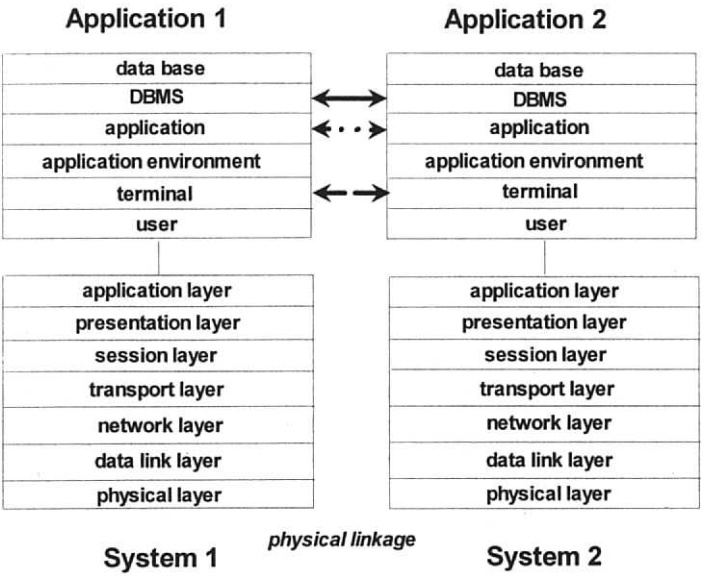


Figure 2.3: Integration model

To support interoperability of applications, the invoking client or application performs a request for services including information as well as methods related to the data. Therefore, *shared care* supporting systems need an integration type *integration* or object (service, functional) integration instead of *interfacing* or integration of data. As mentioned in the context of Figure 2.2, some services are common services, as object identification (naming services), time services, security services, transaction management, etc. Other services are provided by the requested application. The handling of objects, the invocation of services and therefore the provision of adequate interfaces are the objectives of middleware, enabling communication and co-operation of application systems from different vendors on different platforms and with different application environments.

By our knowledge, a comparative analysis of the important healthcare relevant architectural approaches has not been published anywhere before our studies have been performed [Blobel and Holena, 1996; Blobel and Holena, 1997]. The next chapter deals with this analysis facilitating the understanding of our generic model for security enhanced information systems, introduced in Chapter 4 and demonstrated in Chapter 6.

2.7 Summary and Conclusion

The change of paradigms in health systems and health policy must be accompanied by adequate paradigm changes for health information systems supporting this environment. The resulting information systems have to enable interoperability at the semantic and service level providing integration at the integration type level. The way of choice is a very advancd middleware approach facilitating co-operation at service and knowledge level including the self-organisation of complex systems. Factors influencing systems' policy in the sense of categorisation attribute have been isolated.

3 Comparing Implemented Middleware Concepts for Advanced Healthcare System Architectures

3.1 Introduction

The informatics applications, used by the providers to support their care processes information-related, have to support also their communication and co-operation. For that reason, application systems must be capable of supplying each other with information and functionality, i.e. of sending and receiving requests as well as of providing and using services. The most successful approaches implemented for health information systems meeting these challenges are the middleware architectures CORBA of OMG and the European DHE as well as the HL7 standard including the HL7 communication server. In this chapter, these practically available approaches are presented and compared. To realise a real open architecture of health information systems, the possibilities of harmonisation and combination of different approaches are considered.

Some of the presented results are part of our investigations within the HANSA project [HANSA_WWW] funded by the European Commission.

The middleware concepts discussed are under continuous development and improvement as everything we are dealing with in ICT. Years after having performed the presented studies and innovations and in parallel to the process of writing this book, HL7 version 3 and CORBA 3 were under development but also first steps towards e-Health got started. In practice, however, the older versions of the architectural approaches are still in place and often even dominating. Regarding HL7 for example, most of the HCE are still using interfaces according to version 2.1 or version 2.2 specifications. Version 2.3 or even version 2.4 are restricted to the latest developments. The same is true for CORBA meeting CORBA 1 or some CORBA 2 implementations although the ORB vendors are fighting with CORBA 2 and nowadays with CORBA 3. Therefore, both the practically deployed approaches as well as the newer paradigm changes are discussed in the next chapters. Beside these specifications practically used, fundamental changes in paradigms occur. The improvements are still in an experimental stage even if some demonstrators run meanwhile. These innovations mentioned are referred to in Chapter 5.

3.2 CORBA

The *Common Object Request Broker Architecture (CORBA)* has been developed and is being elaborated by the Object Management Group (OMG), a non-profit consortium of more than 1,000 software vendors, developers and users, created in 1989 with the purpose of promoting the theory and practice of object technology in distributed information systems.

3.2.1 Concepts

CORBA is based on the popular *object-oriented paradigm*. Its most fundamental concept is that of an *object* [OMG, 1995b]. An object can represent, in general, anything that is unambiguously identifiable and capable to provide one or more services, which can be requested by some kind of *client*. Associated with each object is a set of *methods* and a set of *attributes*. The former represent the provided services whereas the latter represent the state of the object and the information passed during the request or produced when services are provided.

This general characterisation holds for both the intensional and the extensional aspect of an object. To characterise each of the aspects more deeply, additional concepts are needed.

From the intensional point of view, objects are classified into *types*. A type is determined by the associated attributes and methods. Different types can be connected through a *subtype* relationship, which induces *inheritance* of attributes and methods. Multiple inheritance is possible.

From the extensional point of view, each instance of an object is associated with a computational *object implementation,* sometimes also called a *server.* If a client of a certain request is itself an object implementation, it can act at the same time as a server for another request. However, a client generally does not have to be an object implementation. Each object implementation consists of three parts:

- *Operations,* implementing the services represented by the object's methods,
- *Data,* which implement the object state and information represented by the attributes,
- *Interface,* implementing the ability to accept requests and to return information. It has the form of a specification that specifies passed parameters, return mode, as well as links to request context and to exception handling methods. For each operation, the corresponding part of the interface specification is called *signature* of that operation. As parameters, also interfaces may be passed. An inheritance relationship between objects induces an inheritance relationship between the corresponding interfaces.

One object implementation may be associated with several instances of an object, or a separate implementation may be associated with each instance of the object. Finally, a separate implementation may also be associated with each instance of each of the object's methods.

Interfaces are grouped into *modules* so that each module comprises interfaces to servers implementing related methods. Modules can be hierarchically embedded. To assure unambiguous requesting of services, interfaces are specified in a special *interface definition language (IDL).* However, there exist mappings of IDL to several common programming languages.

Though object orientation is undoubtedly the most important feature of CORBA, the object-oriented paradigm is even in this paradigm not used quite consequently. For efficiency reasons, some important CORBA functionality is not realised using object implementations, they rely on so called *pseudo-objects* instead. Pseudo-objects are superficially very similar to real objects – they are implemented by means of methods and interfaces, and their interfaces are specified in IDL. However, pseudo-objects are actually no objects and can not be invoked in the same way as objects are. Moreover, for most of them, the interface can not be passed as a parameter in a request. Examples of pseudo-objects include *Environment*, *Request*, *Context*, or *Current*. Figure 3.1 represents basic concepts of CORBA.

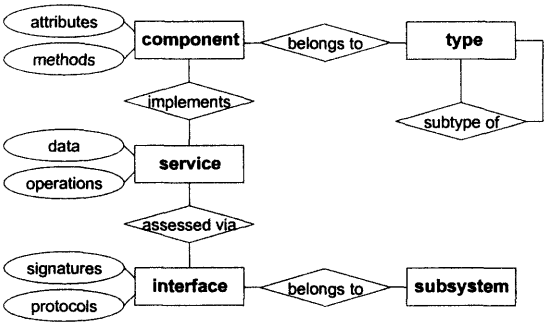


Figure 3.1: Basic Concepts of CORBA

3.2.2 Architectural framework

The architectural framework of CORBA consists of the following fundamental parts:

- A pseudo-object called *object request broker* (ORB), providing mechanisms to handle requests sent by clients to objects. It realises the object interconnection bus.
- Two layers of widely used objects, which form a component situated between ORB and the utmost layer of *application objects*. The lower layer of *common object services* [OMG, 1996a] provides basic functionality for using and maintaining objects. The higher level of *common facilities* [OMG, 1995a] provides general purpose capabilities useful in many applications.
- Mappings and protocols providing interoperability between different ORB implementations.

Each of these components will now be described in some detail.

The *object request broker* is responsible for locating an object implementation, preparing it to receive the request and communicating the data making up the request. It provides both static and dynamic interfaces to object services. It can be implemented in various ways – as a program, as a library, resident in an underlying operating system, or even distributed in the object implementations and clients it supports. However, its external interfaces are standardised and implementation-independent. It is structured into the following components, most of them relating specifically to clients or to servers:

- *ORB core* provides the basic representation of objects and communication of requests. This component is implementation-specific, and it is a matter of the components above the core to mask the differences between individual ORB core implementations.
- *ORB interface* is the primary interface of the ORB pseudo-object. It provides only a few basic operations useful for both clients and servers, such as object reference conversions, list operations or getting the default context.
- *Object interface* is an interface for operations on references to requested objects. It allows clients to create object requests and to obtain references to information on the requested object, stored in the interface and implementation repositories (see below).
- *IDL stubs* provide a client-side interface to object-type specific APIs for object invocation. The APIs themselves must be provided when the corresponding object is implemented. The stubs, generated by the IDL compiler, realise static interfaces to object services.
- *Dynamic invocation interface* is an object-type independent client-side interface for object invocation. The dynamic invocation APIs identify the required service and provide communication with the client at run time. In this way, even new services can be discovered and bound, not yet known to the client issuing the request.
- *Object adapter* provides the interface to servers, accepting the request for service, instantiating the requested server, and supporting the request-service process. The interface to an object implementation is provided by a *static IDL skeleton* and a *dynamic skeleton*, which are server-side analogies of the IDL stubs and the dynamic invocation interface, respectively. The static skeleton is generated by the IDL compiler and provides interfaces to each operation belonging to object implementations for the respective object type. The dynamic skeleton provides binding mechanisms for servers at run time. This mechanism can also be used for building bridges between different ORB implementations. The skeletons are connected to the object adapter through implementation-specific private interfaces. The variety of object granularities, lifetimes, policies, implementation styles and other properties makes it difficult to provide a single object adapter, convenient and efficient for all objects. Therefore, the CORBA standard speci-

fies a *basic object adapter*, which should be appropriate for objects without any special requirements. In addition, other object adapters can be specified, tailored to particular groups of object implementations with similar requirements. In that context, the specification of the *Portable Object Adapter* (POA) of CORBA 3 presented in Chapter 5.5 should be referred to.

- *Interface repository* contains persistent objects that provide the interface specification and related information for registered object implementations in a form available at run time. The repository is specific to a given ORB implementation, but its contents and interface are standardised.
- *Implementation repository* contains information about supported object implementations as well as administrative data (for example, trace information, audit trials, security). It allows the ORB to locate and activate object implementations as well as to obtain other useful information on them (debugging information, resource allocation, etc.). The repository and most of its content are specific to a given ORB implementation.

Common object services provide basic functionality additionally to those provided by the ORB itself. The interfaces of these components are generated by the IDL compiler. The following object services are available: *Naming, Persistence, Transaction services* (using a *transactional client,* a *transactional server*, and a *recoverable server)*, *Object lifecycle management*, *Event notification*, *Relationships*, *Concurrency control* (Locks), *Externalisation, Query management, Licensing, Properties* and *Security services*. Further services under development are: *Time management*, *Start-up services,* and *Trading*.

Common facilities provide higher level service interfaces, typically depending on underlying object services. The interfaces to this component are generated by the IDL compiler, too. The facilities may be horizontal, pertaining to different application areas, or vertical, within the market of a particular area. The following horizontal facilities are available:

- *User interface,* providing connection to the user environment (e.g., OpenDoc, Microsoft's OLE). They include desktop management, help services and presentation of objects in compound documents.
- *Information management*, supporting the creation of information models and schemata, persistent storage of information and its retrieval, and the interchange of data and information.
- *System management*, providing system administration function interfaces for handling distributed objects, such as policy management, instrumentation, data collection, security management and customisation.
- *Task management*, supporting the request/service process through the co-ordination of activities using workflow, agents and the management of long transactions.

Vertical domain facilities are standards for interoperability in a particular application area. Facilities required in healthcare are for example the *Medical record object model framework* and the *Master patient index framework.*

Application objects do not belong to the CORBA standard itself. Nevertheless, they form CORBA's external context, a layer using and build on top of services provided by the ORB, object services, and common facilities. They implement the end-user applications. Figure 3.2 demonstrates the general architectural frame of CORBA [OMG, 1995a].

Due to the openness and flexibility of CORBA, the interfaces provided by different ORBs are likely to be incompatible. Client objects may be able to communicate through their respective ORBs only if they lie in the same *domain*. A domain, in the CORBA approach, is a set of objects sharing certain implementation characteristics and obeying the same naming

conventions and communication rules. Typically, a domain includes only ORBs of the same vendor, or even of the same type.

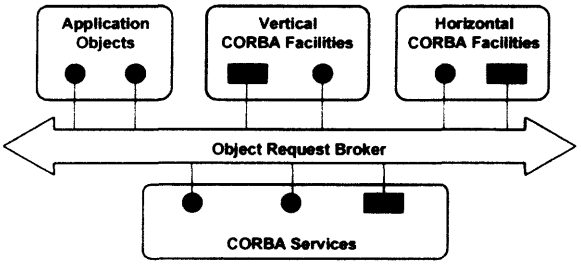


Figure 3.2: Architectural framework of CORBA [OMG, 1995a]

To remedy this situation, version 2.0 of CORBA includes an interoperability standard enabling an ORB to pass requests to an ORB in a different domain. To this end, a *bridging mechanism* is needed, translating requests between domains. The bridging between two domains may be either *immediate*, or *mediated*. The former is accomplished by means of a single mapping between those domains, called *inter-ORB bridge*. The latter is realised by means of two mappings between the respective domains, called *half-bridges*, and by means of an interleaving non-ORB environment (e.g., Internet). Besides bridging, the communicating ORBs need a mutually agreed communication protocol. In the CORBA interoperability standard, the *general inter-ORB protocol* (GIOP) is specified to this end. It is used above of any connection-oriented transport protocol that meets a minimal set of assumptions, and to enable the interoperability between ORBs, it must be completed through a particular transport protocol of that kind. For the TCP/IP environment, a specific transport protocol was defined in the CORBA interoperability standard, called *Internet inter-ORB protocol* (IIOP), which can be used as the transport protocol below GIOP. Instead of GIOP, or even instead of the pair GIOP/IIOP, *environment-specific inter-ORB protocols* may be specified for particular networking or distributed computing environments.

3.2.3 Relevance for healthcare enterprises

Especially European authors have predicted no importance of CORBA for healthcare in the next few years. This was originated by the concentration of OMG towards basic services at the beginning and therefore by the rare availability of domain-related solutions. In spite of that, there are some examples of medical applications under development that rely heavily on the CORBA approach. Some of them will now be briefly introduced.

In a project called *InterMed*, the Brigham Harvard Medical School and the Boston Women's Hospital have been undertaking efforts for several years to construct advanced healthcare systems from CORBA components [Deibel and Greenes, 1996]. To this end, they use the CORBA-based component integration and software development environment and tool-set *Arachne* [Deibel and Greenes, 1995]. The objectives of the project concern the specification of component interfaces, IDL-defined stubs, distributed component invocation, and run-time object services.

Another example, related to the virtual patient record and its crucial role in distributed healthcare, is the telemedical system *TeleMed*. It has been developed by the Los Alamos National Laboratory, in collaboration with the National Jewish Center for Immunology and Respiratory Medicine [Forslund, 1996; Forslund and Kilman, 1996; George, 1996].

Also within the area of electronic medical imaging, significant efforts are directed to the development of an object-oriented framework for high-speed distributed electronic medical imaging systems. These activities are performed by Kodak Health Imaging Systems and the Washington University, St. Louis, as a part of the electronic medical imaging project *Spectrum* [Pyarali et al., 1996a,b]. Their main achievement is the *Blob streaming framework*, combining the flexibility of CORBA-compliant technologies with the efficiency of lower-level transport mechanisms (e.g., sockets).

An important support is expected to come from related activities performed by the Joint Working Group for a Common Data Model in Healthcare, and especially, from *CORBAmed,* the healthcare domain task force within OMG [OMG, 1996b]. CORBAmed explicitly states its mission as „to improve the quality of care and reduce costs by CORBA technologies for interoperability throughout the global healthcare community" [OMG, 1996c]. It has already initiated the technology adaptation process to standardise interfaces for healthcare domain vertical facilities. The first and most important specifications provided are, e.g.,

- Patient Identification Service (PIDS), specifying a Master Patient Identifier (MPI) service connecting different patient ID established at different HCE,
- Terminology Query Service (TQS) (formerly Lexicon Query Service (LQS)), defining a medical data dictionary as knowledge base, terminology services, and directory functionality,
- Clinical Observations Access Service (COAS), which facilitates result reporting,
- Resource Access Decision (RAD) framework, providing access decision support mechanisms on the application level usable in many circumstances as well as for different resources.

Following, the work was dealing with transcription, decision support, clinical image access, pharmacy, OO-EDI, etc.

OMG is currently standardising an interoperable bridge between CORBA and XML considered in Chapter 3.4. This may make HL7 version 3.0 implementable in the OMG technology without further standardisation. For recent developments and innovations although they are of no practical importance at the moment, see Chapter 5.

A more comprehensive list is available at the CORBA Website http://www.omg.org.

Other related approaches competing or co-operating with CORBA are Microsoft's OLE and IBM's OpenDoc. However, these concepts are not considered in detail within the framework of this monograph.

3.3 DHE

The *Distributed Healthcare Environment (DHE)* is a middleware architecture permitting co-operation and data sharing between end-user applications including legacy systems [Ferrara, 1995a,d]. It has been defined and developed in the EU projects RICHE and EDITH, and is now being implemented in more than 20 hospitals as a part of the former EU project HANSA as well as used in activities to the Eastern-European countries.

3.3.1 Concepts

From the functional point of view, probably the most important concept pertaining to DHE is the concept of a middleware *service,* representing an elementary functionality provided or an elementary task performed by the middleware. At the same time, the concept of a service also reflects, from the enterprise point of view, elementary services, functionality and tasks of a healthcare centre [Ferrara, 1995b].

Each service accesses, retrieves, adds, deletes or modifies some information. In DHE, information is represented by means of the basic concepts used in the common entity-relationship (ER) modelling, i.e. entities, relationships and attributes. Thus, each service is connected to one or more *entities* or *relationships* between entities, and the information it deals with is described by means of *attributes* of the involved entities or relationships [Ferrara, 1996].

Finally, there is a concept in DHE that is in a way transversal to services, namely the concept of an *activity* (instantiation of an activity being called *actual act*). This concept allows DHE to support collaboration between individuals from different units of a healthcare centre, as well as their interaction with the outside world in requesting, accepting, planing and performing their tasks [Ferrara, 1995c]. Each activity represents a repeatedly occurring sequence of actions and events, which are either performed through the middleware services, or at least some information about them is recorded by the middleware. The former establishes a connection between activities and services. The latter, on the other hand, establishes a connection between activities and the pertaining information on patients, resources, parts of the body, techniques and methods, results, etc. (more precisely, between activities and the entities, relationships and attributes representing that information). In addition, in the sequence of actions and events represented by an activity, specific sections can be identified corresponding to some specific states of some involved entities or relationships. To represent such sections together with the related states of entities and relationships, the concept of the *status* of an activity is used (foreseen, requested, ..., terminated). To represent their possible ordering (or the actual ordering, in the case of an actual act), the concept of the *life cycle* of an activity / actual act is used. The aggregation of actions and events into an activity, as well as the hierarchical aggregation of sub-activities into a higher-level activity is described using the concept of an *execution profile*. Other connections and dependencies between activities are captured by means of the concept of a *complementary profile*.

Similarly to services, the concept of an activity reflects, from the enterprise point of view, real activities performed in healthcare centres, both clinical and organisational or managerial ones. Figure 3.3 represents the basic concepts of DHE.

3.3.2 Architectural framework

The architectural framework of DHE is mainly determined by its decomposition into individual components corresponding to different task areas [Ferrara, 1996]. DHE comprises the following healthcare-related middleware components:

- *patient manager*, responsible for supporting the patient identification, and for the management of the personal, clinical and epidemiological information on patients needed for administrative, statistical and similar purposes;
- *act manager*, responsible for supporting the interactions between units or individuals, and for the management of activities and individual acts;
- *health data manager*, responsible for managing detailed medical information on patients and interchanging it with other systems;
- *users and authorisation manager*, responsible for providing a common mechanism to describe and perform the authorisation of the individual users, with respect to the access to information, and to the execution of different functionality provided by the applications on top of DHE;
- *resource manager*, responsible for managing the availability, characteristics and actual state of materials, equipment, staff and other kinds of resources;

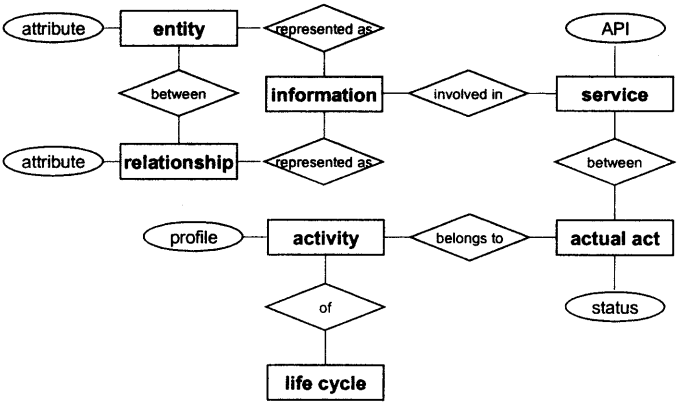


Figure 3.3: Basic concepts of DHE

- *costs and performance manager,* responsible for the management of information concerning the enterprise evolution, in terms of the quantity and costs of resources used, and of the amount and costs of activities performed.

In addition, DHE includes a number of services that are not healthcare-specific, though they are indispensable for DHE to fulfil its role as a middleware. Therefore, they can not be assigned to any of the above healthcare-related components, but could rather be considered as a *generic middleware component.* That component is responsible for managing sessions, communication, for support of dealing with information encoded in extended data, for management of system information, etc. [Ferrara and Sottile, 1996].

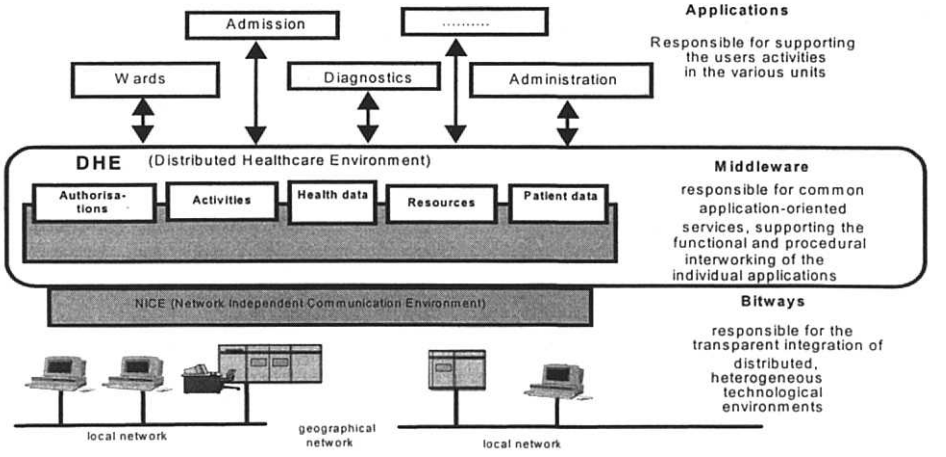


Figure 3.4: Architectural Framework of DHE (after [Ferrara, 1995a])

The architectural framework of DHE is actually only the *middleware layer* of a layered architectural framework of the healthcare information system the DHE belongs to (typically, a hospital information system or a territorial information system) (see Figure 3.4). Below it, there is the *bitways layer,* providing the basic technological infrastructure of the information system, including support of its distribution and network requirements. On top of it, there is the highest one, *application layer,* comprising the specialised individual applications.

3.3.3 Relevance for healthcare enterprises

As noted in Chapter 3.3.1, DHE services and activities reflect the real services, tasks and activities encountered in healthcare centres. Moreover, the ER information model employed in DHE reflects the enterprise view of a generic healthcare centre, which is fixed as the European HISA standard already mentioned [CEN, 1995]. In other words, the semantics of the underlying concepts explicitly supports healthcare interoperability using the DHE middleware in a healthcare environment.

Consequently, the DHE middleware is highly relevant to healthcare enterprises, especially if their enterprise view basically corresponds to that of a generic healthcare centre, as in the case of hospitals and territorial healthcare networks.

3.4 HL7

15 years ago, HL7 (Health Industry Level 7 Interface Standard) was founded in the USA as an association of vendors, users, and organisations who were interested to support and to promote the communication between information systems (applications) within hospital environment based on a healthcare-domain-related electronic data interchange (EDI) standard. Meanwhile, other countries like Argentina, Australia, Canada, Finland, Germany, India, Japan, Korea, Lithuania, New Zealand, UK, Taiwan, and The Netherlands are officially involved as affiliates. The standards efforts for interoperability in healthcare, including the standardisation of HL7, have been forced by activities to harmonise the different electronic data interchange standards of IEEE (Institute of Electrical and Electronic Engineers), ASTM (American Society for Testing and Materials), and other organisations.

3.4.1 Concepts

HL7 is a communication standard for information interchange in healthcare environment, especially in hospitals. Beginning with version 2.3, the original orientation to hospitals has been, and will further be, extended to the complete healthcare sector within the next versions of the standard.

HL7 aims at enabling communication between applications provided by different vendors, using different platforms, operating systems, and application environments (e.g. programming languages, tools). In principle, HL7 enables communication between any systems regardless their architectural basis and their history. That means that HL7 supports communication between real-world systems, newly developed or legacy. This is achieved through syntactically and semantically standardised messages. HL7 interfaces realise the request/service procedure in the sense of sending and receiving these standardised messages. The communication is managed by communication servers.

HL7 is a protocol for the exchange of healthcare information, defining both messages and the message exchange format. Contrary to the approaches described in the preceding paragraphs, HL7 is currently not an architecture supporting design of healthcare applications. It realises a set of message-based transactions between healthcare applications at level 7 of the ISO-OSI model of open systems interconnection. The HL7 version currently mainly used (V.2.3.2) contains *common specifications* (Chapter II: common rules, formats, control segments, queries, encoding rules; Chapter VIII: master index files) and *chapter-specific specifications* (trigger events, messages as well as segments and fields). The variable length delimited ASCII messages, produced from the abstract message definition by standardised encoding rules, are human-readable. HL7 functional areas include typical healthcare (clinical) domains as, e.g., *ADT, Registration, Orders, Results, Financial, Master files, Non-ASCII character sets, Query language support, Medical documents, Clinical trials, Immu-*

nisation reporting, *Adverse drug*, *Reactions*, *Scheduling*, *Referrals*, and *Problems and goals*.

The HL7 message protocol supports *unsolicited messages* and *solicited messages* alike. HL7 realises both *basic* and *enhanced acknowledgement paradigms*. In contrast to the European standards efforts in the EDI domain (CEN TC 251, EDIFACT[7]), neither the previous HL7 versions nor Version 2.3 include modelling.

The modelling and/or architectural issues of the next paragraphs are especially related to the HL7 Version 3, which is based on an object-oriented model of healthcare information and currently under development. This version will contain the above mentioned common specifications, a HL7 reference model, chapter-specific specifications (based on an information model, and an interaction model), a hierarchical message description, and finally, implementable message specifications (related to ASN.1, Microsoft's OLE, and CORBA). In spite of this fact, some of these descriptions, remarks and assessments are valid already for earlier HL7 versions. Figure 3.5 presents the basic concept of the HL7 architecture.

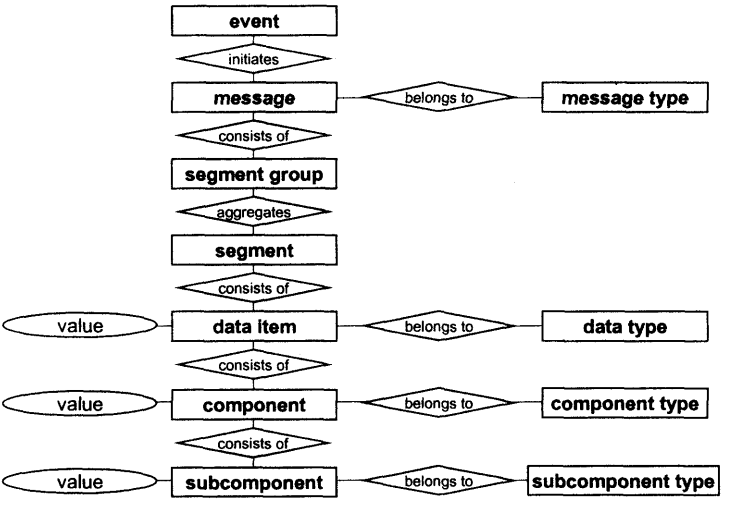


Figure 3.5: Basic Concept of HL7

3.4.2 Architectural framework

The basic principle of HL7 is a *point-to-point information interchange paradigm* (1:1 or 1:n in the case of broadcast). Communication is controlled by trigger events (in the case of the *trigger event paradigm* of process coupling, unsolicited or real-time) or by query/response interchange (in the case of *query/response paradigm* of retrospective interchange or solicited). The data interchange is performed between *communicant applications*. A *source application* initiates the message interchange as the source of a given information set. A *recipient application* acts as the recipient of message interchange. In general, the message interchange between source and recipient is mediated. A *mediator application*, responsible for the mediation, may also provide additional services, e.g. *Transaction management*, *Broadcast*, *Transformation* into meta-protocols (mapping to alternate message encoding and/or alternate content coding schemes). The feasibility and efficiency of information interchange depends on the extent of the common information and functional domain of the communicating applications. This commonality includes the information ob-

[7] EDI for Administration, Commerce and Transport

jects and object properties as well as the equivalent and/or complementary processing functions. High-degree commonality describes the conditions for efficient information interchange and interoperability, also called tight coupling. If high-degree commonality exists, then a robust interface solution can be achieved.

HL7 enables real-time trigger event initiated messages interchange managing concurrent process support. As mentioned above, the HL7 query/response paradigm includes also non-concurrent, retrospective interchanges. HL7 uses, in principle, unique object identifiers and allows a controlled order in time of messages.

According to the HL7 *abstract message definitions*, an HL7 message as smallest exchangeable unit consists of *required* or *optional segments*, which can partly be repeatable. The abstract message starts with a *message header segment* followed by *control segments*, controlling the interchange process, and *data segments*, containing the data elements. The structure of the abstract message as well as the order of their segments is described by the *abstract message syntax*.

The *objects* are defined in the standard. The *instances* of the objects can be defined within the information interchange standard, by references to other standards (e.g. ISO, WHO (ICD)) or by user-defined tables.

Preparing HL7 version 3, the HL7 community started *object-oriented modelling* of processes and the related EDI. In this context, the specification of the HL7 communication standard consists of:

- *common specifications* (common rules, formats, control segments, queries, encoding rules, master index files),
- *the HL7 reference model*,
- *chapter-specific specifications* (information model, interaction model, hierarchical message description), and
- *implementable message specifications*.

Additionally to the above mentioned information model and interaction model, the following message development components are defined in HL7 version 3:

- *transaction scope*,
- *use case model*,
- *messaging (sub-)model*.

The transaction scope (*scope statement*) gives a high level description of the related *use case*. Within the use case model, the use case will be described in detail including resulting use cases and *actions*, and the *actors* will be defined.

The HL7 reference model is an information model of the HL7 domain, defining classes, associations, and attributes. It serves as an information model and source for the message information sub-models (within the hierarchical message description process). The HL7 reference model supports consistent definitions in general, as well as the definition and the reconciliation in the case of new message developments. It is the contribution of HL7 to harmonising models with other standards organisations.

The interaction model defines interactions, the application profile, and trigger events. It describes in detail trigger events, senders, event dependencies, receivers, and receiver responsibilities.

The information model defines classes, connections, and attributes as well as subject areas and objects' lifecycles.

The information model and the interaction model have to be co-ordinated with the HL7 reference model.

For the creation of messages, a message information (sub-)model has to be developed as a subset of the reference model pertaining to messages. The hierarchical message description describes in detail the parts of a message.

Information interchange requires the use of a standardised networking platform to deliver real messages. Currently, special HL7 communication servers are available at the market, managing the information interchange within a middleware based on HL7 standardised messages. Such HL7 communication servers are implemented using centralised or decentralised architectures. Unfortunately, these solutions are used in parallel to other protocols and integration concepts. The implementable message specification enables to adapt to standardised mediator applications or environments like ASN.1, CORBA, and Microsoft's OLE as such networking platforms. ASN.1 (Abstract Syntax Notation 1), the *ISO-OSI data interchange language*, is used in HL7 for *abstract message specification.* More precisely, particular encoding rules (*Encoding Rules-7*) are used for encoding HL7 messages.

For planning and implementation of HL7 interchange, an object-oriented analysis approach has been proposed.

Essential components are

- the business model providing the business context for the data/information model and the resulting message standards; it describes business area, function, role, and role/function, last to define interactions,
- the data/information model, defining subject area, class, and attributes, last needed to describe use case and message type,
- the message standard, defining data element, data interchange transaction, and message type, the latter supports business case interactions,
- the business case that identifies interactions between pairs of role/function combinations.

Figure 3.6 shows the original general scheme of the HL7 v3.0 development paradigm. More information on the recent paradigm change of HL7 version 3 including the new *Message Development Framework* is given in Chapter 5.4.1.

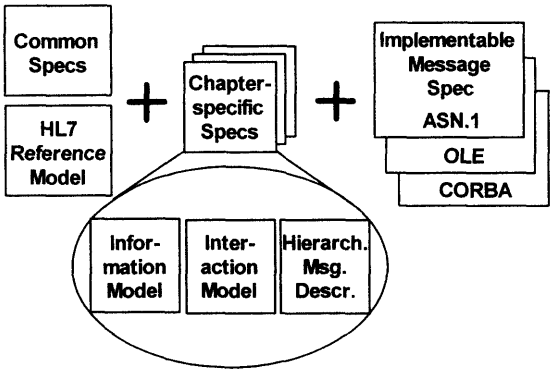


Figure 3.6: Original General HL7 v3.0 Development Scheme

Forming Special Interest Groups (SIG) e.g. for Component Based Messages (SIGCBM, formerly SIGOBT), Visual Integration (SIGVI, formerly CCOW), or Governmental Computer Based Patient Record (SIG G-CPR), recently, HL7 focused the ambitious goal to evolve a comprehensive architectural standard for interoperable health information systems, patient record architecture, absorbing streams like object orientation, component orientation, technologies like DCOM and ActiveX, CORBA, the Internet, methodologies as UML, Java, and XML. As universal document definition language (XML scheme), query lan-

guage (XQL/ML) stylesheet language (XSL) separating presentation and content as well as information exchange format, XML could be considered the most important development in the health information framework, providing a comprehensive universal standard format for health records, common documentation and communication.

3.4.3 Relevance for healthcare enterprises

The relevance of the HL7 communication standard for healthcare enterprises is obvious. Especially in the United States, but also in the countries of the HL7 International Affiliates like Canada, Australia, The Netherlands, Germany, Finland, UK and Japan, HL7 has become the dominant communication standard for healthcare system integration, and it is already available in products and solutions.

As a partner for the European standards agencies (CEN), ANSI formed the Health Informatics Standards Planning Panel (HISPP) in 1991, which was replaced with an ANSI designated Health Informatics Standards Board (HISB) in 1995. In 1992, HISPP together with other standards developing bodies formed the Message Standards Developers Subcommittee (MSDS). In this environment, IEEE has provided the secretariat for the Joint Working Group for Common Data Model (JWG-CDM), including ASTM, HL7, DICOM (Working Group of the American College of Radiology and the National Electrical Manufacturers Association), NCPDP (National Council of Prescription Drug and Pharmacies), and X12N (Insurance Subcommittee of ANSI Standards Committee X12). The objective of the JWG-CDM is the development of models and specifications that are needed to support a generic messaging standard [JWG-CDM, 1996]. The efforts of the JWG-CDM will improve the chance of the domain-related HL7 messaging and other harmonised healthcare information interchange protocols, realising interfaces to standard middleware architecture specifications like ASN.1, CORBA, and OLE. In this context, the activities of the HP-promoted Andover Group should be mentioned, which focussed on products within the JWG-CDM framework, starting with HL7-related procedures using CORBA integration. Also the openness of HL7 to other paradigms as mentioned above including XML will consolidate the position of this standardisation body embedded in activities of the recently formed ISO TC 215 "Health Informatics". For recent developments and innovations even if they are of no practical importance at the moment, see Chapter 5.

3.5 Comparison of the approaches

To compare the architectural, functional, methodological, and technological framework of information systems, the ISO Reference Model – Open Distributed Processing (RM-ODP) can be used [ISO/IEC 10746-2]. This reference model defines possible views on systems such as Enterprise View, Information View, Computational View, Engineering View, and Technology View. These viewpoints are characterised as follows:

- Enterprise Viewpoint is focused on purpose, scope and policies for the system, promoting an understanding of the business environment and its influence upon the distributed system.
- Information Viewpoint is focused on the semantics of the information and the information processing performed. This viewpoint essentially concerns the articulation of business rules and content to be supported by the system. Addressed within this viewpoint are the static, invariant, and dynamic behaviour of the system.
- Computational Viewpoint enables distribution through functional decomposition of the system. In less precise terms, this viewpoint provides the logical design of the system through encapsulation of capability, separation of functionality, and interface definition.

- Engineering Viewpoint is focused on mechanisms and functions to support distributed interaction between the components of the system. Essentially, this is to determine the required distribution aspects of the system (for example, the distribution architecture to be used).
- Technology Viewpoint focuses on the choice of technology to be employed within that system. This is a description of the implementation of the system and testing requirements.

All discussed concepts and architectures now start with business process models to describe the real-world structures, functions and conditions, i.e., with the related real-world objects. They are promoting the currently undergoing paradigm shift from client-server technologies or even monolithic systems to component-based information systems.

CORBA realises a strongly object-oriented concept, providing object-oriented features like global object identification, managing of distributed objects, persistence and inheritance, object lifecycle services, and modularisation up to an atomic level. In its functionality, CORBA also covers the lower-level data interchange protocols. Domain-specific services are built on top of the core functionality of the middleware as application objects or vertical common facilities. Through the use of multiple object adapters, CORBA can support virtually any style of object implementation, including wrapping of non-object-oriented applications, such as legacy systems. It realises direct binding to object-oriented languages like C++ and Smalltalk. Through the support of dynamic invocations, CORBA allows clients to discover and bind new services at run-time. This significantly facilitates the integration of new object-oriented applications including specialised healthcare applications. On the other hand, CORBA does not provide any direct support for the analysis and design phases of such applications through a predefined information model pertaining to the healthcare area. The developer is supported only indirectly, through the information management facility, which can easily handle such models. Through its interface-generating mechanisms, CORBA provides solutions for portability, scalability and interoperability in a heterogeneous environment. Some conceptual and functional weaknesses of the first version of CORBA have been remedied in the current version 2.0. Problems that are still open include, for example, the perfect handling of massively parallel processes, which occur in healthcare. For such applications, only special transaction processing monitors are currently available.

In summary, CORBA realises distributed objects, supported through object services, including transactions, relationships, concurrency, and externalisation. The basic principle of CORBA is the collaboration of client/server components through a common interconnection bus, hosting ORB components, object services and common facilities. The facilities may be horizontal, interconnecting components from different application domains, or vertical within a particular domain. Within the RM-ODP schema, the CORBA Common Object Services Specifications (COSS) provide an engineering view on the system specified. The CORBAmed vertical facility specifications are realising a computational view. With the upcoming problem of reusability, OMG adapts its approach introducing the Business Object Component Architecture (BOCA) and interpreting *facilities* as *component frameworks* [OMG, 1998b]. Such reusability is eased by the component paradigm because the reuse of business objects encapsulating business-relevant data and functions into a separate and consistently affectable unit is more efficient and better manageable than fine-granulated classes. By that way, an enterprise view is realised completing the views needed and overcoming the existing gaps in the CORBA approach.

DHE provides services reflecting elementary functionality and tasks of a healthcare centre. It also realises some non-healthcare-specific services in the sense of generic middleware components. However, DHE does not include any lower level protocols since these are

supposed to be implemented in an underlying bitways layer. The predefined ER-based information model of a generic healthcare centre facilitates the integration of new healthcare applications. On the other hand, unlike CORBA, the developer of such an application must completely know in advance what DHE services will be needed for its implementation. The DHE approach is rather globally process-oriented and definitely non-object-oriented. Nevertheless, DHE includes some concepts usually encountered in object-oriented approaches, like object status and object lifecycle. Some of the DHE managers provide functionality required from the CORBA vertical common facilities for the healthcare application domain. This indicates a possible strategy for future development of the DHE middleware, as well as for the specification of prospective healthcare-related CORBA vertical facilities. An important advantage of DHE is the full conformance to the CEN TC 251 draft proposal of the Healthcare Information System Architecture standard [CEN, 1995].

Both CORBA and DHE consequently realise the concept of a domain-specific *middleware layer*, an intermediate layer between the domain-independent infrastructure (such as operating systems, network management), and the end-user applications. The necessity of a middleware layer for an efficient integration of both new and legacy systems in a heterogeneous environment is now being increasingly recognised even in the area of healthcare information systems [Cooper, 1996].

Also HL7 can be considered realising a sort of middleware layer since it defines domain-specific abstract messages, interfaces for information interchange and standards for communication management which support communication between different applications, and between applications and the underlying infrastructure. However, the availability of standardised messages is actually the only provision for the integration of applications, both legacy and new ones, on top of HL7. No other support for the development of new applications is provided, neither for analysis and design phases, nor for the implementation phase. The activities of JWG-CDM and the Andover Group will focus the HL7 efforts to the message definition and generation domain using object-oriented modelling. The communication management will be delegated to standardised platforms like ASN.1, CORBA, and OLE. In that context, HL7 will introduce component interfaces. With respect to OLE, the interfaces support for example the sharing of healthcare data components and functions or automation components. In the future, a context interface will be provided, supporting medical context sharing features. The migration activities are aimed, e.g., starting with the former HL7 Special Interest Group on Object Brokering Technology (SIGOBT) [Rishel, 1996]. The efforts of JWG-CDM and the Andover Group and the integration of other activities and paradigms mentioned in Chapter 3.4.3 should lead to a fast melting and combining of different approaches. Finally, the XML orientation should not be underestimated.

HL7 does not allow free invocation of objects within the entire address space. In particular, a dynamic invocation is impossible. The above mentioned object-oriented modelling serves only the purpose of generating fixed messages, which have to be standardised. Due to the reduction of optional items in version 3, the degrees of freedom, but also the absent or reduced communication encountered in the preceding versions will be diminished. In that context it should be mentioned once again that HL7 currently does not tackle the EDI communication in an object-oriented way. Regarding the schema of views on systems, HL7 provides the information ones.

Schematically, the most important differences between CORBA, DHE and HL7 are compared in the table below (Table 3.1); their scopes are juxtaposed in Table 3.2. Regarding the concerns of the different approaches reflected on the RM-ODP, relations can be described according to Figure 3.7.

Table 3.1: Layered Scheme of the Architectural Approaches

Application Layer			
Protocol Layer	EDI (HL7, EDIFACT)		
Common HC Middleware Layer	Communication Servers	Distributed Health-care Environment (DHE)	Vertical Common Facilities
Basic Distribution & Communication Middleware Layer		NICE	CORBA

Table 3.2: Juxtaposition of the Scope of the Compared Approaches

Property	**CORBA**	**DHE**	**HL7**
Basic paradigm	object orientation, management of distributed objects	layered architecture, relies on a healthcare-specific data model	messaging concept
Architectural concept	yes	yes	no
Middleware of common services	yes, generic; healthcare-specific facilities can be developed	yes, healthcare-specific, also generic where needed	no
Interoperability	yes	yes	no
Level of standardisation	standardised architecture and common object services	European pre-standard CEN TC251 architecture	standardised messages
Adding new applications	partial development, using existing objects	partial development, using existing services and the underlying model	to be developed independently, using standardised messages
Support	limited	limited	many suppliers
Future developments	new services and facilities, including healthcare-specific ones	additional middleware services announced	next release with new messages

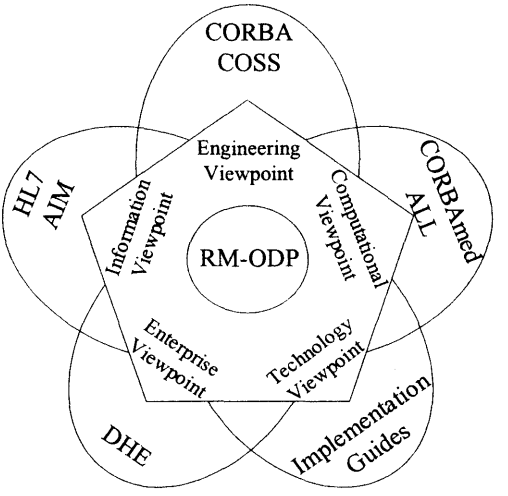


Figure 3.7: The Considered Approaches' Relation to the RM-ODP

All architectural approaches mentioned before are moving towards a component-oriented paradigm, absorbing each other's progressive ideas as well as new developments (e.g., XML) by forming liaison agreements, special working groups, etc. By that way, a harmonisation of the architectural paradigms proceeds.

3.6 Other Concepts

3.6.1 Distributed System Object Model

The Distributed System Object Model (DSOM) of IBM provides both an object model and a run-time implementation. It enables the provision and the management of a platform-independent and language-neutral binary class library. Originally, DSOM provides the workplace shell of OS/2 and OpenDoc, which is a standard for document linking based on Dynamic Link Libraries (DLLs). The access of a client to a server is provided by an interface reference. DSOM demonstrates similarities with CORBA as well as with Microsoft's DCOM. For the interface definition, the OMG IDL is used. IBM's DSOM is CORBA-compatible.

3.6.2 Distributed Component Object Model

Originally, the Distributed Component Object Model (DCOM) is Microsoft's version of document linking which is based upon DLLs like DSOM. DCOM has similarities to DSOM, is based on Remote Procedure Calls (RPC) similar to the Distributed Computing Environment (DCE), and is not compatible to CORBA. The object model COM is the basis of Microsoft's component architecture Object Linking and Embedding (OLE). Meanwhile however, there is a separation between both.

3.6.3 ActiveX

ActiveX is one part of the Microsoft's object-oriented program interaction technology that is built on the Component Object Model (COM). The re-usable software components (called ActiveX Controls) use the COM interfaces (and are therefore written in a language recognised by COM, like C/ C++, Java and Visual Basic) and run in an ActiveX environment (called container) on the same computer or in a distributed network consisting of Windows and Macintosh systems. In implementation, an ActiveX control is a Dynamic Link Library (DLL) module.

3.6.4 Distributed Computing Environment

The Distributed Computing Environment (DCE) of the Open Group provides an integrated solution for distributed applications in heterogeneous networks, which are based on Remote Procedure Calls (RPCs). Procedures (instead of object methods in CORBA) provided by servers are defined by a specific IDL. The functionality of DCE can be used by CORBA as underlying system services.

3.6.5 JavaBeans

JavaBeans (JB) is an object-oriented, platform-independent programming interface from Sun Microsystems for visual construction of re-useable program building blocks that can be assembled within the component architecture to build up applications. The blocks or Beans are written in Java so reasoning the name JavaBeans. The concept uses the Remote Method Invocation (RMI) of those Beans. CORBABeans must be considered in the same framework.

Starting with its own approach of Open Blueprint Architecture offering a well-documented comprehensive corporate architecture, IBM now moves towards an extensive component architecture based on Enterprise JavaBeans (EJB).

3.6.6 .NET

Recently, Microsoft launched a new initiative for using successful developments such as Java, J2EE, CORBA, and so on. .NET offers a much more advanced environment for programmers in the Microsoft domain using VB or Visual C/C++. .NET is a framework or platform consisting of different components such as a new fully object-oriented programming language C#, a common language runtime which runs bytecode in an Internal Language (IL), a set of base components providing various services (containers, networking, etc.), a new Active Server Pages version ASP+, Win Forms and Web Forms as new UI component frameworks usable from Microsoft Visual Studio, and finally a new generation of Active Data Objects ADO+. The latter enables data access eXtended Markup Language and the HTTP Simple Object Access Protocol (SOAP) for data interchange. IL, XML, and SOAP offer bridges into the framework for non .NET applications or environments.

3.7 Summary and Conclusions

The assessment of the different middleware approaches has shown similarities and differences concerning underlying concepts and architectural frameworks. The European approach DHE has the advantage of an advanced methodology in modelling and approval of models, terminology, and methods. The enhanced activities of JWG-CDM, including also experiences and results of the CEN working groups, will compensate that gap very soon. The official membership migration of different groups as CORBAmed, HL7, DICOM etc. will promote this process. In that context, the launch of ISO TC 215 "Health Informatics" plays an important role. A disadvantage of non-object-oriented approaches as DHE is the loss of adequately fine-grained components supporting flexible integration in the sense of the dynamic invocation. Therefore, the standard system model (standard process model) of DHE is fixing structure and functions in a "closed shop" solution. Such restrictions can also be observed in pseudo-objects of some vertical facilities.

All discussed approaches aim at communication and co-operation of distributed information systems including both newly developed and legacy applications. The co-ordination of the efforts and the co-operation within the G7 framework will accelerate the further development of the discussed concepts and the availability of related products. The increasing efforts to achieve harmonisation on an object-oriented basis could lead, in our opinion, to the combination of those different approaches. The results of CORBAmed Requests for Proposals (RFPs) encourage the migration on the basis of the presented juxtaposition of the compared approaches. The future development of DHE managers should take into consideration the CORBA concepts and specifications, proving quickly available healthcare-specific vertical facilities and pushing the general middleware concept in a common way.

All concepts promote the component-based information systems' paradigm. Therefore, each of them can be seen as supporting the development and implementation of really open distributed systems, including systems based on the Internet. Because of the urgently needed structural changes in the healthcare systems of all developed countries, the isolated and proprietary solutions must be replaced step by step by interoperable architectures enabling the informational support of realistic healthcare processes on the basis of fine-grained system integration. The start in that direction has already been made.

4 A Generic Component Model to Evaluate Architectural Approaches

In Chapter 4, the Middleware approaches CORBA, DHE and HL7, which are dominant in healthcare, have been introduced and discussed in detail. Additionally, the current Component Object Model (COM) architecture as well as the health information systems solutions merging the basic architectural paradigms presented as the SYNAPSE project and some other HISA[8]-related approaches have to be shortly referred.

4.1 Component-Based Analysis and Design of Systems

In this chapter, the knowledge of system analysis, design, and implementation within the software development will be reviewed and afterwards expanded to provide an appropriate methodology for our challenge for secure health information system architecture.

In many papers and thinking, the term component-based is immediately associated with a specific technology: CORBA, Microsoft ActiveX/COM/DCOM, JavaBeans (Enterprise and others), etc. providing a White Box view with an open and clearly defined internal structure. There is a need however to propose techniques and tools that are independent from a specific technology, but still practically enough to be instantiated in any of these technologies or emerging technologies in the sense of a Black Box consideration not regarding implementation details. Contrary to objects, components provide a higher and therefore more efficient granularity, support business processes as well as the data management. Contrary to objects whose interaction protocols are not entities separated from the objects and, therefore, that can only be reused in a new context if the same interaction protocol is used by other objects thus often inhibiting their reuse in other business processes, components are adaptable separating components with their basic functionality from connectors describing the components' relationships. Components enable the integration of legacy systems via *legacy wrapping*.

Fighting for a modelling paradigm enabling an open approach even for systems dealing with legal, organisational, and technological issues as security does, this ambitious challenge is especially true. Other paradigms as object-oriented development will co-exist with the component paradigm [Brown, 1996; Frost and Allen, 1997].

In this chapter, the objectives and current results of component-based analysis and design of complex systems are roughly investigated, looking for definitions, specifications, implementations, and corresponding tools to provide these steps. In that context, the widely accepted modelling methodology of UML (Unified Modeling Language) as well as the related Rational Rose product for system analysis, design and implementation is referenced. Component-based analysis and development is an interface-focused design approach which is characterised by a clear separation of component specification and its design and implementation. It supports Plug & Play and is architecture-centric. The characteristics of Component-Based Software Engineering (CBSE) are given in Table 4.1. It comprises also systems in general as shown below.

A component can be defined as follows:

"A component is a non-trivial, nearly independent, and replaceable part of a system that fulfils a clear function in the context of a well-defined architecture. A component conforms to and provides the physical realisation of a set of interfaces" [Kruchten, 1998].

[8] Health Information System Architecture, a European standard elaborated by the CEN TC 251.

A component represents a fundamental building block upon which systems can be designed and composed. Due to the underlying recursiveness, the component paradigm is very generic. Therefore, a system at one level of abstraction may simply be a component at a higher level of abstraction [Kruchten, 1998]

Table 4.1: Comparison of Development Models [Aoyama, 1998]

Characteristics	Conventional	CBSE
Architecture	Monolithic	Modular
Components	Implementation & White-Box	Interface & Black-Box
Process	Big Bang & Waterfall	Environmental & Concurrent
Methodology	Build from Scratch	Composition
Organisation	Monolithic	Specialised: Component, Vendor, Broker, & Integrator

An interface is a collection of operations that are used to specify a component service offered by a component (or a class) that is in turn implemented by a class or a component. The interface

- focuses on a component's behaviour, not the structure,
- serves to name a collection of operations and specify their signatures and protocols,
- offers no implementation for any of its operations,
- allows complete separation of specification from implementation.

The realisation of a component's interface is the offer of operations defined by that interface [Kruchten, 1998].

The description of a component interface describing the black box view of that component consists of

- a signature part, describing the operations provided by a component, and, based on that
- a behaviour part, describing the component's behaviour [Bergner et al., 1998].

If a component is depending on an interface, it requires the services of those components which realise the interface. A component may be dependent upon any number of interfaces.

4.1.1 The UML Modelling Methodology

In general, analysis and design of systems in hardware and software is based on a model describing state and/or behaviour of that system. Also the currently popular OO modelling techniques of Grady Booch, James Rumbaugh and Ivar Jacobson provide such an overall model consisting of the components classes, class categories, objects, subsystems, modules, processors, devices, and the relationships between them [Booch, 1994; Jacobson et al., 1992; Rumbaugh et al., 1991]. These model components mentioned possess properties which identify and characterise them. They can appear in none, one, or several of a model's diagrams associated with other components [Eriksson and Penker, 1998]. Thus, looking for the different components,

- the class category contains class diagrams and scenario diagrams associated with its components: classes and their objects, and nested class categories.
- the subsystem contains module diagrams associated with its components: modules and nested subsystems.
- the class contains its state diagrams.

- a model's top level contains the diagrams for its top level components as class categories, classes, subsystems, and modules, and its process diagram.

In OMT-2, four partial models allow capturing as well as analysis and design of the considered system or domain: the logical, the physical, the static, and the dynamic model. Contrary to other approaches the UML methodology, which is based upon the Booch methods, the OMT-2 methods of Rumbaugh, and the OOSE and Objectory methods of Jacobson, facilitates different views of the overall model described verbally by specifications and through different diagrams (e.g., logical diagrams, class diagrams, class structure diagrams, scenario diagrams, collaborations diagrams, component diagrams, distribution diagrams, activity diagrams, use case diagrams, sequence diagrams).

The UML views are:

- The use case view showing the functionality of the system as perceived by external actors. The use case view is described in use case diagrams and activity diagrams. Use case diagrams are basic descriptions influencing the other views. While the use case looks from outside the system using natural languages to describe the use case, the collaboration (context and interaction) diagram has an inside of the system perspective to describe interactions in time (sequence diagram), in space (collaboration diagram) and concerning the work (activity diagram). Finally, the scenario diagram describes a scenario in time (sequence diagram), in space (collaboration diagram) and concerning the work (activity diagram) via an execution path through the system.
- The logical view showing how the functionality is designed inside the system, in terms of the system's static structure and dynamic behaviour.
- The component view showing the organisation of the code components.
- The concurrency view showing concurrency in the system, addressing the problems with communication and synchronisation that are present in a concurrent system.
- The deployment view showing the deployment of the system into the physical architecture with computers and devices called nodes.

A scenario is a sequence of important interactions between objects as instances of classes within concrete application environments. Scenarios are used to represent critical requirements, depict the action of key mechanisms, and demonstrate desired series of operational cases. The scenarios can be described, considered and manipulated by two types of isomorphic scenario diagrams: the object message diagram and the message trace diagram. An object message diagram illustrates the existence of objects and the communication as the flow of messages among them. According to [Quatrani, 1998], Figure 4.1 characterises the UML diagrams as architectural views.

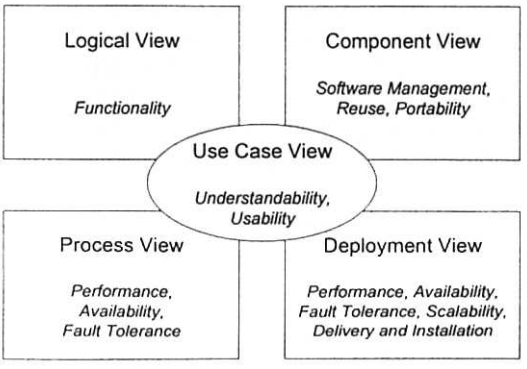


Figure 4.1: The UML Views of Architecture (after [Quantrani, 1998])

Figure 4.2 gives an overview about dynamic models in UML.

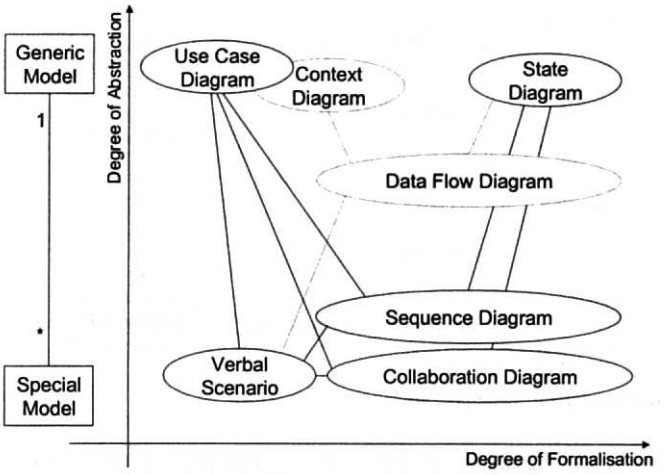


Figure 4.2: Overview about Dynamic Models in UML (after [Hruschka, 1998])

4.1.2 Basic Concepts and UML Presentation of Components

Component statics may be described by the UML component diagram. Component dynamics can be shown using sequence diagrams. Vertical bars represent the focus of control of the components over time. Messages (horizontal arrows) represent invocation of operations on the interfaces realised by the components.

A user describes the static structure of the elements of interest within a domain as a set of related types in a type model. The structural relationships among types represent the static constraints that exist among elements of the domain [Brown, 1998].

For each type in a domain the user describes its features (attributes and operations) in detail. Particularly important are the pre and post conditions that define the semantics of each operation by describing the state that must exist before the operation can take place, and the state that will result having executed the operation. Informal definitions of the pre and post condition can be given. However, more valuable are pre and post conditions in some formal, verifiable notation supported by the component modelling tool.

Interactions among types are modelled as collaborations. A collaboration diagram records the interactions among types in the domain as a sequence of messages (operation invocations).

The design perspective of a component encompasses more than a single class: it represents a number of classes which interact to provide a set of services. In UML, this can be represented as a subsystem: a type of package which realises one or more interfaces. To clarify use of a subsystem for particular use as the design representation of a component, the UML stereotype «component subsystem» can be applied. There is typically a 1:1 relationship between components and component subsystems, though for complex designs, subsystems may be nested to represented composite components. Like a component, a subsystem realises one or more interfaces and is dependent on none or more interfaces [Kruchten, 1998].

A component may occur as an identified component, as a service-oriented component, as package or framework, in that range increasing its granularity [Veluwen, 1999]. Figure 4.3 presents the basic concepts of components which will be refined furthermore.

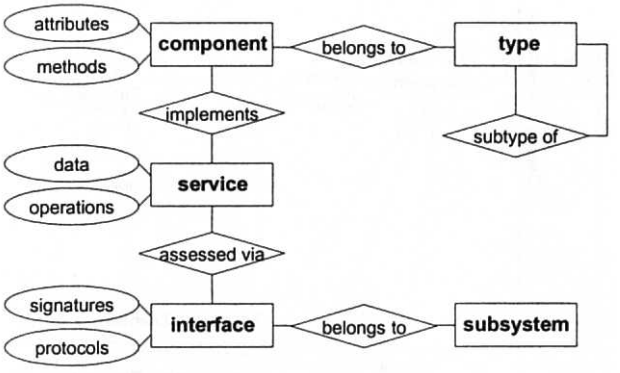


Figure 4.3: Basic Concepts of Components

For specification of the signature part of interfaces, component interface diagrams (CIDs) can be used. The behaviour part on the interface level can be described, for example, with state machines or sequence diagrams [Bergner et al., 1998].

4.1.3 The Domain Concept

As already introduced in the OMG specification [OMG, 1995c] and deployed for security modelling [Blobel et al., 1997], in the generalised component paradigm domains can be introduced giving afterwards evidence to our approach introduced during the last couple of years. Thus, to focus attention on some set of types or interactions in a domain, a user is allowed to create views focussed only on those elements that are of interest for some specific purpose in the component world. Therefore, domains can be considered to act as scope boundaries for describing behaviour. A user can import a domain into another, or can decompose a larger domain into a number of smaller domains. This supports both top down and bottom up development methods [Brown, 1998].

This component-related definition is consistent with the domain model used in Chapter 6.

4.1.4 Component Models for Real-World Systems

As demonstrated, the component paradigm provides a generalisation of the object-oriented paradigm neglecting underlying technology as well as some very strong theorems defining objects and their behaviour. Looking for policy-controlled security in complex information systems of Health Care Establishments (HCE), an appropriate description methodology must be established. According to Han [Han, 1998], the following excursus develops the expansion of the description up to the component paradigm.

In object programming, objects are characterised by attributes and by operations supposedly embedding all constraints about the object structure and interaction:

OOP: Object = attributes + operations (1)

Expanding the view to object-oriented analysis and design, the characterisation of objects is enriched to capture the sequencing and interaction of object manipulation:

OOAD: Object = attributes + operations + sequencing [~ interactions/scenarios] (2)

As mentioned above, the sequencing of object operations can be described using UML state transition diagrams for the object (class). The interactions of objects are considered in the context of scenarios (rather than relative to individual objects) and described using object diagrams and interaction diagrams.

Regarding architectural components defined within the Software Architecture Description Language (SDAL), framework, constraints, sequencing and interactions are specified explicitly:

SAD: ArchComponent = attributes + operations + constraints/sequencing/interactions (3)

In the context of Software Architecture Description, the attributes are the structural elements of the architectural component, and are usually those relevant to its interface (i.e., observable). The operations are those allowable on the architectural component. The constraints, sequencing requirements and/or interactions are those parts of the architecture description that constrain the usage/interaction and internal composition/state of the component.

Finally, also properties describing a component's environment can be included, such as reliability, performance, security, safety, quality, which are essential for the acceptance of an application system. Therefore, components can be characterised by:

Component = attributes + operations + structural constraints + operational constraints + events + multi-interfaces * scenarios + safety + reliability + security + ... (4)

As shown in Chapter 5.3 and restricted to software architecture, the component paradigm is an appropriate tool for analysis, design, acquirement, and implementation of open, distributed systems. However, the scope of considerations and interpretations must be enlarged. This is especially true, if different views on systems created by different user groups (management, specialised users, system administrators, implementers) have to be modelled. These different views are characterised by different languages used for formalisation and abstraction, by different levels of details (granularity), and by different tools for describing the models used. However, the objectives and the essential properties dealing with the information created, recorded, stored, processed, and transferred are the same. Regarding the RM-ODP schema of views on systems mentioned already, as Enterprise View, Information View, Computational View, Engineering View, and Technology View, all these essential views needed for a precise description of really open systems are provided. In the next section, ways for unification of the different approaches will be discussed, providing legitimation for harmonisation of such different issues as legal, social, ethical, organisational, and technological aspects of security, organisational structures and programs with lines of code. For this purpose, the experiences of system analysis, design, and implementation have been adapted expanding the view abroad the pure software architecture of formal functionality.

4.1.5 Unification of Different Modelling Approaches

The different views of real world systems, concepts, models for analysis and design as well as implementation details and programs are distinguishable by the set of elements describing the systems as well as by the degree of granularity realised to describe the models on each level of abstraction. The processes of state transitions from one level of abstraction to the next one have to be done faithfully mapping the structure of the representations (states) used thus preserving the essential information. Each resulting structure must be defined. Formally, this process can be described deploying the UML methodology as well as using the theories of (recursive) functions, formal languages, and automatons as follows.

The theory of automatons considers behavioural models of systems sufficiently agreeing with the systems themselves, their constraints, properties and objectives. An automaton realises processes independent of direct human interactions but in interrelationship with the human being. Automatons may be described using algorithmic definitions or structural definitions. An abstract automaton can be defined as a septuple

$$A = (I, O, K, \alpha, \omega, \tau, \pi) \qquad (5)$$

with

I: set of inputs,

O: set of outputs,

K: set of configurations,

connected by the functions

α: I→K input function,

ω: K→O output function,

τ: K→K transition function.

π: K→{0,1} is stop predicate (considering the Boolean algebra).

The automaton is running until the stop predicate is fulfilled indicating a final state being achieved.

Figure 4.4 presents the Graph diagram of the abstract automaton described in formula (5).

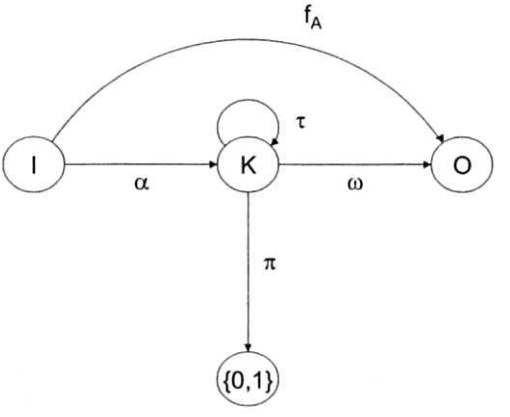


Figure 4.4: Scheme of an Abstract Automaton

The function f_A calculated by the abstract automaton is formally defined as

$$f_A: I \rightarrow O \tag{6}$$

$$f_A(i) = \begin{cases} \omega(\tau^{t_A(i)}(\alpha(i))), \text{ if } t_A(i) \text{ defined} \\ \text{undefined otherwise} \end{cases} \tag{7}$$

The number of configuration transitions (runtime) t_A of the abstract automaton is given as

$$t_A(i) = \min \{m \in N | \pi(\tau^m(\alpha(i))) = 1\} \tag{8}$$

The set of final configurations E is defined as

$$E = \{k \in K \mid \pi(k) = 1\} \tag{9}$$

The model (5) can be transferred into other automaton models appropriately defining its elements. By that way, finite, Moore, Mealy, Medwedjew, non-deterministic, initial, weakly initial, autonomous or combinatorial automatons can be specified. The objective of minimisation (reduction) of an automaton A is to achieve an automaton Amin with the same behaviour but minimum number of states.

To analyse complex systems, the automaton may be approximated by

- investigating autonomous partial behaviour by separation of autonomous automatons with constant input,
- projecting autonomous partial automatons in a single transition state space,
- classifying states or transitions according to specific criteria.

Providing "global states" by semantically specified classification, the last coarseness may be used to exclude the transition between states located, e.g., in the same state class. Thus, the automaton reacts more inert easing the analysis, however remaining the "global state". This can be shown, describing an automaton by the Lagrange equation (10). The related outputs characterise the class tags of the state or transition classes respectively [Wunsch, 1986].

$$L(i,k,k') = 0 \tag{10}$$

k' means the state after a transition. The states k in the original model are represented through new states k* in the roughened model by the function

$$k^* = q(k) \tag{11}$$

Therefore, the behaviour of the roughened automaton can be described by

$$L^*(i,k^*,k^{*\prime}) = 0 \Leftrightarrow \exists(k,k') \, (L(i,k,k') = 0) \tag{12}$$

$$k^* = q(k) \tag{13}$$

$$k^{*\prime} = q(k') \tag{14}$$

With some assumption it could be derived, that the component model changing to another level of granularity keeps the essential properties consistent.

The idea of abstract, encapsulated state automatons was also shortly mentioned in [Selic and Rumbaugh, 1998], however without any theoretical or practical background provided. Equation (5) is identical with a process P defined as a triple

$$P = (S, f, s) \tag{15}$$

with state space S, action function f, and set of initial states. The transition is hereby interpreted as the partial algebra on the set I×O×K. Mostly, processes are described by recursive functions or algorithms.

Demonstrating the basis to harmonise different model using reduction techniques, appropriate characteristics must be found to describe systems and their different states fulfilling the statements made.

Chapter 4.1.4 and Chapter 4.1.5 have specified the properties of component model, which are independent of the underlying technology, very robust and with only a few paradigm-related basic statements and prerequisites. Transferring the specifications to different levels of granularity, the abstraction depending on the component's characteristics seems to be the semantics to appropriately and openly describe the state of the elements using the findings above.

Each level of presentation may be represented by a set of abstractions characterising the underlying concept or paradigm respectively. An abstraction itself can be defined as a recursive set of qualities representing attributes or values. Due to the recursiveness, qualities may themselves be abstractions containing other qualities and so on. Adding qualities to an abstraction (or to existing qualities) provides a new abstraction representing another level of presentation due to the specialisation happening.

Based on these specifications, equation (15) can be refined to

$$P = (S, f, a, g) \tag{16}$$

with the initial state vectors abstraction *a*, characterising the paradigm (or domain) reflecting concepts, and granularity *g*, specifying the level of refinement within the current paradigm framework. For clarification, the relationships will be explained using a simplified example as well as the developed generic component model, presented in the next section and derived from the results achieved here. Keeping *a* constant, the increased granularity g moves the model (current process state) from a business domain to a business process and finally to a workflow level within the business components abstraction level. Regarding the same relationship within the logical components domain, the process state transition occurs, e.g., from program systems via programs to program modules and code lines. Keeping *g* constant, the change of *a* causes a process state transition from business components to (software) design components or furthermore to program modules and their technical implementations.

As shown, the state transition[9] process represented by the abstraction evolution allows for an unified approach the system modelling independent from the level of granularity and the related elements defined at this stage, like organisational entities, components, objects or structures and modules respectively (see also [Port, 1998]).

4.2 A Generic Model of Component Systems

As mentioned already in the section before, component systems allow analysis and design of distributed, interoperable, and scaleable systems from the systems theory „black box" point of view. The structure of individual components needs not to be known. They can be specified using object-oriented or non-object-oriented paradigms. However, a protocol must explicitly specify the contractual agreement about the detailed behaviour, pre- and post-conditions as well as error management. Communicating and co-operating components can be coupled tightly or loosely. Tight coupling involves implemented long-time connections, whereas loose coupling is a short-time and status-depending coupling (e.g. messaging systems as EDI). Recently, several papers have been published analysing and systematising legacy and newly developed systems with respect to their distribution, scalability, and co-operation from the system's point of view instead of the software one [Saleck, 1997a,b] supporting our last section considerations. Based on all these efforts mentioned, a generic and more consistent model has been developed describing the system-related component paradigm. Implementing scalability and interoperability, a common view and a development template of component systems is needed which describes the general relationships of component systems being consistent with the theoretical consideration of recursive functions and abstract automatons made.

Considering the system view, the last chapter's definitions of a *component* and a *process* have to be combined also structurally elucidating the relationships of components in respect to their static and dynamic behaviour in the context of defining, implementing, managing, and using such components. Implementing a process in the sense of its instantiation, the service specified can be offered. It will be provided by invocation via interfaces. Defining some of the terms used in the chapter, the understanding of the generic component system model will be improved.

A *process* is a set of *attributes* as well as *methods* and their *management* respectively. An *instance* is the concrete realisation of a process. Attributes represent the state or information belonging to a process. Methods are functionality, algorithms, or services in the sense of implemented objects associated with a process. The management defines controlling rules or algorithms of a process used for information issues. *Courses* or *services* in the sense of implemented processes are a complex functionality provided by one or more processes. They describe the complex state and behaviour of those processes. *Interfaces* provide an access method to a process. Interfaces can be grouped into *frameworks*. A *strategy* rules services and courses respectively by the definition of goals and the algorithms to its achievement.

Intensionally viewed, processes are classified into *types*. Different types can be connected through a *subtype* relationship. Modules, components, and businesses can be aggregated by subtype relationships.

Comprehensively expressed, a basic component system consists of processes which are associated with attributes and methods or, more generally, with information issues and their management respectively. Interfaces enable the access to implemented processes provided

[9] Talking about state transitions doesn't necessarily mean that a specific component changes its state but that the meta-model describes different views of different specific components like states in a consistent way.

for services following a strategy. Figure 4.5 represents this general structural scheme describing the component statics.

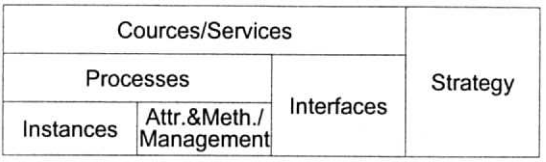


Figure 4.5: General scheme of components architecture

The component paradigm abstracts from the underlying software development methodology and even from traditional systems technology and inter-system relationships. Different component views as enterprise-related components (organisational components, business components), logical components, or technological components, are characterised by different abstraction levels and consist of the component architecture at all levels of granularity from micro components up to macro components. Therefore, neglecting the recursive character of abstractions describing the underlying concept, the granularity vector and the abstraction vector brace up the state space of the components as shown in Figure 4.6.

The composition/decomposition in the sense of component aggregation can only be performed related to a specific parameter, if this parameter remains consistent during the state transition as shown in the former chapter. In that context, the ISO RM-ODP related ISO standard General Relationship Model (GRM) has to be mentioned [ISO 10165-7]. This GRM specifies relationships between business processes containing pre- and post-conditions as well as relationships between business functions not containing pre- and post-conditions. One practically important facit from the GRM is the exploration of useless functional relationships between business processes not changing the post-conditions in comparison with the pre-ones.

The composition of components enables the invocation of services through the component's interface, which arise from the associations between the sub-components. Therefore, this functionality cannot be addressed at the sub-component level.

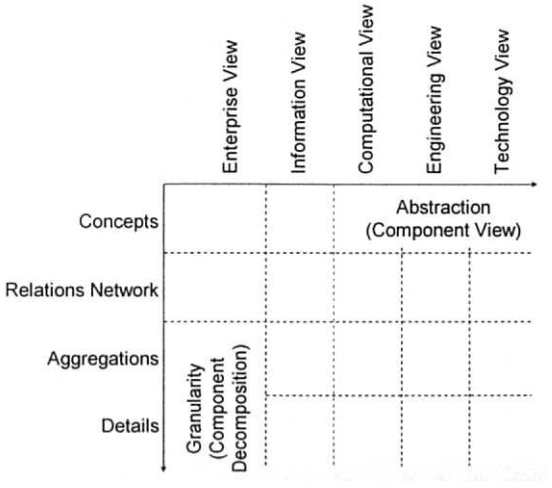


Figure 4.6: Discrete Component State Space Braced up by the Granularity and the Abstraction Vector

Considering the modelling paradigms widely used, the discrete states possible to be adapted by the components can be grouped on the one hand into concepts, relations networks, aggregations, and details at the granularity dimension, and on the other hand into the business, the logical, and the technological view on components at the abstraction dimension (Figure 4.6). As discussed above, also other levels of abstraction and granularity may be specified. The resulting state matrix may be described like Figure 4.7 defining the components like the models usually deployed.

Enterprise	Applications	Computer Networks
Business Domain	Program System	Computer Clusters
Business Process	Program	Computer, Network Nodes
Workflow	Modules, Routines	Computer Components

Figure 4.7: Component State Matrix

As a general architectural scheme, this model pertains to different aggregation levels. Hence, it includes also vertical facilities of the management defining business objects, and interfaces to build up a business process. Component systems enabling the implementation of the general components architecture at different levels of system granularity provide higher scalability and interoperability in a distributed environment. Different mechanisms of component actions correspond to different levels of communication between components (Table 4.2) [Saleck, 1997b].

Thus, the compound architectural scheme of the considered component systems can be used to describe and to evaluate distributed co-operating systems including information systems. The resulting compound component system does not need to belong to a single architectural concept (middleware approach), i.e., there could be a merge of different architectures on different levels (Table 4.2).

Table 4.2: Communication levels of components [Saleck, 1997b]

Level		Properties
7	Request Broker with load distribution	Optimised co-operative data processing
6	Request Broker	Standardised co-operative data processing
5	Client/Server	Shared data processing
4	Process call, Dynamic Link Library (DLL)	Co-operation of autonomous programs
3	Module call, library functions	Different sources with common loader
2	Macros, copying functions	Copy of external elements to the source
1	Subroutine call, local procedure call	Internally of the source

The generic meta-model presented in this paper is based upon a conceptual scheme (meta-model of a meta-model) as shown in Figure 4.8. Currently, a categorical model abstracting from semantics of the ER-model is under development.

Regarding the hierarchical component structure of complex, distributed, interoperable middleware concepts, there are differences with respect to how consequently the generic model is applied at the different levels of granularity, as well as with respect to the communication solutions (properties) provided at the different communication levels. EDI (not considering

any additional functionality of communication servers, which are not part of the EDI standard) allows only calls at different levels. DHE provides a client/server-type co-operation, not implementing lower level services. Only CORBA covers most of the considered levels of granularity, for newly developed applications ideally meeting the structural principles. Also a sufficient efficiency and performance including load distribution is supposed to be achievable for CORBA soon. On the other hand, however, CORBA as well as the other approaches discussed are depending on the technological basis ignoring the other levels of abstraction. Therefore, they facilitate design and implementation, but not domain description and analysis. This relation of the different advanced approaches for distributed health information systems' architectures to the ISO RM-ODP is given in Figure 4.9.

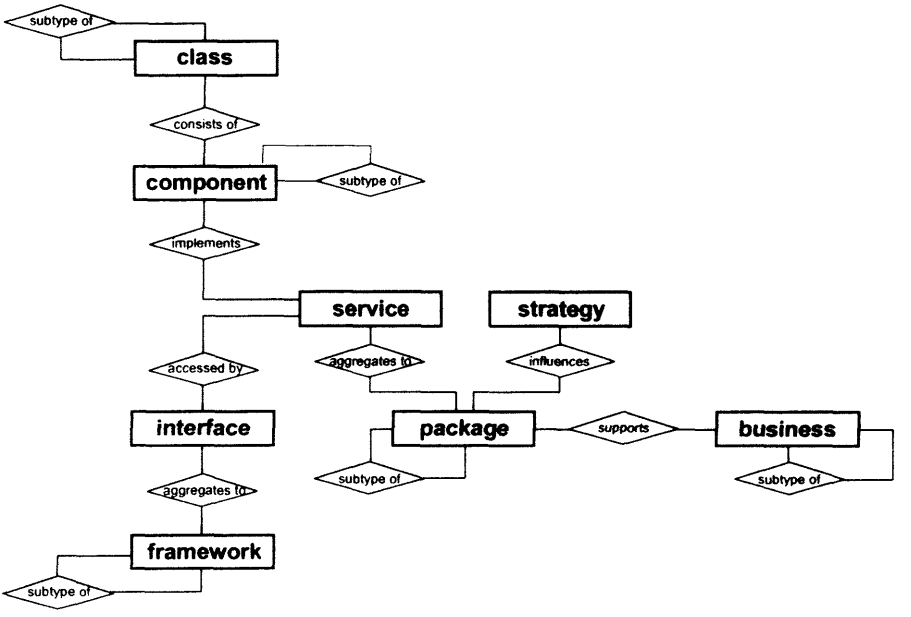


Figure 4.8: Basic concepts of component architectures

4.3 Summary and Conclusions

For a quantitative comparison, evaluation, and harmonisation a component based analysis and design is helpful using a meta-model independent of the approaches compared. To evaluate the common middleware approaches for health information systems such as HL7, DHE, and CORBA, the component paradigm has been analysed, interpreted and adapted systematically. The paradigm's suitability for consistently harmonising and comparing different architectural approaches for health information systems as well as different views of different user groups involved has been demonstrated using the theories of recursive functions and abstract automatons. Such investigation is especially important when deploying the underlying models for such complex issues as security and its legal, social, ethical, organisational, and technological aspects. A generic component model has been developed for system analysis, system design, and system assessment. On each different level of aggregation, the components consist of objects or aggregated objects (business objects), interfaces or management systems, providing appropriate services or courses. By such way, the components enable an architecture following a given strategy.

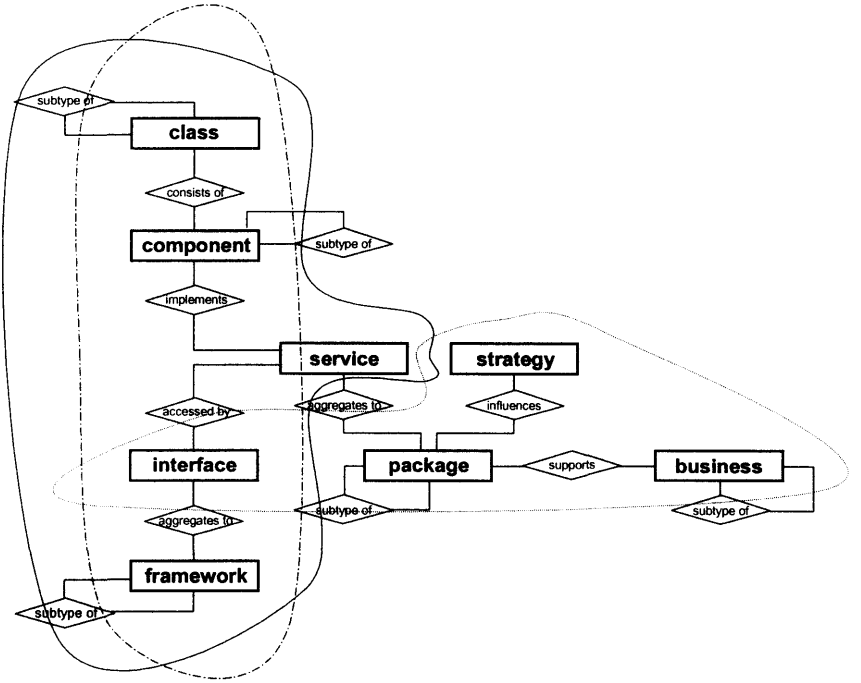


Figure 4.9: Middleware approaches reflected at the generic component model schema (— original CORBA, -- HL7 V2.x & early V3, ···· DHE)

Components abstract from underlying mechanism, methods, and domains specialised, considering the black box behaviour of the components only. Therefore, component models can be developed for organisational, logical, and technological domains. On the different aggregation levels from system details up to concepts, different communication levels can be considered characterised by different properties. Regarding the diverse middleware concepts, the provided services can be connected with corresponding aggregation levels. On each aggregation level, different communication levels with their properties can be defined, on this way characterising possible abilities for interoperability and scalability of distributed systems.

Reflecting the generic component model and its ISO RM-ODP relations, the views of the RM-ODP (abstraction levels of the generic component model) define very generic constraint models describing specific aspects, concepts, and knowledge about a system.

5 The Electronic Healthcare Record in the Architectural Context

5.1 Introduction

For establishing efficient and high quality care of patients, comprehensive and accurate information about status and processes directly and indirectly related to patient's health must be provided and managed. Such information concerns medical observations, ward procedures, laboratory results, medical controlling, account management and billing, materials, pharmacy, etc. Therefore, health information systems within healthcare establishments (HCE) converge to Electronic Patient Record (EPR) systems as a kernel enabling the management of all other business processes as specific views on the EPR and building the informational basis for any communication and co-operation within, and between, HCE. So, inter-organisational virtual electronic healthcare records (EHCR)[10] are built. Introducing an EHR is a long and stony way which cannot be gone in one step. The internationally acknowledged US Medical Record Institute located in Newton, MA, defined a step by step approach for establishing EHR as shown in Figure 5.1.

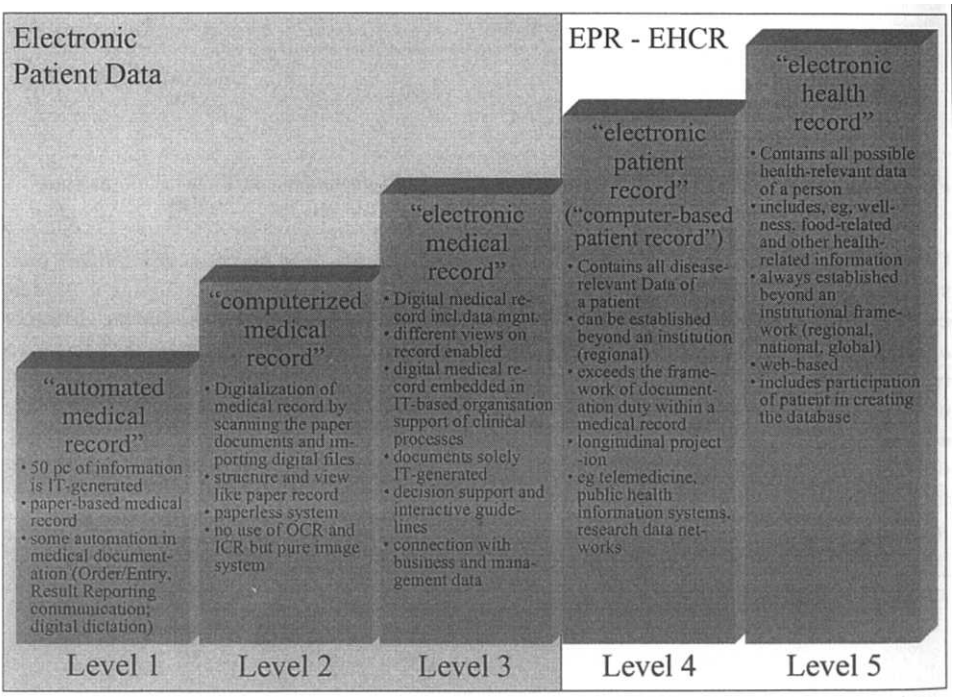


Figure 5.1: EHCR Development Levels according to Medical Record Institute [MedRecInst_WWW]

This virtual EHCR has to meet shared care requirements of providing any information needed and permitted at the right time to the authorised user at any location in the right format including mobile devices. In that context, it has to fulfil all needs of the HCE and its

[10] If the record includes also issues beyond patient's care such as, e.g., social aspects and health prevention of citizens, an electronic health record (EHR) is created. In the paper, the EHCR view will be used knowing the validity of the statements also for EHR, however.

principals involved reflecting all views defined in ISO/IEC RM-ODP introduced already in Chapter 3.5. These views are different in different HCE with their different scenarios for meeting different requirements under their specific conditions and constraints. Constraints are expressed on structure, names, and values. For providing information and functionality needed, EHCR must be structured and operating appropriately. Furthermore, the EHR system has to comply with the ISO GRM (see Chapter 4.2).

There is the need to create the technological, architectural, methodological, and functional framework required for supporting nowadays business challenges enabled by corresponding new types of information systems. Additionally to this technical consideration, also legal and ethical as well as strategic aspects have to be mentioned. These issues are referred to the security-related chapters of this book, however.

Considering health information systems' evolution, the important characteristics of such systems are distribution and interoperability, scalability and flexibility, feasibility of change management regarding properties, trustworthiness, etc.

The way to provide such system behaviour is the orientation on objects or, even better, on components, service orientation, analysis, design, implementation, and maintenance based on modelling methodology, Web orientation, and the use of nowadays means to formulate and manage the specifications.

Currently, Web services and related standards are under development not always offering the maturity needed to be considered the basis for wide, large scale, and perhaps even critical applications.

5.1.1 EHR-Related Definitions

First, some definitions related to EHCR should be introduced:

An EHCR is a repository of information about the patient's health available in a computer-readable format.

An EHCR system is a set of components establishing mechanisms to generate, use, store and retrieve an EPR.

The EHCR architecture describes a model of generic properties required for any EPR for providing communicable, comprehensive, useful, effective, and legally binding records, which preserve their integrity over the time independent of platforms and systems as well as of national specialities.

Following, the basic EHR requirements will be shortly introduced, ignoring the minor differences between EHCR and EHR but treating them synonymously instead. Afterwards, different approaches and an optimal way for meeting the aforementioned requirements and characteristics will be described in more detail focusing on the architectural and modelling aspects of EHCR.

5.1.2 EHR Requirements

Any information system must be built in a way to satisfy user requirements and to meet user expectations. Therefore, the analysis of user requirements on EHR must be performed very carefully. In this section, only some basic requirements will be discussed. For more details, the reader is referred to the excellent work performed by the CHIME Department of the University College of London within the EHCR SupA project of the European Commission [CHIME_WWW].

Because the EHR reflects all information directly or indirectly related to patient's health, which is expressed using any thinkable data types, precision, dimension of consideration and knowledge (see SNOMED dimensions or axes), it has to be adaptable to any progress,

development, techniques, etc. occurring during patient's life and beyond. Therefore, basic requirements for EHR to be managed are, e.g.,

- long time maintenance of information,
- extreme size of domain as well as rate and extension of changes,
- need for sharing information regarding both content (terms, quantities, signals) and structure (concepts, architectural components, messages structures) at knowledge level,
- appropriate security and privacy features.

The instances of domain knowledge to be shared are, e.g.,

- concepts,
- extracts,
- queries, responses, reports.

Principle solutions for fulfilling these requirements are

- an implementation-independent methodology provided by model-driven design based on platform independent models,
- interoperability at knowledge level,

which are practically achieved by

- change resistance of software and information by openness and harmonisation of methodology and approaches,
- harmonisation of syntax, semantics, and exchange format,
- harmonisation of infrastructure including security infrastructure.

Furthermore, the context of information created and stored in the EHR must be preserved. The edition of new contexts must be enabled. Regarding the range of contexts, compositional context, data value context, qualifier context, ethical and legal context, care process context should be mentioned. For more detail see [CHIME_WWW].

5.1.3 EHR – A Document or a Service?

Depending on the scope's scale, the information system unit used could concern one, several, or all views of the ISO RM-ODP by that way representing data, objects, or components including all aspects of dealing with them. The resulting difference of the approaches is the level of interoperability ranging from data level interoperability through knowledge-level interoperability up to service-oriented ones.

Referring to the principles dealt with in Chapter 2, legacy systems meanwhile realise the highest level internally. Transferring them into an open environment, the feasibility moves quickly down to the simplest level of stupid data exchange, however.

As already mentioned in one of the first German papers published on HL7 issues, most approaches are under continuous improvement providing a kind of harmonisation between currently occurring differences [Blobel, 1993]. The following sections discuss actual and emerging approaches for EHR architectures and systems, at the moment

- reflecting only the information view of RM-ODP and therefore belonging to the document-oriented EHR approach, or
- referring to a more or less comprehensive view on information systems belonging to the service-oriented approach. Thereby, the mentioned weaknesses related to the existing EHR approaches will be overcome.

Because the document-orientation as well as the content and structure of exchangeable information is strongly connected to the XML Standard set, a short introduction into the XML stuff will be given which will be used in the most of the EHR approaches.

5.1.4 The XML Standard Set

XML is a very robust meta-language for specifying content and structure of documents, their presentation or format as well as for messaging defining a data exchange format. If the semantics in the old HL7 exchange format can be explored by the sequence (position) of elements using delimiters, the markup languages use tags for structuring and labelling elements. The roots of XML are the Standard Generalised Markup Language (SGML) and Hypertext Markup Language (HTML), the language of the Web.

Because HL7 gives a good example of using both types of exchange format representation, sequence-oriented and tag-oriented structuring of an HL7 message are shown in the following figures.

```
OBX|1|NM|9804-6^Weight^LN||135|lb|||||F
```

Figure 5.2: Sequence-Oriented Structuring of an HL7 OBX Segment

```
<ORM_O01>
  ...
  <OBX>
    <OBX.1>1</OBX.1>
    <OBX.2>NM</OBX.2>
    <OBX.3>
      <CE.1>9804-6</CE.1>
      <CE.2>Weight</CE.2>
      <CE.3>LN</CE.3>
    </OBX.3>
    <OBX.5>135</OBX.5>
    <OBX.6><CE.1>lb</CE.1></OBX.6>
    <OBX.11>F</OBX.11>
  </OBX>
  ...
</ORM_O01>
```

Figure 5.3: Tag-Oriented Structuring of an HL7 OBX Message

```
<ORM_O01>
  ... <OBX>
    <OBX.1 LongName='Set ID - OBX' Type='SI' Item='569'>1</OBX.1>
    <OBX.2 Table='125' LongName='Value Type' Type='ID'
      Item='570'>NM</OBX.2>
    <OBX.3 LongName='Observation Identifier' Type='CE' Item='571'>
      <CE.1 LongName='identifier' Type='ST'>9804-6</CE.1>
      <CE.2 LongName='text' Type='ST'>Weight</CE.2>
      <CE.3 LongName='name of coding system' Type='ST'>LN</CE.3>
    </OBX.3>
    <OBX.5 LongName='Observation Value' Type='WILDCARD'
      Item='573'>135</OBX.5>
    <OBX.6 LongName='Units' Type='CE' Item='574'>
      <CE.1 LongName='identifier' Type='ST'>lb</CE.1>
    </OBX.6>
    <OBX.11 Table='85' LongName='Observation Result Status' Type='ID'
      Item='579'>F</OBX.11>
  </OBX> ...
</ORM_O01>
```

Figure 5.4: Extended Tag-Oriented Structuring of an HL7 OBX Message

An XML well formed document has exactly one root element. Every sub-element including recursive sub-elements has delimiting start tags and end tags. The elements are properly nested within each other. DTDs[11] define well formed XML documents. An XML Schema on the other hand is a model describing the structure of information for a whole class of documents. Because it is expressed in XML, it is a document describing the valid format of an XML data set. This definition includes ([Stuart, 2001], extended)

- elements that are allowed and elements that are not allowed at any point,
- attributes for any element,
- the number of occurrences of elements, etc.

A valid document is well formed and conforms to a specified set of production rules. Using proper tools, XML Schemata can be checked.

Actually, the XML standard set consists of a growing number of specifications mostly defined by the World Wide Web Consortium (W3C) [W3C_WWW], e.g.:

- Namespace: Defines the use of namespaces to avoid name clashes when working with documents from multiple sources
- XML Information Set: Establishes a kind of an "HMD" for an XML instance document
- XML Schema Definitions: Overcomes the insufficiency of DTDs. The specification is performed using the XML Schema Definition Language XSD
- XML Query Language: Serves to extract information
- XML Digital Signature, XML Dsig: Defines digital signatures in XML
- XML Encryption Specification: Ensures confidentiality of XML messages
- XML Key Management Specification, XKMS: Shields XML application developers from the complexity of traditional PKI implementations.
- XML Path Language, XPath : Addresses parts of XML documents
- Canonical XML: Provides a method for generating the canonical form
- Extensible Stylesheet Language, XSL, XSL Transformation, XSLT: Transformation which enables the conversion from one document class to another and facilitates adapted, re-usable presentation
- XML Metadata Interchange, XMI: Enables the bridging of different presentation languages as well as vocabulary generation at meta-data/meta-model level
- XLink, Xpointer: Supports linking between documents
- Document Object Model, DOM: Specifies a tree-based API for parsed documents
- XML Topic Map, XTM: Provides relations between topics, occurrences, and associations
- Simple API for XML, SAX: Specifies an event/stream-based API for parsed XML documents
- XML Event: Enables the observation of, and the rreaction on, events (see Chapter 12)
- XAML/SAML: XML-based markup languages for defining transaction security (see Chapter 12)

5.1.4.1 XML DTDs

Inherited from SGML, the Document Type Definition (DTD) defines the legal building blocks of XML or any other SGML-based document. DTDs are used to define content

[11] Document Type Definitions

models as valid order and nesting of elements and, to a limited extend, datatypes and attributes. DTDs have important pros such as widespread tools support, widespread deployment (e.g. DTD definitions in HTML, XHTML, DocBook, TEI, J2008, CALS) as well as widespread expertise in practical use for many years. However, there are important cons against DTDs causing the movement towards XML Schema. In that context, the use of another (non-XML) syntax for specifying DTDs, missing of namespace support, the limitation in expressivity, the extreme limitation in datatyping (10 types) only offering a few coarse string formats and explicit enumeration for attribute datatypes, the limited and hardly manageable extension mechanism, and finally, the missing feasibility of making relationships explicit have to be mentioned.

An example for specifying a well-formed XML is demonstrated using simple radiology report document (Figure 5.5).

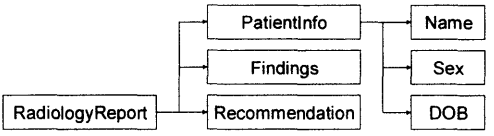


Figure 5.5: Structure of a simple Radiology Report (after [Heitmann, 2001])

Using the given document structure, Figure 5.6 presents the corresponding DTD.

```
<!ELEMENT RadiologyReport
  (PatientInfo, Findings, Recommendation)>

<!ELEMENT PatientInfo (Name, Sex, DOB)>

<!ELEMENT  Name (#PCDATA)>
<!ELEMENT  Sex (#PCDATA)>
<!ATTRIBUTE Sex Table CDATA #REQUIRED>
<!ELEMENT  DOB (#PCDATA)>

<!ELEMENT Findings (#PCDATA)>
<!ELEMENT Recommendation (#PCDATA)>
```

Figure 5.6: DTD for the Given Radiology Report ((after [Heitmann, 2001])

5.1.4.2 XML Schemata

In schemata, models are described in terms of constraints. Two kinds of constraints can be defined: content model constraints and datatype constraints. Content model constraints describe order and sequence of elements. Datatype constraints describe valid units of data.

Remembering the introduction of components in Chapter 4.1, things are characterised by data, operations, Therefore, XML Schemata can be used to describe classes of components. Because these components can reflect different views regarding their levels of abstraction as well as different levels of granularity, XML schemata will be deployed in the following chapters for representing the taxonomy of the components in their respective states. By that way, our mid-nineties generic component model got another confirmation in the actual development of the openEHR architecture (see Chapter 5.3.2).

The W3C XML Schema specification consists of three parts: the Primer, XML Schema Structures, and XML Datatypes.

Regarding datatypes, the XML Schema specification defines boolean, number, date and time, URI, integer, decimal number, real number, interval of times, and many more. Other datatypes and aggregate types can be created [Walsh, 2001]. Meanwhile, more than 45 datatypes have been specified.

For our component based health information system analysis and design purposes, another apparently unlimited set of datatype is essential: user defined types, also called *Archetypes*. These archetypes can be refined, so representing inheritance behaviour. Specifying fact by constraints for data interpreting them, concepts and knowledge can be formulated. So, the logical and pragmatic aspect of information may be addressed.

Other features offered by the XML Schema specification are grouping attributes and the namespace support. By means of attribute grouping, relationships between attributes as part of the expressed concept can be made explicit.

Figure 5.7 presents the XML Schema for the radiology report introduced in the last section.

```
<element name="RadiologyReport">
 <complexType>
  <sequence>
   <element ref="PatientInfo"
    minOccurs="1" maxOccurs="1" >
   <element ref="Findings"
    minOccurs="0" maxOccurs="unbounded" >
   <element ref="Recommendation"
    minOccurs="1" maxOccurs="1" >
  </sequence>
 </complexType>
</element>
```

```
<element name="PatientInfo">
 <complexType>
  <sequence>
   <element ref="Name" minOccurs="1" >
   <element ref="Sex">
   <element ref="DOB"
    minOccurs="0" maxOccurs="1" >
  </sequence>
 </complexType>
</element>

<element name="Name" type="string">
<element name="Sex" type="string">
<element name="DOB" type="date">
```

Figure 5.7: XML Schema for a Radiology Report (after [Heitmann, 2001])

5.2 Principles of Existing EHR Approaches

Replacing the old relation paradigm of some architecture models for health information systems such as, e.g., the Distributed Healthcare Environment (DHE) architecture, the actual EHR architecture standard models follow the object-oriented or even component-oriented paradigm. However, they are distinguished by a fundamental difference in their approach of establishing the EHR model. One single group intends to develop the complete EHCR architecture within one comprehensive model of structures, functions, and terminology in the classic way covering all the concepts known at the development time. Such one model approach, however, reveals some essential weaknesses and problems related to technical, complexity, and management issues which are now shortly resumed [Beale, 2001].

Considering the technical problems of the one model approach, the mixture of generic and domain-specific knowledge concepts with their own expressions, but also weaknesses in basis class stability must be mentioned.

Regarding the complexity problems, the size of the resulting model leads to difficulties in managing so many concepts in parallel, in completing the model which might be unachievable, in standardising such models and in providing interoperability due to the needed agreement on a huge number of aspects and details.

Related to the management of the one model approach, different developer and user groups dealing with their own concepts expressed in their specific language must be managed, combined and harmonised. The generic part of the EHR concepts concerns the grammar of the IT system domain which is specified by computer scientists. The health domain specific concepts representing the domain knowledge are specified and maintained by medical experts. Both groups are characterised by their own terminology and their specific way of thinking. The dependency of both groups results from the fact that there is only one common development process using the same formalism.

The other group provides a dual model approach establishing a generic object or component model and a set of specialised models reflecting organisational, functional, operational,

contextual, and policy requirements presenting the knowledge about the detailed circumstances of practical EHCR instances overcoming the one model approach's problems.

An example of the first group is the CEN ENV 13606 "Electronic Healthcare Record Communication". HL7's version 3 models and the Australian GEHR approach belong to the second group, despite of the differences explained in detail in the next chapters.

5.3 Examples of the EHR One Model Approach

5.3.1 The European Standards' Approach for Electronic Healthcare Record Extended Architectures

In its Part 1, the CEN ENV 13606 "EHCR Communication" defines an extended component-based EHCR reference architecture [CEN ENV 13606]. Such an extended architecture is mandated to meet any requirement throughout the EHCR's complete lifecycle. According to CEN ENV 13606, an EHCR comprises on the one hand a *Root Architectural Component* and on the other hand a *Record Component* established by *Original Component Complexes* (OCC), *Selected Component Complexes*, *Data Items*, and *Link Items*. OCC consist of 4 basic components, such as folders, compositions, headed sections, and clusters. These OCC sub-components can be combined in partially recursive way. Beside its Part 1 "Extended Architecture", the CEN ENV 13606 offers Part 2 "Domain Term List", Part 3 "Distribution Rules", and Part 4 "Messages for the Exchange of Information". The CEN ENV 13606 follows the one model approach.

5.3.2 The Governmental Computerised Patient Record

Launched by a consortium formed by the US Department of Defense, the US Department of Veterans Affairs, and the Indian Health Service, the Governmental Computerised Patient Record (G-CPR) established a model and tools for implementing and managing a proper business as well as technical environment to share patient's information [G-CPR_WWW]. The main goals concern

- the establishment of a secure technical environment for sharing sensitive personal information,
- the development of a patient focused national information technology architecture,
- the creation of a common information model and adequate terminology models to ensure interoperability between disparate systems.

The solution should be based on advanced national and international standards. Using object-oriented specifications for interoperability, the approach was service-oriented rather than architecture based.

5.4 Examples of the EHR Dual Model Approach

5.4.1 The Recent HL7 Approach on Electronic Healthcare Record

5.4.1.1 HL7's Paradigm and Architecture Changes

Responding to the development of the Internet, e-Business and e-Health as well as reflecting newer evolutions and revolutions of ICT, HL7 changed its paradigm, methodology and architecture slowly but continuously and fundamentally as well. HL7's nowadays Version 3 objective is to provide a framework for coupling events, data elements and messages, to improve clarity and precision of specification as well as the adaptability of standards to change, and finally to begin to approach "plug and play".

This is done by bringing modern software engineering practices such as Object-Oriented Analysis and Design and formal modeling using UML to the standards development process in a better yet not consistent way. In that context, HL7 aims to bring the highest level of quality, understandability, and flexibility to a messaging standard to compete with other SDOs which intent to follow similar approaches.

Elements of the HL7 Version 3 standard are:

- *Use Case Models* as hierarchies of tasks and actors;
- *Interaction Models* describing trigger events, abstract messages & application profiles;
- *Information Models* such as the RIM specifying generic classes, relations, and core attributes;
- Message design models such as *Domain Information Models* (DIM), also called *Refined Message Information Models* (R-MIMs), and *State Transition Diagrams*, specifying domain-specific classes, relationships, states, and lifecycles, but also *Abstract Message Definitions* (HMDs) for instantiating the concrete message;
- *Vocabulary* with sophisticated domain definitions, representations and mappings;
- *Implementation Technology Specification* (ITS).

Vocabulary issues are out of the scope of this book. Following, the other HL7 components are described in some more details.

5.4.1.2 HL7 Reference Information Model

Within its Version 3 *Message Development Framework*, the well known health industry standard for communication HL7 specified a comprehensive *Reference Information Model* (RIM) covering any information in the healthcare domain in a generic and comprehensive way [HL7_WWW]. During its evolutionary development process which is still ongoing, the HL7 RIM as a crucial issue of the HL7 paradigm changes from an entity centred to an act centred view.

The HL7 RIM deals with the associations between the six core classes *entity* (physical information object in the healthcare domain), the *role* the entity can play (competence for action), *participation* (performance of action), the *act* as well as *role relationship* mediating interaction between entities in the appropriate roles and *act relationship* for chaining different activities. The HL7 RIM core classes imply six kinds of attributes: Type_CD (Class_CD), CD, Time, Mood (determiner), Status, ID. Figure 5.8 presents the RIM core classes, attributes, and core attribute values. Obviously, the core classes *role* and *participation* are specialisations of the related *entities*. In that context, *roles* realise competence-related specialisations but *participations* act-related specialisations.

For specifying HL7 Version 3 messages, the communication and co-operation scenarios are described using UML *Use Case Diagrams* and *Story Boards*. *State Transition Diagrams* define the domain-specific pieces belonging to the use cases.

Afterwards, corresponding classes, their specialisations and their associations forming a part of an HL7 message are presented as a domain-specific information model. These models are called *Message Element Types* (METs). Standardising these models as a LEGO®-type building elements, a set of *Common Message Element Types* (CMETs) is provided. Using special HL7 tools, these graphical specifications can be transferred into HMDs and finally translated into XML-based HL7 Version 3 messages. The CMETs are an essential issue of the HL7 standard. They can be developed step-by-step, updated, and replaced easily. Another specification of domain-specific basis components using verbal instead of graphical formalisations are the HL7 *Clinical Templates* which belong to the *Vocabulary* concept. The CMET ↔ *Cinical Template* relation is similar to that of an *Archetype Model*

and the *Archetype Schema* which is used in the other dual or multi-model approaches discussed below.

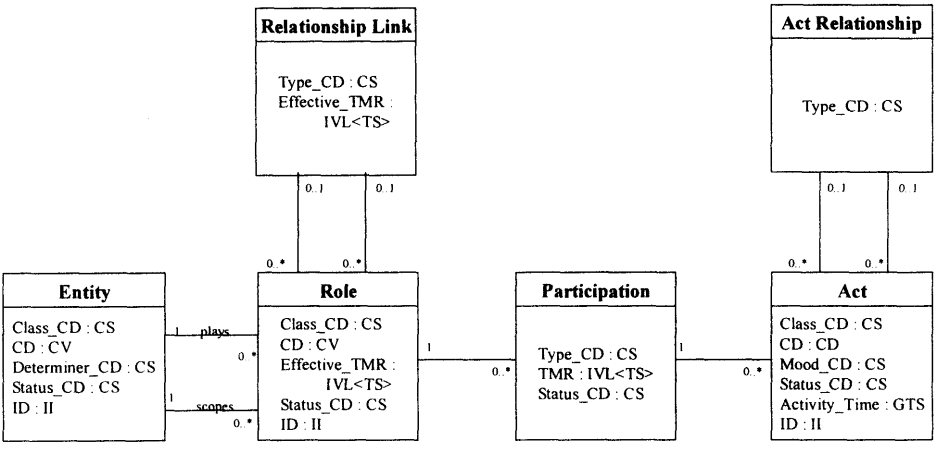


Figure 5.8: HL7 RIM Core Classes, Core Attributes, and Core Attribute Value Sets

With its RIM and its CMETs, HL7 moved from the original one model approach to a dual model approach. Domain-specific concepts and knowledge can be described consistently deploying the RIM and an object-oriented UML-based methodology. So, HL7 Version 3 enables interoperability at knowledge level. Examples for *Story Boards*, *State Transition Diagrams*, CMETs and HMDs in an authorisation context are given in Chapter 6.13.

HL7's RIM and vocabulary provide domain knowledge which is exploitable, e.g., for knowledge representation (representation of concepts and relations) in the GEHR Object Model and archetypes discussed below.

The specialised model for *Clinical Document Architecture* (CDA) has been specified for developing appropriate messages to support EHR communications. It is based on the generic RIM and its refinements as *Refined Message Information Model* (R-MIM) and *Common Message Element Types* (CMET) for EHR-related scenarios. It establishes a dual model approach analogous to the GEHR approach.

The HL7 approach reflects solely the information viewpoint of ISO RM-ODP and provides step by step some recent associations to the computational as well as to the business viewpoint. Within information models, it describes classes, attributes and their specialisations for developing messages. Therefore, HL7 provides interoperability at data level but not at functional level. Following, some more details are given to shortly introduce HL7's CDA.

5.4.1.3 HL7 Clinical Document Architecture

After starting some activities to specify a *Patient Record Architecture* (PRA) which would violate HL7's traditional messaging paradigm, related activities have been turned towards the *Clinical Document Architecture*. Defining a document structure being transmitted as a message, the original HL7 approach could be met easier. A clinical document is a documentation of clinical observations and services, characterised by persistence, stewardship, potential for authentication, wholeness, and human readability. These scopes will be shortly explained as follows.

A *persistent* clinical document continues to exist in an unaltered state, for a time period defined by local and regulatory requirements.

Stewardship means that a clinical document is maintained by a person or organisation entrusted with its care.

A clinical document is an assemblage of information that is intended to be legally *authenticated*.

Authentication of a clinical document applies to the whole and does not apply to portions of the document without the full context of the document.

A clinical document is human readable.

5.4.1.4 CDA Levels

The CDA specification is a step by step approach. Therefore, first a simple container has been introduced which will be refined in the future.

CDA Level One is the root of the hierarchy and is the most general document specification. RIM classes are used in the specification of the document header, while the document body is largely structural, although terms from controlled vocabulary can be applied.

CDA Level Two will be a specialisation of CDA Level One, and will constrain the set of allowable structures and semantics based on document type code.

CDA Level Three will be a specialisation of CDA Level Two that will specify the markup of clinical content to the extent that it can be expressed in the HL7 RIM.

Figure 5.9 shows a CDA hierarchy example.

CDA Level One
CDA Level Two
Level Two :: Progress Note
Level Two :: Cardiology Progress Note
Level Two :: Endocrinology Progress Note
Level Two :: Diabetes Mellitus Progress Note
CDA Level Three
Level Three :: Progress Note
Level Three :: Cardiology Progress Note
Level Three :: Endocrinology Progress Note
Level Three :: Diabetes Mellitus Progress Note

Figure 5.9: Example for the CDA Hierarchy

5.4.2 The Australian Good Electronic Health Record Project

5.4.2.1 The GEHR Object Model

Based on the European Commission's Third Framework Programme project "Good European Health Record (GEHR)", but also acknowledging the results of other R&D projects and efforts for standards around the globe, the Australian Government launched and funded the Good Electronic Health Record (GEHR) project [GEHR_WWW]. The basic challenge towards GEHR is knowledge level interoperability.

The GEHR model consists of two parts: the *GEHR Object Model* (GOM), also called reference model, delivering the EHCR information container needed on the one hand, and the GEHR meta-models for expressing the clinical content on the other hand (Figure 5.10).

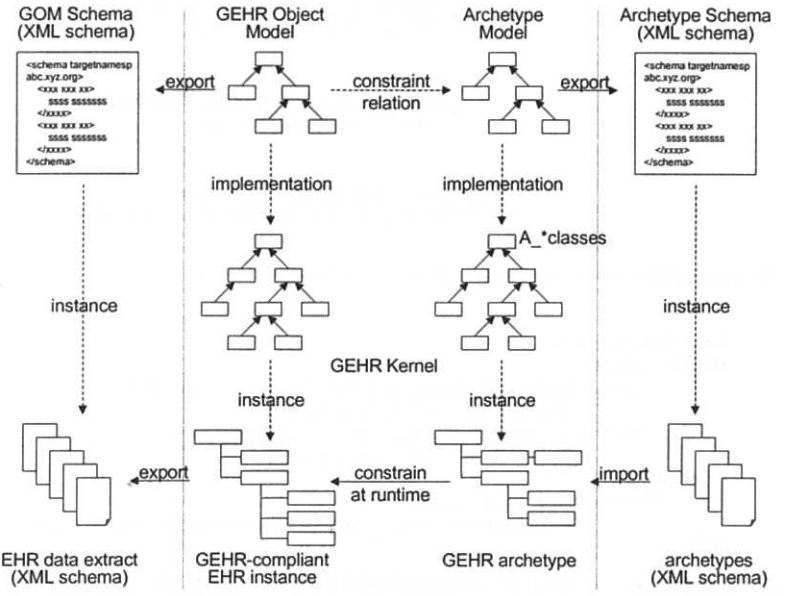


Figure 5.10: GEHR Architectural Schema (after T. Beale [Beale, 2001])

Bearing the medical knowledge in the sense of healthcare speciality-specific or the organisation-specific, department-specific or even person-specific views and constraints, the meta-models are commonly called *Archetypes*. An archetype constitutes a formal model of a domain concept easily understandable by a domain expert. As introduced in Chapter 5.1.4 already, the archetypes describe user defined schemata which can be expressed as user defined XML schemata. Therefore, the corresponding model is also called *Archetype Model* and the schema *Archetype Schema*. Because the archetypes are separately developed, they can be instantiated step by step at the technical model level until the complete medical ontology has been specified. In summary, the GEHR approach consists of small flexible pieces like LEGO® bricks which can be combined in a proper, health domain specific way following construction plans defined in archetypes. Summarily, the reference model is the concrete model from which software can be built, and of which EHR data are instances. The archetype model establishes the formalism whose instances are domain concepts which are directly processable by health information systems.

5.4.2.2 GEHR Archetypes

According to the generic component model approach for health information systems, archetypes can be applied to specify user defined schemata for the different levels of abstraction and granularity reflecting the corresponding conditions, content, and constraints. These archetypes or schemata are provided by the experts of that special domain or the users of the respective views. Reflecting our generic component model and its ISO RM-ODP relations, it should be reminded that the views of the RM-ODP (abstraction levels) define the components' properties generalising the GEHR archetype concept. Figure 5.11 demonstrates a simple Blood Pressure model as an archetype example. Figure 5.12 shows its corresponding declarative expression.

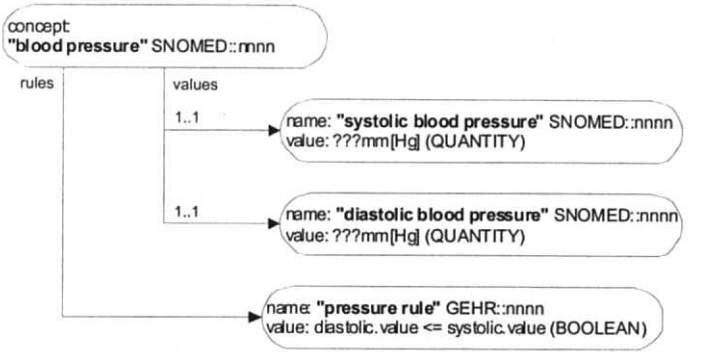


Figure 5.11: Simple Blood Pressure Model (after [Beale, 2001])

```
class "blood pressure"
feature -- values
        "systolic blood pressure" [SNOWMED term nnnnn]: QUANTITY
                ensure: Result units = UNIT mm [Hg]
        "diastolic blood pressure" [SNOWMED term nnnnn]: QUANTITY
                ensure: Result units = UNIT mm [Hg]
invariant
        "pressure rule" [GEHR term nnnnn]:
                "diastolic blood pressure".value <= "systolic blood pressure".value
end
```

Figure 5.12: Declarative Expression of the Simple Blood Pressure Model (after [Beale, 2001])

Applying the generic component model principles, the archetype models can be changed transferring them towards other levels of abstraction and/or granularity. In that context, the models can be refined as shown for the Blood Pressure concept (Figure 5.13).

The next step is the transformation of constraint models expressed graphically into constraint models expressed in an information description language like XML. As a result, instances derived from such constraint models must represent valid XML documents as defined in Chapter 5.1.4. The challenge consists in the appropriate description of the constraints mentioned in the Blood Pressure Concept figures. Regarding those constraints, we have to manage formal constraints and such ones related to special Blood Pressure Concept rules reflecting the domain knowledge like the relation between components of the generic component model. By that way, the names and the cardinality of elements or attribute, but also patterns or facets of datatypes have to be specified. After introducing the terms of the content such as "diastolic" and "systolic", their value restriction to positive integer numbers between, e.g., 0 and 300, and the basic requirement of always pairwise specification of blood pressure, the components of the protocol such as device, position, and cuff size must be specified. Thereby, the restrictions of the latters' types to allowed values such as "sitting" or "standing" for positions as well as "wide" or "narrow" for cuff size must be introduced. Next, the (diastolic blood pressure) < (systolic blood pressure) logic of the component's relation established in medical knowledge must be formulated. Because the current W3C XML Schema Specification enables only fixed patterns or facets compared to fixed values, and thus does not support the expression of relations between two values, the blood pressure rule must be specified deploying the annotation of application information. This step enables the binding of a specific term and its value to the related content specification. Afterwards, the resulting content specifications can be applied in the blood pressure relation constraint rule expressed using the W3C XPath Specification [W3C_WWW].

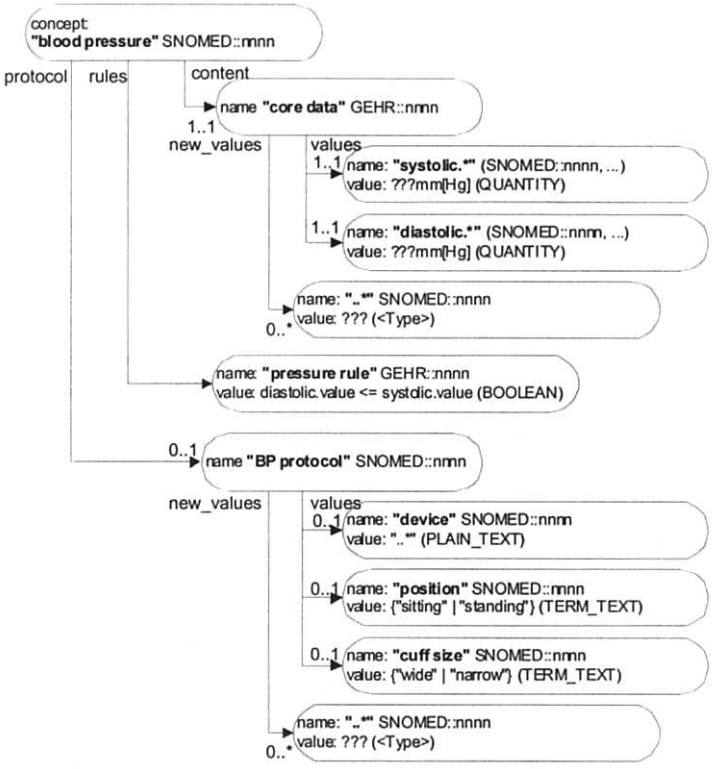


Figure 5.13: Refined Model of Blood Pressure (after [Beale, 2001])

The next figures present examples how to express the refined GEHR Blood Pressure Concept as XML Specification based constraints models using both Document Type Definitions (DTDs) and the above explained XML Schemata to realise an XML instance like that one shown in Figure 5.14.

```
---GEHR.XML
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XML Spy v4.3 (http://www.xmlspy.com)-->
<?xml-stylesheet type="text/xsl" href="H:\3schemas\gehr\Gehr.xsl"?>
<BloodPressureConcept
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="gehrexpl.xsd">
    <Content>
        <Term>systolic</Term>
        <Value>120</Value>
        <UnitOfMeasurement>mmHg</UnitOfMeasurement>
    </Content>
    <Content>
        <Term>diastolic</Term>
        <Value>80</Value>
        <UnitOfMeasurement>mmHg</UnitOfMeasurement>
    </Content>
    <Protocol>
        <Device>xxx</Device>
        <Position>sitting</Position>
        <CuffSize>wide</CuffSize>
    </Protocol>
</BloodPressureConcept>
```

Figure 5.14: XML Instance of the Refined Blood Pressure Concept

Figure 5.15 shows the XML Schema of the refined Blood Pressure Concept with a related XML Stylesheet (Figure 5.16).

```
--GEHR.XSD
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="BloodPressureConcept">
        <xsd:annotation>
            <xsd:appinfo>
                <rp:rule>Content[1]/Term/text()='systolic' and Content[2]/Term/text()='diastolic'</rp:rule>
                <rp:rule>Content[2]/Value > Content[1]/Value</rp:rule>
            </xsd:appinfo>
        </xsd:annotation>
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="Content" minOccurs="2" maxOccurs="2"/>
                <xsd:element ref="Protocol"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="Content">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Term">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:enumeration value="diastolic"/>
                            <xsd:enumeration value="systolic"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
                <xsd:element name="Value">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:integer">
                            <xsd:minInclusive value="0"/>
                            <xsd:maxInclusive value="300"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
                <xsd:element name="UnitOfMeasurement"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="Protocol">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Device" type="xsd:string"/>
                <xsd:element name="Position">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:enumeration value="sitting"/>
                            <xsd:enumeration value="standing"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
                <xsd:element name="CuffSize">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:enumeration value="wide"/>
                            <xsd:enumeration value="narrow"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Figure 5.15: XML Schema of the Refined Blood Pressure Concept

```
---GEHR.XSL
<?xml version='1.0' encoding="iso-8859-1" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" indent="yes" version="4.01" doctype-public="-//W3C//DTD HTML 4.01//EN"/>
    <xsl:template match="/BloodPressureConcept">
        <html>
            <head/>
            <body>
      <xsl:text>
          Testing rules:
      </xsl:text>
    <xsl:if test="not(Content[1]/Term/text()='systolic')">
      <xsl:text>
          - no Term systolic
      </xsl:text>
      </xsl:if>
    <xsl:if test="not(Content[2]/Term/text()='diastolic')">
      <xsl:text>
          - no Term diastolic
      </xsl:text>
      </xsl:if>
    <xsl:if test="not(Content[1]/Term/text()='systolic' and Content[2]/Term/text()='diastolic')">
      <xsl:text>
          - not both Terms diastolic and systolic
      </xsl:text>
      </xsl:if>
    <xsl:if test="Content[2]/Value > Content[1]/Value">
      <xsl:text>
          - diastolic value greater than systolic value
      </xsl:text>
      </xsl:if>
        </body>
          </html>
      </xsl:template>
</xsl:stylesheet>
```

Figure 5.16: XML Stylesheet for Processing the Blood Pressure Concept Rules

Figure 5.17 presents the DTD for the refined Blood Pressure Concept. Comparing the DTD specifications given in the XML Schema context, the weakness of DTD regarding data types is obvious. Especially the needs of constraint languages expressing rules and logics properly cannot be met sufficiently. The value constraints are still missing in the figure.

```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT BloodPressureConcept (Content+, Protocol)>
<!ATTLIST BloodPressureConcept
        rules CDATA #FIXED "Content[1]/Term/text()='systolic' and
        Content[2]/Term/text()='diastolic';Content[2]/Value > Content[1]/Value">

<!ELEMENT Content (Term, Value, UnitOfMeasurement)>
<!ELEMENT Term (#PCDATA)>
<!ELEMENT Value (#PCDATA)>
<!ELEMENT UnitOfMeasurement (#PCDATA)>

<!ELEMENT Protocol (Device, Position, CuffSize)>
<!ELEMENT Device (#PCDATA)>
<!ELEMENT Position (#PCDATA)>
<!ELEMENT CuffSize (#PCDATA)>
```

Figure 5.17: DTD of the refined Blood Pressure Concept

Even XML Schema will get some extension in the future to enable more elegant solutions which might be based on XPath specifications.

An interesting solution could occur by the current activities of openOASIS regarding the harmonisation and improvement of IT standards [openOASIS_WWW]. Beside the Interoperability Summit efforts provided by many SDOs including HR-XML Consortium [HR-XML_WWW], OMG [OMG_WWW], OASIS [OASIS_WWW], XBRL [XBRL_WWW], and others, openOASIS' engagement for global XML specifications has to be mentioned.

In that context, RELAX NG Specification as a simple schema language for XML can be promising as an alternative to the W3C schemata discussed (see e.g. [Vlist, 2002]).

The integration of the different models used is provided by the basic properties of objects or components such as overriding and inheritance.

5.4.2.3 The openEHR Project

Based on the Australian GEHR project and supported by the joint engagement within the revision of CEN ENV 13606 "Health Informatics – EHR Communication", the global openEHR initiative has been established in 2001.

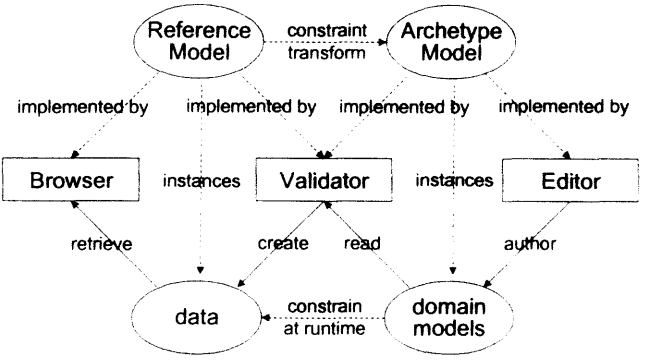


Figure 5.18: Meta-Architecture for Implementing and Use of OpenEHR

Regarding implementation and use of the openEHR approach, the needed model components and system components can be presented as shown in Figure 5.18. Based on the models introduced in Chapter 5.4.2, editors are needed to author the domain knowledge in domain models. Read by the openEHR kernel, this information is used to create the object model instances, i.e., the data the principal is interested in to be retrieved and presented by a browser.

5.4.3 OpenEHR Package Structure

For implementing openEHR, several system components or packages have to be established. The EHR basic structure is the *Record*. Its sub-packages describe the compositional structure of an EHR. The *Record* package contains the packages *EHR* (incl. EHR extracts), *Transactions* (incl. audit trail), and the related content. The latter contains the *Navigation*, *Entry*, and *Data* packages, whose classes describe the structure and semantics of the contents of transactions in the health record. The Entry package contains the *Structure* package which itself concerns, e.g., the *Representation*. The *EHR Extract* package addresses the EHR class describing EHR Extract semantics but also related services. The *Path* serves for item location. Because it uses URI links, it is sometimes also called *Link*. The basic package defines the core classes used in the openEHR approach. *External* refers to external packages providing interoperability with non-EHR systems via identification of principals involved such as users, systems, components, devices, etc.organisational issues, parties, etc., but also of terminologies. The *Archetype* (sometimes called *Basic*) package addresses the concept representation with its core classes locatable and archetyped.

Figure 5.19 presents the package structure of an openEHR system as described [Beale, 2001].

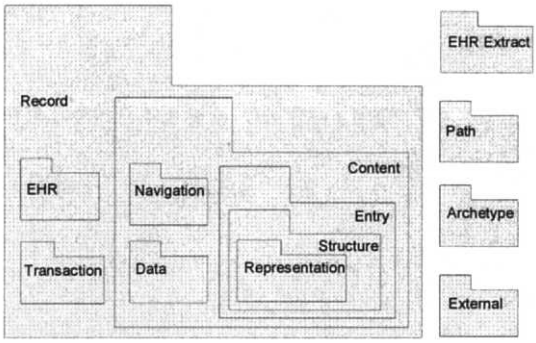


Figure 5.19: Package Structure of an openEHR System [Beale, 2001]

5.4.4 EHCR/EHR Architecture Model Harmonisation and Emerging Projects

Establishing formal and informal liaisons, organisations engaged in EHCR or EHR specification and implementation intend to improve the existing standards. In that context, several activities have to be mentioned especially such as

- the recently started revision of CEN ENV 13606 now called "Electronic Health Record Communication". According to the CEN rules, CEN TC 251 had to deal with ENV 13606 in 2002 again either to confirm, to revise, or to reject this standard. The decision was made to form a task force the author is prominently involved in to revise the standard fundamentally. After analysing the international ongoing EHR-related activities, the dual model approach as well as an open specification procedure including non-European groups has been decided.
- the refinement of G-CPR in the sense of emphasising HL7 communication instead of CORBA service orientation,
- the establishment of the European Commission's EuroRec organisation, and
- the openEHR approach.

Collaborating with HL7, both CEN and openEHR will narrow and harmonise their approaches. Establishing an (initially funded) national EuroRec organisation in all European Union member states, the EuroRec initiative concerns the improvement of awareness for, and the wider implementation of, EHR in the European practice. In that context, a European Electronic Health Record institute has been founded in November 2001.

5.5 CORBA 3 Component Architecture

CORBA is a well-known approach for an architecture oriented on distributed services. Implementing the clear three tiers architecture for the Internet with refinement of the three layers into clients, type-specific servers, content management, business logic, data layer, the client is served with services managed by a server. Recognising the same difficulties in CORBA 1 and CORBA 2 (Chapter 3.2) leading to our development of the generic component model in the mid-nineties as presented in Chapter 4, the CORBA community is now working on CORBA 3. The main objective of the new approach is to meet the challenge of e-Business in the Internet world by enhancing CORBA towards the *CORBA Component Model* (CCM) and the *Model Driven Architecture* (MDA).

Figure 5.20 demonstrates the CORBA basis architecture referring to its architectural components including newer developments. As most of the components presented have been explained in Chapter 3.2, the details about the *Portable Object Adapter* (POA) replacing the *Basic Object Adapter* (BOA) will follow soon.

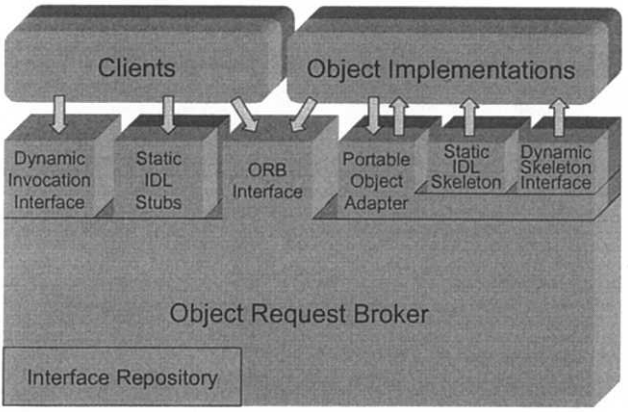


Figure 5.20: CORBA Architectural Model (after [Siegel, 2001])

The foundations of the *CORBA Component Model* (CCM) are

- the *valuetype* concept,
- the *Portable Object Adapter* (POA),
- the *Persistent State Service* (PSS).

Following, these CCM basics will be described in a bit more detailed manner, mainly based on John Siegel's excellent writings, e.g. in [Siegel, 2001].

5.5.1 CORBA Valuetypes

A *valuetype* is a programming language object. It gives programmers an alternative construct for passing by value – instead of the original CORBA object feature of passing by reference. A *valuetype* exists only while the object is running. When its reference count goes to zero, both itself and its state vanish. A *valuetype* is designed to externalise its state in a form usable by the same or another vendor's ORB. The valuetype specification narrows CORBA and Java, by that way supporting Java-to-IDL mapping as well as the use of EJBs as basic-level CORBA components. Furthermore, the valuetype allows to present an XML document as a tree of valuetypes enabling XML/Value mapping and it enables asynchronous invocations. In the latter case, polling agents return *valuetype*. So, the *valuetype* is one pillar offering CORBA and Java a convenient way for implementing Internet-based applications to facilitate any kind of e-Business.

5.5.2 CORBA Persistent State Service

CORBA 2 has already specified an Interoperable Object Reference (IOR) also used in the Security Service Specification (Chapter 8.2), but it even needed object localisation for preserving persistence of the object, however. For guaranteeing performance and availability of services at growing but still restricted bandwidth of the Internet, objects (and corresponding services as their implementation result) should be kept active only if in use. For continuing a process on the other hand, the invoking object has to find the original object state to perform its business. The PSS preserves an object's state from one activation to the next. The PSS is required to internalise an object's state so that it can be restored by the same service only. PSS is not part of CORBA itself, but a CORBAservice.

The *Persistent State Service* automates storage and retrieval of a servant's persistent state which will be shortly introduced in the next section. Two modes for programming PSS are available:

- Using the *Persistent State Definition Language* (PSDL) defined by the specification;
- Declaration of the object's state directly in the programming language which is called transparent persistence.

Generally speaking, the PSS enables the CORBA 3 open approach replacing the original object feature by a CORBAservice.

5.5.3 CORBA Portable Object Adapter

Remember from Chapter 3.2 that an object adapter is the mechanism that connects a request using an object reference with the proper code to service that request. Using tailored object adapters, a specific flexible object behaviour defined by its correspondingly adapted business logic can be offered to the client. Such service enables new horizons for specifying, implementing and maintaining new generations of distributed intelligent information systems based on components and introduced in Chapter 12. Starting with the *Portable Object Adapter* (POA), the relevant CORBA components and services to do this will be discussed first, however.

The POA standardises the interface and operations the server uses to interact with the object implementation for resource control. For optimising resource usage, the server does not establish a permanent association between the code that serves a request (called a POA servant), and the CORBA object reference that represents the object to the client. The POA replaces the Basic Object Adapter (BOA) as mentioned already.

A POA definition consists of three pieces: The object reference, a mechanism that connects a request, and finally a code to serve the request. The object reference is established by an address, the name of the POA that created the object reference as well as an object ID, all invisible to the client.

A servant is a code that contains the business logic of an object. More precisely, a servant contains the methods for a CORBA object, where a method is defined in CORBA as the programming language code that implements an operation defined in an IDL interface. A servant is normally written by a system programmer.

In OO languages, a servant is an instance of a class, which is a declaration entity in those languages. Creating a servant in these languages requires knowing the declaration for the class and then using the language function "new" on the class name. Since a servant implements the operations of an IDL interface, it contains computational language entities in its class corresponding to operations on the IDL interface. The computational entity for Java is called a class method, and for C++ it's called a member function.

Getting closer to the servant, the IDL compiler next generates a skeleton, also called the servant base class – a class declaration and code that contains interface-specific details for runtime use on the server. A server programmer uses the servant base class in two ways: First, the servant base class code is compiled and linked into the server executable binary; like stubs, the programmer doesn't look at or modify this code. Then, after the IDL compiler has provided the servant base class declaration, the programmer codes the servant class, inheriting all methods required for the object from the servant base class declaration and providing the code for them.

ServantBase is a class definition and code provided by an ORB vendor; it serves as the base class for all servants. The compiler-generated servant base class inherits from ServantBase; the real servant programmed by the programmer inherits from the servant base class. At runtime, the class definition of servant is made into an instance of a servant (something that can be executed) by the programming language new function.

Remember that the POA is an object. It is created, has an object reference, is invoked and is destroyed. Because a POA is locally constrained (contrary to other objects), the POA object

reference makes sense only to the ORB on the server on which the POA was created. The POA reference cannot be passed to any other computer. It supports navigation through a distributed system. The POA is part of the implementation of an object. The implementation of an object is the combination of a POA and a servant. The POA is a stateful object. Its behaviour follows specific policies.

POA policies control the specific kind of object reference, routing, lifetime of the object and object reference. They define the assignment as well as the use of the object ID and control the permission of taking part in transactions. Furthermore, the POA policies control creation, registering, use, and destruction of the servants. Therefore, the following POA policies have to be considered: the POA LifeSpanPolicy {TRANSIENT|PERSISTENT}, the POA RequestProcessingPolicy, the POA IdAssignmentPolicy, and the POA ServerRetentionPolicy.

Regarding the use of CORBA specifications for EHR architecture purposes, the POA enables the flexible behaviour of components as needed for interrelations between components reflecting different views. It results from the fact that in addition to policies assigned at POA creation, a POA has also dynamically specifyable optional behaviours for its administration. Finally, there are some POA features that are available primarily for convenience.

POA can be used by explicit object activation or as single servant for all objects. In the latter case, on-demand activation for the duration of a single method and on-demand activation for indefinite duration may be distinguished.

5.5.4 CORBA Component Model

The CCM packages up services such as persistence, transactions, security, and event handling, along with POA's server-side resource handling capability into a single, standard development and run-time environment. Because it pre-selects service configuration for that server environment, the CCM is able to present to the programmer service APIs at a much higher level than those of the services themselves. Therefore, a CCM application can be made transactional or secure by adding a single line to a configuration file, without changing a single line of language code. This enables to move the server programming work from system programmers to business domain experts, who can tailor the server to sophisticated business algorithms more precisely and respond to new business opportunities [Siegel, 2001].

The advantage of moving from the CORBA object paradigm towards the CCM is that CCM applications are very compact, modular, and easier to code. For implementing a CCM, the *Component Implementation Definition Language* (CIDL) is used. CCM applications scale to enterprise and Internet usage level. The CCM Container connects to an implementation of the PSS to provide persistence.

As predefined in the generic component model already at mid-nineties, small components can be grouped into assemblies and larger components can be divided into segments. Therefore, the optimal size of a component depends on performance requirements rather than on reusability constraints. Analogue to CORBA objects, CORBA components are created deploying component factories. Thereby, clients invoke the factory to create their own component instance. So, an application contains at any instant only exactly the number of specific components needed.

Components can be written in different categories, depending on how long they and their object references are expected to last, whether or not they have persistent state, and how this state is exposed to the client. These have names like service, session, entity, and process.

Writing down special declarations, code and functionality will be generated automatically. At runtime, the factories create component instances on-the-fly as needed. The ability to create, activate, deactivate, and destroy components according to the actual needs is the key to the CCM's scalability.

Each component type (not instance) has its own ComponentHome type-kind class of an object for the type. ComponentHomes bear lifecycle interfaces for their type: create, find, (for entity components only), and remove. The ComponentHome types must also be declared in the IDL file with the result that IDL and code for ComponentHome operations are generated automatically from a simple declaration.

A component is not a CORBA object with super powers; instead, it is a new CORBA metatype that supports or provides interfaces that you define separately in your IDL. This means that all interfaces must be pre-declared before declaring their components. Afterwards, the components can be declared using the new IDL keyword component. Summarising, for each component, the interfaces they support or provide as well as a ComponentHome have to be declared.

5.5.5 Model Driven Architecture

During its evolution, the CORBA approach moved from the strict object paradigm via the interoperability protocols towards a component orientation and *Model-Driven Architecture* (MDA). This way is characterised by the generalisation of the underlying concepts but also by opening the environment including other conditions, constraints, often also called platform. Obviously, the term platform is not restricted to hardware, but includes any characteristics in technology, organisation, function, etc. which classifies and therefore distinguishes a platform from another one. By that way, other platforms could be enabled until abstraction and openness finally allows platform independent specifications.

A *Platform-Independent Model* (PIM) is defined in UML, although other notations are allowed "where appropriate". Additional description – behaviour and constraints, primarily – can be defined using either UML or natural language.

For defining the syntax for a *Platform-Specific Model* (PSM) two ways are used. The first deploys UML diagrams using an officially adapted platform-specific profile. The second way uses interface definitions in a concrete implementation technology (OMG, IDL, XMI, Java). In both cases, additional description – behaviour and constraints, primarily – may be specified in either UML or natural language.

5.6 Comparison of the Advanced EHR Approaches

For specifying and implementing successful architectures, systematic analytical comparison of the different approaches, their harmonisation as well as a generic improvement has to be performed. This procedure follows partially the evolutionary and partially the revolutionary paradigm. Such way protects investments, enables stepwise improvements, preserves the knowledge accumulated, and allows defining the next generation sometimes even clearing intermediate versions. In that context, a certain paradigm change can be observed in EHR standardisation initiatives.

Standards move from harmonising different solutions enabling compatibility, flexibility, interoperability, and portability towards the specification of future requirements and solutions. Standards don't have extended lifetime anymore but being changed after a more or less short time. Therefore, standards change towards higher-level specifications defining principles and paradigms rather than implementation details. Introducing components, fragmented solutions allow the replacement of outdated parts. The complexity of definitions is followed by references to available specifications as much as possible. Such refer-

ences also include specification elaborated by non-accredited standard developing organisations (SDO) such as the W3C.

5.6.1 Common Features of the EHR Approaches Presented

Characterising the EHR architectural approaches presented, some features are in common and others are different. Common properties of the EHR approaches presented are the object-orientation and the model-based development framework. For dealing with concepts, contexts and knowledge expressed as constraints, most of the approaches changed towards the component paradigm. This is true for approaches at the services level, but not for approaches following the message paradigm. If constraints are managed, actual specifications deploy XML schemata to express them properly. If some solutions offer openness and platform independence (e.g. CEN ENV 13606, GEHR, openEHR as well as the generic component paradigm), some others refer at least to certain architectural basics (CORBA requires an ORB and not all services are really compatible and portable).

CORBA's MDA offers an interesting way for modelling and implementing component-based systems. It will be deployed to specify, implement as well as maintain EHR architectures and systems. Starting from platform independent models and platform independent domain models dealing with the domain specific knowledge expressed as constraints and relations, platform specific models as well as platform specific domain models are derived from the unspecific ones. In the projects GEHR and openEHR, unspecific models are called object models. Specific models have to meet constraints. Models representing constraints in content, datatypes, procedures, etc., are called archetypes. Therefore, specific models have been called archetype models. Object model and archetype models can be derived from complex models or vice versa. Instantiation of archetype models and object model are bound together at runtime. The archetypes concern any concepts such as medical specialities, organisational restrictions or personal peculiarities. So, different concepts within one view (abstraction level) or related to different views on a system can be specified. Combined with other CORBAservices and specifications, the POA enables such behaviour. For binding the models in an unambiguous way, digital signature mechanisms might be deployed as shown in Chapter 12.

5.6.2 Missing Features

With the exception of CORBA's MDA, all other EHR architectural approaches presented deal with just one or a restricted number of views of the ISO RM-ODP. For example, HL7 is restricted to the business (use cases, CMETs) and the informational (RIM) view, GEHR adds some aspects of the computational view, G-CPR is a service-oriented approach which moves towards a message paradigm restricted to the inheriting approaches. Therefore, the approaches leave gaps in the system's lifecycle. Regarding the challenge of portability as well as the support for implementing the solutions, corresponding features are sadly missed.

5.6.3 Harmonisation Platform

The generic component model developed by the author in the mid-nineties seems to comprise the positive features of the different approaches and adds some important aspects missed so far. It offers a harmonisation platform which serves as the theoretical and methodological basis for the emerging projects the author is involved in such as the CEN ENV 13606 revision.

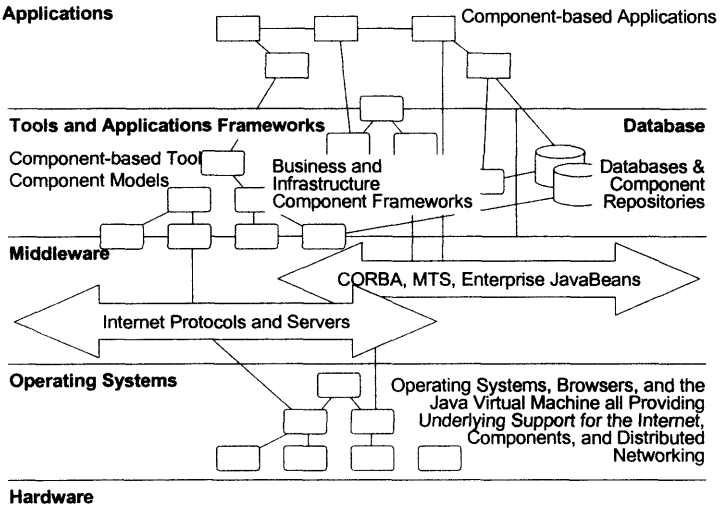


Figure 5.21: The Distributed Computing Architecture Elements (after [Cutter, 1999])

Figure 5.21 presents the distributed computing architecture elements proposed by the Cutter Consortium [Cutter, 1999]. The different elements defined are components to be modelled and implemented. At all abstraction and granularity levels (layers) represented by component models, object models and constraint models have to be developed. By that way, the application component's object model and the corresponding archetype models, but also the related models of all the other architecture elements have to be specified.

5.7 Summary and Conclusions

Starting with the Medical Record Institute EHR development schema, definitions and requirements around the EHR are introduced in this chapter. Reflecting the evolutionary way of EHRs from documents to services, the XML standard set is presented, also mentioning XML capabilities far beyond documents. Due to the importance of XML schemata as constraint models, the XML schema specification is especially considered and compared with XML DTDs.

Existing EHR approaches are compared using the Generic Component Model described in the previous chapter as well as the ISO RM-ODP introduced in Chapter 3. Also the CORBA 3 MDA is referred to as a type of taxonomy. The reference system is characterised by a reference model, several constraint models reflecting the different domains' knowledge, and the RM-ODP views for all those models.

The assessed EHR approaches comprise the CEN ENV 13606 "EHCR Communication" and the US G-CPR as examples for a one model approach as well as HL7 Version 3 and the Australian GEHR project as examples for a dual model approach. Table 5.1 summarises some of the evaluation results.

Resulting from this comparative evaluation of the most relevant EHR specifications, the chapter concludes with a harmonised EHR model with references to CORBA3 and HARP. While emerging projects proceeding in that direction are mentioned, missing features are presented and a future-proof architecture discussed in Chapter 12 is prepared.

Table 5.1: Main Characteristics of the Main EHR Approaches

	ENV 13606	G-CPR	HL7/ CDA	GEHR/ openEHR	CORBA3	HARP
Business view supported	x	x[12]	x	x	(x)[13]	x
Information view supported	x	x[10]	x	x	(x)[11]	x
Computational view supported	x	x[10]	x	x	(x)[11]	x
Engineering view supported			(x)[14]		(x)[11]	x
Technology view supported					(x)[11]	x
Reference model defined	x[15]	x[13]	x	x[14]		x
Health domain models defined	x[13]	x[13]	x	x		x
Terminology defined	x		x	x		x[16]
Methodology defined		x	x		x	x
Specification tools available			x			x
Implementation tools available						x

[12] Originally service-oriented
[13] Possible according to the defined methodology
[14] To a certain degree, ITS may fulfil this requirement
[15] One model approach combining both challenges
[16] Imports available terminologies

6 A Systematic Approach for Secure Health Information Systems

6.1 Introduction

Security and privacy contain political, legal, social, organisational, and technical issues everybody is talking about. Therefore, many organisations and institutions deal with different orientation, competence, efficiency and efficacy with this important challenge. As a result, incalculable papers and books have been written, many standards and recommendations are available. In that context, the ISO/IEC Standard 17799 "Information technology — Code of practice for information security management" [ISO/IEC 17799], but also its roots, the US TCSEC and the European ITSEC specifications [EC, 1991] as well as the ISO/IEC IS 15408 "Information Technology – Security Evaluation Criteria" (also known as Common Criteria) have to be emphasised [ISO/IEC 15408]. All these specifications are domain independent, however. Therefore, this book mainly focuses on health-specific work and documentation.

As introduced in Chapter 1, the *shared care* paradigm is the commonly accepted answer to the challenge for efficient and high quality healthcare systems which is caused by the political, societal and economic constraints. It must be supported by appropriate information systems architectures as healthcare networks, distributed Electronic Health Care Record (EHCR) systems, etc. Dealing with sensitive, personal medical data, such information systems have to meet comprehensive security requirements to respond to threats and risks in distributed health information systems. Regarding security in general, we have to look for security, safety and quality concepts [Laske, 1995]. To keep the approach feasible, the consideration is restricted on the concept of security only.

6.2 Security Threats and Risks

Information systems are always exposed to threats influencing their intended objectives, behaviour, and functionality. The possibility that such threats happen establishes risks to principals involved depending on the probability and the results of the threats (changing or damaging systems and issues, legal responsibilities, liabilities, lost of image, money, etc.).

Threats occur either by accident (errors) or with intent (attacks). *Active* and *passive* attacks may be distinguished depending on whether or not attackers stimulate or influence their victims before evaluating their behaviour. Of course, active and passive attacks can be combined in any way and any order.

A threat model summarises all threats that might occur disturbing or disabling a secure communication and co-operation of systems.

Trust models define expectations and requirements the principals have for using the system as well as communicating and co-operating with other principals in an acceptable way.

Given the (security) specification and the trust model (or alternatively a threat model) of all stakeholders of a system, the system is called secure if, from each stakeholder's point of view, it meets his specification even if all (untrustworthy) parts of the system are under attack (according to the threat model). For more details on the mentioned aspects, see, e.g., [SEISMED, 1996].

Risk in the context of IT security is defined [EC, 1991] as an aggregate of

- the likelihood of something untoward happening, i.e., the likelihood of a threat actually occurring,

- the degree of ability to cope with "the happening", i.e., the vulnerability to a threat if it did occur, and
- the resultant consequences if "it" did happen.

The risks faced by a real system are largely determined by the social and economic context in which it is run and by the security it provides. The impact of social and economic factors can be limited by adequate codes of conduct and security policies [SEISMED, 1996] whereas the security of a system can be increased by appropriate technical countermeasures. We call a system *trustworthy* if its risk is acceptable in a certain sense for the participants working with it. Naturally, risk and trustworthiness are subjective matters that have to be cultivated constantly.

6.3 Methods

For a systematic and open analysis, design, and implementation of security services and mechanisms in shared care information systems, within the ISHTAR as well as the MEDSEC projects [ISHTAR_WWW; MEDSEC_WWW] we have developed some models based on components of different levels of abstraction and granularity. For formal and comprehensive descriptions, an agreed or even standardised methodology is inevitable. The popular object-oriented and component-oriented paradigms, extended as shown in Chapter 4, as well as the further development and harmonisation of the corresponding tools for analysis, design and implementation based on the Unified Modeling Language (e.g., [Eriksson and Penker, 1998]) provide an open and comprehensive solution to respond to these challenges. Establishing objects as well as components as the basis for information systems, newly developed and legacy systems may be handled alike.

6.4 The General Conceptual Security Model

Looking for security in health information systems, we have to respond to two kinds of challenges. Firstly, based on the general security model (Figure 6.1) published in several papers, e.g. [Blobel, 1996b; Blobel et al., 1997], the security services needed to protect such systems from the security threats and risks they are exposed to must be specified.

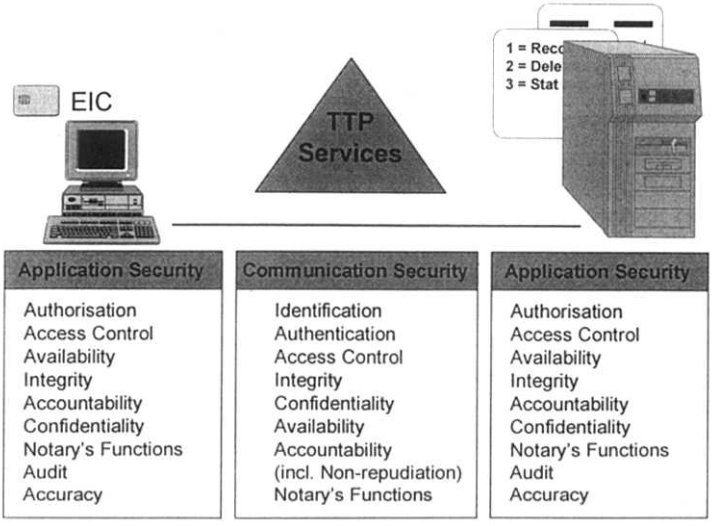


Figure 6.1: General Security Model (EIC = Electronic Identity Card, TTP = Trusted Third Party)

The model distinguishes between the concepts communication security and application security. Communication security concerns the services identification, authentication, access control, and accountability including non-repudiation of communicating and cooperating principals as well as integrity, confidentiality, and availability of communicated information. Additionally, some notary's functions are needed. In the case of non-human principals, accountability services don't make sense and *auditability* has been introduced instead. Application security deals with the services authorisation and access control as well as the accountability and audit of the principal for information and procedures handled, including their accuracy. Furthermore, the application security concerns the services integrity, confidentiality, and availability of recorded, stored, processed, and communicated information as well as some notary's functions mentioned already. In that context, the way to characterise an information object by its attributability should be established. Identification and authentication are basic services needed in the context of most of the other services in the context of communication and application security. The model concerns communicating principals in a really generic way including users, systems, applications, system components or even atomic objects in the sense of either active or passive entities. Therefore, also CORBA-based information systems complain to the communication (invocation of objects) and application security services approach.

Distinguishing the two concepts mentioned, the model allows to focus only on the interesting part as, e.g.,

- on communication security in the case of security enhanced EDI (HL7, EDIFACT, xDT, XML) in the sense of secure messaging (secure objects) or secure connection based on SSL[17]/TLS[18] in the sense of secure channelling on the one hand [Blobel et al., 1998a,b; CEN ENV 13608], or
- on application security to improve, e.g., authorisation and access control including the definition of roles and decision support on the other hand.

As Web services are based on communication between components, all the communication security services mentioned have to be provided. The communication includes service requests, service definitions, and service responses. Using those components invoked, proper application security services are requested.

Secondly, medical and afterwards security-related use cases (scenarios) must be specified to select the sets of security services as well as sets of security mechanisms providing these services needed in the context of a specific use case. The consideration can be refined by the higher granularity of algorithms and data, facilitating the implementers' view. Figure 6.2 presents this layered security model based on the concepts-services-mechanisms-algorithms view with different levels of granularity containing possible elements for each level.

Examples for the service-mechanism relationship are given by dotted lines. The relationships between the elements of the classification scheme are of 1,n:1,m type. Furthermore, these relationships can recursively occur as happens in the case of multiple wrapping to provide accountability for information communicated. Table 6.1 demonstrates the security services-mechanisms relationships for IT-related mechanisms in general. Additionally, there are many non-IT mechanisms mentioned, e.g., in the IBAG criteria [CE, 1993; EC, 1991; Hutt et al., 1995]. Among others, examples are fire protection, water protection, prevention of theft etc. to guarantee availability, or physically secure environment etc. to provide confidentiality.

[17] Secure Socket Layer
[18] Transport Layer Security

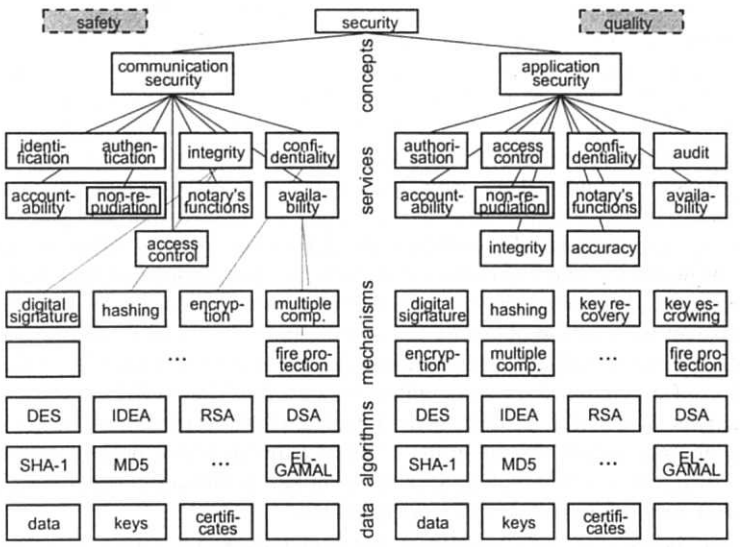


Figure 6.2: Layered Security Model Based on a Concepts-Services-Mechanisms-Algorithms View[19]

Table 6.1: Security Services and their Enforcing Security Mechanisms

Security Services	Security Mechanisms[20]	
	Asymmetric Techniques	Symmetric Techniques
Authorisation, Access Control	Digital Signature, cryptographic check value, Access Control Lists	Encryption, cryptographic check value (MAC), Access Control Lists
Principal Identification and Authentication	Digital Signature, TVPs[21]	Encryption, cryptographic check value (MAC), TVPs
Data Origin Authentication	Digital Signature, cryptographic check value, DN	Encryption, cryptographic check value (MAC), DN
Integrity	Digital Signature, cryptographic check value	Encryption, cryptographic check value (MAC)
Confidentiality	Encryption	
Accountability	Security Audit (using reports, log files, receipts, time stamps and distinguished names)	
Non-repudiation[22] (of origin and receipt)	Digital Signature, cryptographic check value, time stamps, DN	Encryption, cryptographic check value (MAC), time stamps, DN

The security services defined provide the link between security requirements and objectives as described in a security policy, and the security mechanisms and management are to satisfy these requirements. Therefore, the former concerns the medical end-user, the latter concerns system administrators. On the one hand, security services and security mechanisms may be associated; on the other hand, it is possible to distinguish between the level of the security service and its realisation. As derived in Chapter 4, several layers can be

[19] A list of abbreviations is given in the annex.
[20] For details of cryptographic algorithms, see e.g. [Menezes et al., 1997].
[21] Time Variant Parameter(s).
[22] Non-repudiation is a part of the accountability service.

defined according to different levels of granularity and/or abstraction used to satisfy the different user needs (views) as shown in Table 6.2.

Table 6.2: Security Services Levels and their Realisations

Level of Security Services	Realisations
Application	sFTP, PEM, PGP, SHTTP, ...
Service	Identification and Authentication, Integrity, ...
Mechanism	Digital Signature, Encryption, Check Values, ...
Procedure	Security proxy or security toolkits with libraries
Cryptographic Syntax	PKCS#7, S/MIME, PGP/MIME, CMS, ...
Algorithm	DES, RSA, IDEA, MD5, RIPEMD, SHA-1, ...
Level of Security Services	**Realisations**
Technical means	Tokens (Smartcard, Key disk), Software-based PSE, ...
Hard- and Software	Directory server, Certificate server, CRL server, ...

Guided by the services-mechanisms-algorithms-data relationship, the implementers have to look for appropriate protocols agreed and available products to provide the required services by implemented mechanisms using algorithms mentioned above. Table 6.3 gives an overview about protocols on different layers of the ISO-OSI model for open systems interactions, which provide the security services needed [Blobel et al., 1998a,b]. As shown in Table 6.3, communication between systems can be provided on different level of the ISO OSI model.

On the one hand, security services are provided at the application layer requiring security aware applications (and users). By that way, *secure messages* (*secure objects*) are exchanged providing *end-to-end security* also in an insecure environment. In the context of Internet-based architectures (see below in this chapter) as well as uncertainty and insecurity in the current standard PC world, this security approach should be preferred, especially if regarding such security services as accountability and liability. However, it requires integration at the application side hardly to be added to existing solutions (legacy systems, badly designed systems). Further disadvantages of such package wrapping mechanism in general but also in the context of security solutions are additional demands in real-time environments.

On the other hand, security services needed may be provided at lower layers (Transport Layer, Network Layer, Data Link Layer) by establishing a *secure channel* (*channel security*) between communicating systems. Such solution can be used in the context of security unaware applications. Thus, secure channels may be easily integrated in the environment of legacy systems. They are also applied in interface systems facilitating open communication and co-operation as Internet browsers, DICOM communication systems, etc. As demonstrated by the DICOM example, message systems are using the secure channel approach too. The advantage of secure channelling is the real-time ability (dialog-orientation), borrowed however at the expense of remaining uncertainty and difficulties in the context of legal requirements (accountability, digital signature only related to aware human user actions, etc.). Often, both solutions are combined.

Table 6.3: Security Services Provided by Protocols on Different ISO-OSI Model Layers[23]

Security Services / OSI Layers	Confidentiality	Integrity	Entity Authentication	Data Origin Authentication	Non-Repudiation of Origin	Non-Repudiation of Receipt
Data Link	SILS/SDE, PPTP[24], L2TP	SILS/SDE, L2TP	PPTP, L2TP, L2F	SILS/SDE, L2TP	–	–
Network	IPSEC, NLSP	IPSEC, NLSP	IPSEC, NLSP	IPSEC, NLSP	–	–
Transport	SOCKS, TLSP, SSL, TLS, PCT, SSH	SOCKS, TLSP, SSL, TLS, PCT, SSH	SOCKS, TLSP, SSL, TLS, PCT, SSH	TLSP	–	–
Application	SHTTP, SPKM, MHS, MSP, PEM, SFTP, PGP/MIME, MOSS, S/MIME, PKCS#7	SHTTP, SPKM, MHS, MSP, PEM, SFTP, PGP/MIME, MOSS, S/MIME, PKCS#7	SHTTP[25], SPKM, SFTP	SHTTP, SPKM, MHS, MSP, PEM, SFTP, PGP/MIME, MOSS, S/MIME, PKCS#7	SHTTP, SPKM, MHS, MSP, PEM, SFTP, S/MIME, ESS	SPKM, MHS, MSP, SFTP, S/MIME, ESS

While also some other relationships support the implementers' view on security solutions (see for example [Blobel et al., 1998a,b]), especially the medical user or the healthcare information systems administrator may (consistently with implementation details) analyse, specify, and manage security requirements and solutions by the classification scheme (Figure 6.2), selecting the sets of services and mechanisms needed corresponding to the use cases established as demonstrated in the next chapter.

Starting with the security services specification for the given architectural approach, the security solution by corresponding security mechanisms can be implemented in a really open fashion. The security services in general are independent of special scenarios and implementations, which make them simple and pretty stable. Use cases and implementation-related details on scheme level with higher granularity (especially protocols, algorithms and data) have to be updated and can be handled separately. Separating services (and partly mechanisms) from implementation details, the medical users' view, awareness, education, responsibility, ability for navigation through guidelines and for co-operation can be facilitated. Because the security model considers communicating and co-operating principals in a very generic way, the model can be used for any type of use cases (application scenarios), communication protocols and system architectures including EDI and middleware approaches alike. For different use cases another set of security services (and a set of security mechanisms) has to be selected. Contrary to this approach, other guidelines are based on implementation details and must be updated continuously. Furthermore, these guides are

[23] A list of abbreviations is given in the annex.

[24] PPTP does not address any security issues in the current version, but end-to-end security is addressed by PPP which is tunnelled by PPTP through an IP network.

[25] Only the client is authenticated to the server by showing that he is able to apply message enhancement according to the security requirements of the server.

not so clearly focusing the attention on the essential security issues the non-specialists are interested in, but they are splitting the security aspects in a crowded scheme (see for extended reference [SEISMED, 1996]).

Demonstrating the principles established in this chapter, threat-solution relationships as well as services-mechanisms relationship, depending on the concrete requirements and environment of the systems considered, will be given for such concrete examples in Chapters 10 and 11.

6.5 Domain Model and Domain Interoperability

In the mentioned case of *shared care*, an increasing number of different persons from different organisations use different methods at different times, forming temporary (or permanent) teams with the purpose to provide optimal health as physical, psychical and social well-being to the patient. To keep such complex *shared care* supporting information system manageable and operating, components of the system are grouped by common organisational, logical, and technical properties into domains. This could be done for common policies (policy domains), for common environment (environment domains), or common technology (technology domains) [Blobel et al., 1997; OMG, 1997c].

A policy describes the legal framework with rules and regulations, the organisational and administrative framework, functionalities, claims and objectives, the principals involved in, agreements, rights, duties and penalties, and the technological solution for collecting, recording, processing and communicating data in information systems. For development and management of policies, a verbal policy description, the use of policy templates expressed, e.g., in XML, or formal policy modelling can be used. The formal modelling best fits into the model driven system architecture, the template-based approach can be successfully applied for defining and negotiating policies, however. Figure 6.3 gives an example of a roughly defined policy instance.

```
<policy>
    <policy_name/>
    <policy_identifier/>
    <policy_authority/>
    <domain_name/>
    <domain_identifier/>
    <target_list>
        <target_name/>
        <target_ID/>
        <target_object>
            <operations/>
            <policies/>
        </target_object>
    </target_list>
</policy>
```

Figure 6.3: XML Policy Template Example

A High Level Policy is derived from the culture of the society that provides the environment which drives perceptions about human rights and privacy, however reflected in a specific national or local administrative environment. It describes specific operational steps that should be followed in order to fulfil a specific principle, pointing out what, but not it, must be done.

Regarding the flexibility in handling properties and policies, the domain is of a generic nature, consisting of subdomains and building superdomains. The smallest domain is the working place or sometimes even a specific component of a system (e.g. in the case of

server machines). The domain will be extended by chaining subdomains to superdomains forming a common domain of communication and co-operation, which is characterised by establishing an agreed security policy defining legal, organisational and technical security issues and the functionality required or permitted (Figure 6.4). Such transaction-concrete policy has to be negotiated between the communicating and co-operating principals, which is therefore also called policy bridging.

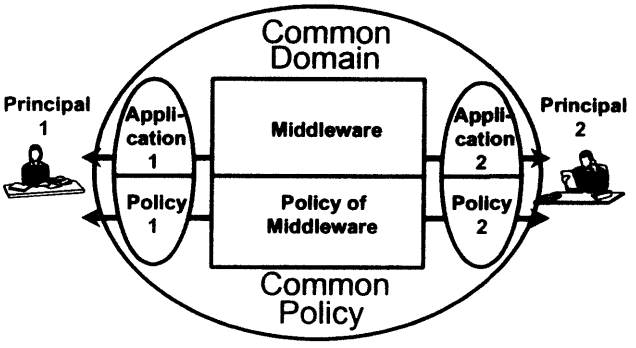


Figure 6.4: Policy Bridging

Increasingly, the distributed architecture of *shared care* information systems is based on networks. Due to their user friendliness, the use of standardised user interfaces, tools and protocols, and therefore their platform independence, the number of really open information systems based on the Internet or Intranets (corporate networks, virtual private networks) has been growing during the last couple of years.

Any kind of communication internally to a domain is called an intradomain communication, whereas the communication between domains is called an interdomain communication. For example, communication could be realised between departments of a hospital internally to the domain hospital (intradomain communication), but externally to the domain of a special department (interdomain communication).

The general purpose of communication is the provision of services to a client requesting these services. Most of the services have to be provided by the functionality of the health-care information system often combined with human users' interactions. Such application services are end-system services, indicating the case that the communication domain is only providing communication services but not additional application functionalities (see Figure 6.5). Application security services are restricted to the requested principals' domain.

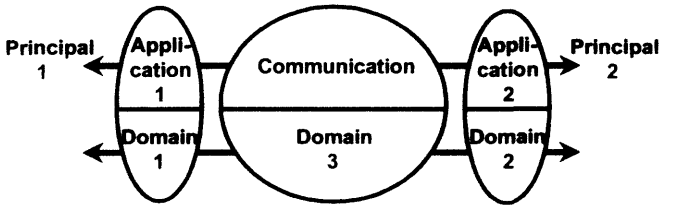


Figure 6.5: Domain Concept with Pure Communication Services

Currently, increasingly middleware concepts will be introduced into practice of healthcare information systems [Blobel and Holena, 1997]. In that case, requested services have been provided by both, principals and/or the middleware. Such architecture could be presented by chains of different domains as shown in Figure 6.6.

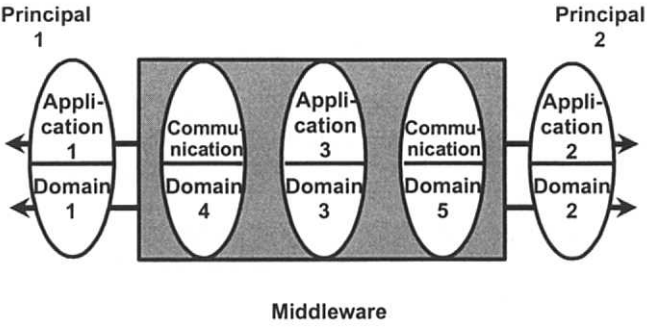


Figure 6.6: Domain Concept with Middleware Services

From the security point of view, a domain ensuring intradomain communication according to their own policy is commonly considered with need of protection only at its boundary to external domains with their specific policies (or even the policy-free domain of the Internet). This is done by, e.g., firewalls, proxy servers, etc. Regarding the external environment, a domain is therefore often handled as a closed system (e.g. Intranet). Thereby, the internal domain is assumed as secure, often neglecting internal threats and attacks. However, we should mention, that most of the security attacks are caused by insiders. Investigations have shown, e.g., that about 70% of the attacks in German health information systems and even about 95 % of such attacks in the US healthcare domain are caused by insiders. Therefore, the solution recommended is the realisation of networks of distributed security, also called end-to-end security networks or Virtual Private Networks (VPN) not only between the domains but also inside them.

Most of the security services currently available are based on system authentication (e.g. Kerberos, IPSec) [Blobel and Katsikas, 1998; Katsikas et al., 1998]. Regarding the specific requirements and conditions of healthcare, the underlying security model must consider the whole spectre of security services and mechanisms. Thus finally, a more realistic concept is solely that of secure micro domains only (e.g. [Blobel et al., 1997]).

It should be reminded finally however, that grouping of entities into domain could be performed by any common grouping parameter which might be legislation, organisation, function, technology, etc. Therefore, two special diagnosis workstations, a department, a regional network of oncologists, a hospital, a pan-European network, a group of independent consultancies, or even three Apple Computers within a Windows PC environment can form a domain.

6.6 Methodology Proposed

Summarising the approach proposed, analysis, specification and implementation of security services and related mechanisms can be managed in the following way:

1 Definition of the appropriate domain for establishing the security policy agreed on.

2 Definition of security objects, e.g. in the sense of the OMG/CORBA approach specifying security services and mechanisms needed in healthcare [Blobel and Holena, 1998]. The definitions are available and can be used by everybody (see paragraph 3).

3 Specification of use cases (healthcare scenarios, application scenarios) and the remaining set of security services (security-related use cases formulating security requirements from the user's point of view) including the valuation of these services. This step is the genuine task of end users like doctors, managers, and application systems administrators.

4 Specification of the architectural approach, which defines the general security model's concept (communication security, application security, see also paragraphs 3 and 4) to be considered and therefore the basic set of security services. This work has to be done by the IT decision makers.

5 Realising a detailed threat and risk analysis and specifying security requirements considering the use case specifications. This is a task of security officers and specialised users or administrators respectively.

6 Selection and specification of security mechanisms for provision the security services at the level needed according to the risk assessment. On that level, the approach proposed meets the IBAG Control Functions within the countermeasures framework.

7 Considering IT-related security mechanisms, implementation of the security environment needed using appropriate algorithms and protocols. This is an implementer's task.

In steps 2 and 3, also the domain specification (security policy domain, security environment domain, security technology domain) including policy bridging are required (see for details paragraph 4).

Services like identification and authentication are related to the invocation of objects or in a broader scope of principals. Therefore, identification and authentication are communication security services also providing the basis for important application security services as, e.g., authorisation and access control. Within the concrete healthcare environment, identification and authentication involve, e.g., patients, Health Professionals, but also common materials, specimen, medical products, devices, etc.

Because services can be generally considered as implemented objects invoking other objects (objects uniquely identifiable include data and methods applicable to that data as well), all kinds of object instances (documents, procedures, etc.) can be addressed.

6.7 Security Services

A security service defines a set of security functions. In (health) information systems internal and external security services can be distinguished. Internal security services describe functions provided by communicating and co-operating information systems. External security services are functions provided by Trusted Third Parties (TTPs) to facilitate trustworthiness between the principals involved in communication and co-operation [Blobel and Pharow, 1997b; Pharow and Blobel, 1999; TRUSTHEALTH_WWW]. Such services are, e.g., registration services, naming services, certificates, directory services or secure time services. Services within a security infrastructure facilitate the doctor's freedom of choice for communication and co-operation.

Some of the security services are preventing security breaches, just as strong authentication inhibits masquerading. Other security services give the evidence of security breaches without hindering them technically. Examples of the latter are services like integrity or accountability. Thus, in an open insecure environment the lost of integrity cannot be prevented but detected. By multiple wrapping (signing of signed information using countersignatures) accountability (including non-repudiation) of information communicated can be proved

which is sufficient for legal reasons. Cutting the original signature and replacing it by another one, e.g. in the cases of intellectual property rights violations, is not avoidable, unless techniques like water sign or steganography are used for some media. Because these mechanisms are not yet mature enough, some advanced attacks as compression (fractal, wavelet) and other techniques are thinkable.

The need for strong user authentication is essential for all business which requires accountability (and audit) for legal or ethical reasons. A further service related to the user's secure identity is the confidentiality of information and procedures. Additionally, the demand of user authentication in healthcare is motivated to fulfil the „need to know"-principle, to accept the privacy of patient's information, to bind information to the care purpose, and to facilitate the trustworthy doctor-patient relationship. Therefore in Europe, but increasingly also in other regions of the world, security tokens as personal and/or professional smart cards (chip cards with a crypto controller), in the future combined with biometric measures, have been introduced. They keep private keys and certificates thus providing security services as authentication, digital signature, and encryption. As general security services and mechanisms independently of the Internet, cards and card readers, as well as principles and tools of the security infrastructure like TTP services are currently under standardisation [CEN ENV 13729; TRUSTHEALTH_WWW].

In healthcare, such cards contain certificates related to the profession of the card holder and his/her roles, therefore also called Health Professional Cards (HPCs). Details on the European security infrastructure based on security tokens as HPCs and the related TTP services are discussed in [Blobel and Pharow, 1997b; Blobel and Pharow, 1998; TRUSTHEALTH_WWW].

6.8 Security Mechanisms

As mentioned in paragraph 3, the implementation of security mechanisms is depending on the state of the art, the development of (new) technologies and their availability to potential attackers. In that context, especially the Internet development provides new threats, risks, challenges and solutions. Therefore, the implementation of security mechanisms (security solutions) is a highly dynamic procedure which can and must be handled widely outside the end-users' view on domain-specific use cases and corresponding, rather stable, security services. Nevertheless, there is a need for education and improvement of awareness about problems and solutions including the end-users. Security mechanisms are described in the ISHTAR Deliverables D09 and D23 [Baum-Waidner et al., 1998; Blobel et al., 1997] as well as in further papers.

6.9 Modelling of Users' Security Needs

Responding to the end-user requirements for security enhancement, only a part of the UML methodology is really needed. In that context, the use case diagram and sometimes also the sequence diagram as well as the activity diagram must be mentioned. Using the same advanced methodology supported by appropriate toolkits such as Rational Rose™ for implementation too, also other UML diagrams are used as currently happening.

The use case defines a framework for using an information system. Starting with an abstract use case type, the use case instances describe concrete application scenarios in the sense of the description of business processes and their communication/interaction with actors. Actors in the healthcare domain are principals like Health Professionals (e.g. doctors, nurses, administrators, technical staff, management), patients, people from other domains, but also organisations or systems like Policy Councils or TTPs. Often, the domain-

specific description of the use case is done verbally. Looking for security in information systems, especially security-related use case instances must be considered.

To model the needs of the Health Professionals (medical users, medical and technical staff, administration, management, legal experts), the use of the UML tool-set should be recommended. Depending on the different user groups' need, an appropriate granularity of the model may be depicted. The specific components can be described by abstract types using the OO properties like inheritance etc. Complex scenarios may be created combining the abstract or basic types needed. The methodology presented has also been used within the TrustHealth-2 project to investigate legal implications on health information systems' security solutions [TRUSTHEALTH_WWW].

6.10 Health Use Cases

Analysing and grouping the real-world scenarios, basic scenarios or abstract use cases may be defined as mentioned above, which enable the description on any real scenario by combination of use cases types specified.

Regarding the last 2 years of activities and the related results of the ISHTAR project funded by the European Commission [Blobel and Roger-France, 1998], a use case diagram has been developed showing the basic health use case types occurring (Figure 6.7). As possible actors, Health Professionals (HP), patients, patient-related persons, and external parties may occur. Sometimes it might be helpful to define specialised HPs controlling authorisation and access to sensitive medical information in the sense of trusted authorities (e.g., EHCR authorities), who could also audit other processes acting as quality assurance authority.

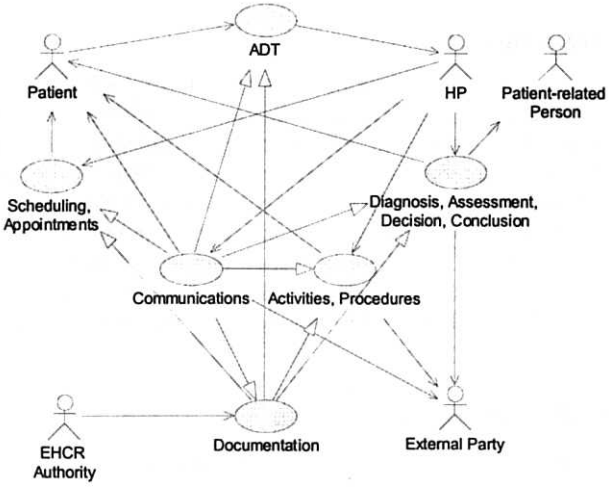


Figure 6.7: Abstract Health Use Case Types

Refining and exemplifying the business processes occurring, administrative tasks (use cases) and medical tasks (use cases) might be distinguished. Grouping these tasks, the abstract administrative and medical use cases presented in Table 6.4 can be found in analogy to Figure 6.7. The relationship to the Swedish approach described in the next paragraph is mentioned by reference numbers.

Table 6.4: Abstract Administrative and Health Use Cases

Administrative Use Cases	Medical Use Cases	Ref. #
Admission, discharge, transfer		1
	Diagnosis, assessment, decisions, conclusions	2
Scheduling and appointments		3
Financial transactions	Activities: Visits, Diagnostic procedures, Treatments, Care procedures	4
Non-medical communications: Insurance Communications, Supplier communications	Medical communications: Order entry, Result reporting, Access to patient information	5
	Reports (medical documentation)	6

Currently modelling and developing a Swedish Electronic Health Record based on the EHCR communication standard [CEN ENV 13606], the groups involved have found the following medical abstract use case types [TRUSTHEALTH_WWW]:

1 Establishment of contact between patient and Health Professional
2 Assessment/conclusion by the Health Professional
3 Creation of a specific healthcare plan for the patient
4 Activities are initiated, performed and looked after
5 Access to patient information
6 Record of healthcare information
7 Conclusion

6.11 Health Use Case Examples

To illustrate the specification of medical scenarios within the Figure 6.7 framework, examples of health use cases will be discussed shortly. For that reason, the real environment of the Magdeburg regional cancer registry is deployed showing two typical communication scenarios between the registry and co-operating institutions: the transfer of doctor's report (Figure 6.8) and the request for patient data to retrieve or distribute patient-related information according to the policy agreed and the rights given (Figure 6.9).

Fundamentals are the patient's consent for communication of his/her data, the strong authentication of the HPs to the communicating and co-operating principal as well as appropriate access rights according to the organisational and process-related role of the information requester.

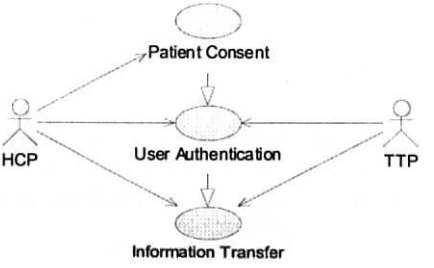


Figure 6.8: Use Case "ReportTransfer"

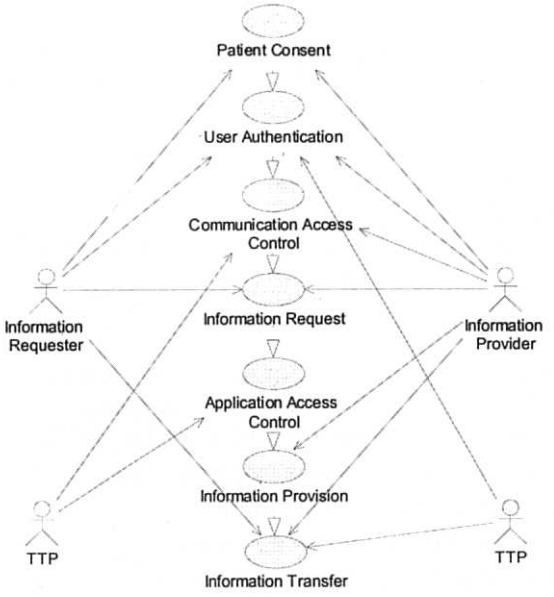


Figure 6.9: Use Case "PatientDataRequest"

6.12 Security Use Cases

To describe security-related use cases for open systems communication and co-operation, a set of abstract use case types has been defined. Afterwards, the different security-related use cases can be created combining the appropriate basic use cases.

6.12.1 Abstract Security Use Cases

The abstract security-related use case types defined which fulfil all the requirements of the health use cases are:

- the users management,
- the user authentication,
- the patient consent,
- the communication initialisation,
- the information request,
- the access control,
- the information provision, and
- the information transfer.

The use cases describe business processes with interacting principals. To facilitate the understanding of the models, sometimes the principal "user" or "HP" has been introduced. It should be stressed, however, that the use cases are valid for any type of principals as, e.g., applications event driven communicating or objects services invoking. The abstract security-related use cases will be explained in some details now.

6.12.1.1 Abstract Basic Use Case "UserManagement"

To handle any kind of user-related issues, the management of users including the specification of their possible roles and the rules applied to fulfil a security policy is needed as basis for all other application and communication security services. Regarding the possible roles, we can distinguish basic roles as the general profession (e.g. physician) expressed within the X.509v3 certificate. However, there are some other roles controlled and regulated by different organisations. In that context, qualifications like the medical profession and further specialisations, in Germany managed by the Physician Chambers of the different German states, as well as permissions like the right to practice in special domains and locations, in Germany decided by the corresponding Statutory Health Care Administrations (Kassenärztliche Vereinigungen) for GPs or by the employer for employed physicians, should be mentioned. Thus, several sets of attribute certificates according to parts of the needed X.509v3 standard will be used (see the next paragraph and also Chapter 9).

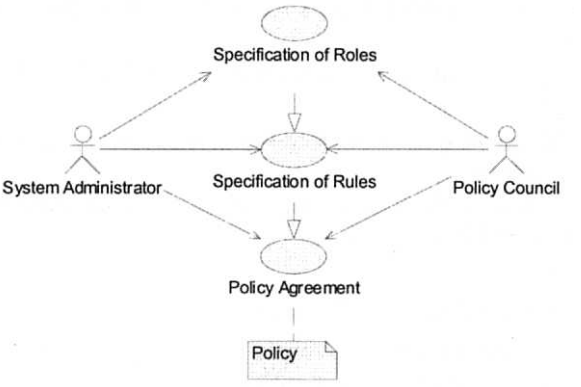


Figure 6.10: Abstract Basic Use Case "UserManagement"

Whereas the roles may be defined globally, the locally valid rules and the corresponding policy framework must be agreed between the (at least locally) trusted authorities policy council (definition) and system (security) administrator (implementation and control).

Because the definition and instantiation of roles provides the basis for healthcare-specific information systems management, roles and credentials are considered in more detail in the following subsection. The considerations are based on the TrustHealth-1 work we have been involved in, which has been extended.

6.12.1.1.1 Characteristics of Professional Data to be Certified

The following criteria may be helpful to characterise the professional data to be certified in general:

- Data to be certified should not be of highly temporary nature. Otherwise, the certificates have to be up-dated and thus reissued very frequently. So, information like telephone numbers, postal addresses, etc. may not be certified but of course included in the public directory service to help to identify Health Professional uniquely.
- Only information which is relevant to be authenticated globally (centrally) has to be certified, e.g. across the world, across one or more continents, one or more countries, or at least one or more regions (e.g. border regions).

Especially locally defined authorisation information dedicated to what is called access control should not be certified. This has to be managed by the local IT systems and administrators respectively themselves as the local authorisation information varies extremely from case to case and from application to application.

It is also recommended that signed attribute certificates of the professional status are issued directly by the relevant bodies nowadays responsible for authorisation of the various categories of HPs. This implies that in case a Health Professional is registered for more than one profession (e.g. a licence as a physician and as a dentist), the respective professional authorities operate separately and thus independently: The Health Professional gets a separate certificate for each profession. However, being registered for more than one profession occurs rarely.

In addition, (if possible asymmetric) professional status-oriented group or class keys for anonymous or group access, especially in relation with the use of patient data cards[26] and digital archiving systems including encrypted medical data should be considered. Such issuing procedures have to be done in close connection with the personalisation of related certificates and token.

Finally, the following professional data set has been identified to be sufficient in general for most of the applications:

1. Reference data of the respective professional authority (e.g. physicians chambers or similar other bodies)
2. Unique professional identifier within one professional authority (see definition scheme of distinguished names)
3. Qualification

- Profession (e.g. physician, dentist, nurse)
- Speciality (e.g. cardiologist, nephrologist)
- Further qualification (for e.g. sonographical devices)

4. General authorisation

- Licence for a certain professional role (e.g. as a hospital physician in a certain department)
- Licence for a certain professional function or activity (e.g. as a medical officer for a health insurance company)
- Licence to practise the profession in a certain environment (county, state, country)
- Licence to practise as a specialist in a certain environment (county, state, country)
- Right to prescribe special drugs (e.g. morphine)
- Special locally available rights, etc.

It can be imagined that various fields of the professional data set are delivered by different professional authorities. In that case, the level of assurance concerning their authenticity may vary from case to case, too. The number of the fields and their possible values has been considered as an open issue. It depends on the security policy (access rights).

With regard to professional certificates, it is recommended that if a Health Professional gets a certificate for some speciality, then there is no need to keep the „old" certificate for the respective profession without any speciality in addition any more, as the one for the speci-

[26] For card-card interactions card-verifiable certificates are the way of choice (see also Chapter 11).

ality implies now the one for the profession. Both items are part of one and the same certificate.

Of course, in this case the respective professional authority for the speciality has also to be responsible for the up-to-date authenticity of the profession, as well.

Revocation of the right of practising as a specialist is a very seldom case. However, if this occurs, then the Health Professional should get a certificate for the profession only, as long as the practise right for the profession itself is not revoked.

6.12.1.1.2 Professional X.509v3 Certificates

An important advantage of using the X.509v3 certificates not only for identity certificates but also for professional attribute certificates is that several available products for instance for public directory services could be used even if one does not certify a public key but only the assignment of a certain professional status (a set of attributes) to a certain Health Professional.

With particular regard to this, the extension "subject directory attributes" should be used. This extension should contain the professional status information (professional data set), whereby the professional status information is a sequence of attributes to be stored in the X.500 entry of the subject.

The key usage restriction filed need not be adapted from the public key certificate as the professional certificate does not concern any public key.

6.12.1.1.2.1Specification

The below specification is a variant of the X.509v3 certificate. The extension "subject directory attributes" should consist of the professional data set HealthProfData (see below). Note: The fields identical to those of the public key certificate are not elaborated. They are mentioned in [TRUSTHEALTH_WWW] as well.

```
Certificate     ::=     SIGNED SEQUENCE
    {
    version                         [0]     Version DEFAULT v1,
    serialNumber                            CertificateSerialNumber,
    signature                               AlgorithmIdentifier,
    issuer                                      Name,
    validity                                    Validity,
    subject                                     Name,
    subjectPublicKeyInfo                    SubjectPublicKeyInfo,
    issuerUniqueIdentifier      [1]         IMPLICIT UniqueIdentifier OPTIONAL,
    subjectUniqueIdentifier     [2]         IMPLICIT UniqueIdentifier OPTIONAL
    extensions                  [3]         Extensions MANDATORY
    }
```

version is the version of the encoded certificate. I The certificate version SHALL be v3.

6.12.1.1.2.2Professional Data Set

The proposal for the professional data set HealthProfData is mainly based on the data object HPProfData to be stored in addition to many other objects on Health Professional Cards [TRUSTHEALTH_WWW]. This data object takes into account CEN/TC 251 WG7 document named N45/46 but additional fields have been added to fulfil the overall requirements of the healthcare and welfare sector.

The terms written in italic style are additional recommended fields.

HealthProfData ::= SET

{

HPNatInfo	**[0]**	**National Information of HP**	**(SEQUENCE)**
HPProfessions	**[1]**	**SEQUENCE OF HPProfession**	
		Codes corresponding to recognised professions	
HPSpecialisations	**[2]**	**SEQUENCE OF HPSpecialisation**	
		Codes corresponding to recognised specialities	

}

6.12.1.1.3 Recommendation

The professional data set should take only one profession into account. That means, for each profession a separate professional data set (and thus a separate professional certificate) should be issued. The bodies' responsibility for data items in certificates is another key issue for separate certificate handling. Consequently, the respective sequences of fields [0] and [1] should contain only one entry. In this case, field [2] regards the respective profession.

HPNatInfo ::= SET

{

HPRegCountry	**[0]**	**Registration Country of HP**
HPNatRegId	**[1]**	**National Registration Identification**
HPNatProfession	**[2]**	**OCTET STRING (SIZE(1..5))**
		This will allow national codes for exact definitions of registered HP profession.
HPNatRegSpecialisation	**[3]**	**OCTET STRING (SIZE(1..5))**
		OPTIONAL
		This will allow national codes for exact definitions of registered HP specialisation. The references have to be mentioned anyhow.
HPNatRegRole[4]	***OCTET String (SIZE(1..5))***	
		OPTIONAL
		This will allow national codes for definitions of registered HP roles (e.g. testifying doctor, licence to work on behalf of a health insurance company)

}

HPProfessions ::= SEQUENCE OF HPProfession

HPProfession ::= OCTET STRING (SIZE(3..5))

Codes corresponding to recognised professions (has to be coded[27])

{

Physician	**[0]**
Dentist	**[1]**
Pharmacist	**[2]**
Midwife	**[3]**
Nurse	**[4]**
Physiotherapist	**[5]**
Psychologist	**[6]**
Psychotherapist	**[7]**
Speech therapist	**[8]**
Chiropractioner	**[9]**
Optician	**[10]**

[27] The codification of professions (Work-item for CEN/TC251 former WG7) can be based on the European list of recognised diplomas: List of Council Directive 93/16/EEC, April 5 1993

Dental nurse **[11]**
Dental hygienist **[12]**
Dispending pharmacist **[13]**
Administrator ***[14]***
Researcher ***[15]***
etc.
}

HPSpecialisations ::= SEQUENCE OF HPSpecialisation
HPSpecialisation ::= OCTET STRING (SIZE(3..5))
Codes corresponding to recognised professions (has to be coded similarly[28])
{
Cardiologist ***[0]***
Nephrologist ***[1]***
etc.
}

The specifications have been introduced in several European countries for establishing health networks at national and international scale. A European proposal for dealing with Health Professional data has been performed in context of the European Prestandard on Secure User Identification – Strong Authentication using Microprocessor Cards (SEC-ID/CARDS) [CEN ENV 13729]. This proposal widely reflects the German HPC specification [HCP-Protocol_WWW].

Based on the activities described, further specifications have been provided at global scale such as the work within the ISO TC 215 "Health Informatics" launched in 1998. These specifications are presented in detail in chapter 6.13.6.

6.12.1.2 Abstract Basic Use Case "UserAuthentication"

The authentication of principals communicating and co-operating via information systems is the basic service also needed for other security services and mechanisms as authorisation, access control, accountability etc. Authentication in distributed health information systems must be provided mutually and in a strong way using cryptographic algorithms. In our context, we consider human users keeping in mind the generalisation to principals. The TTP provides the user's identity certificate. For any functional and security services in context of both application and communication the user identification and authentication is needed to be compared with the system-wide unique user identifier. In some countries however, unique person identifiers are not allowed. In that particular case, a vehicle has to be found in order to allow the use of specific services dealing with a unique name in terms of security-related services.

The user initiates the process of "UserAuthentication" by starting an identification process (e.g. performing his/her HPC) followed by an authentication process (e.g. presenting his/her PIN or the characteristic biometrics). The result of this abstract basic use case is the "UserIdentifier" to be used for the system's components. "UserAuthentication" is used for, e.g., the identification and authentication process between a principal "User" and his/her PC.

[28] The codification of specialities (Work-item for CEN/TC251 former WG7) can be based on the European list of recognised diplomas: List of Council Directive 93/16/EEC, April 5 1993

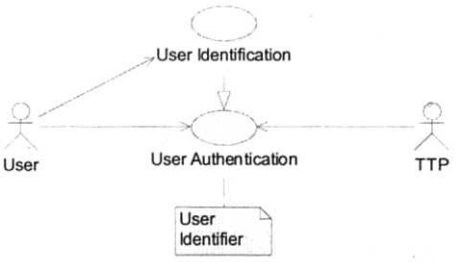


Figure 6.11: Abstract Basic Use Case "UserAuthentication"

6.12.1.3 Abstract Basic Use Case "PatientConsent"

According to the European Union Directive 95/46/EC on the Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of such Data as well as the national legislation of several European countries [CE, 1995], the verifiable and therefore usually written consent of the patient is needed to collect, record, process and communicate personal medical data, if there are no other reasons as underlying legislations or specific reasons (e.g., the protection of the data subject's or third parties' life or health).

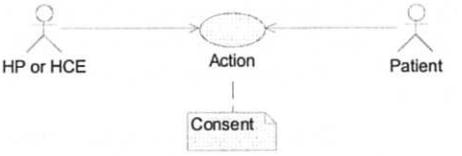


Figure 6.12: Abstract Basic Use Case "PatientConsent"

6.12.1.4 Abstract Basic Use Case "CommunicationInitialisation"

To initialise communications, a mutual (preferred) three way authentication procedure must be provided. The authentication must be verified by the certificates provided by a TTP. Because communication between domains means policy bridging, policy negotiation is required to define extension, issues, rights etc. for communication.

This use case realises an identification and authentication process between two principals (users or systems) by identifying and authenticating each other and negotiating a specific policy.

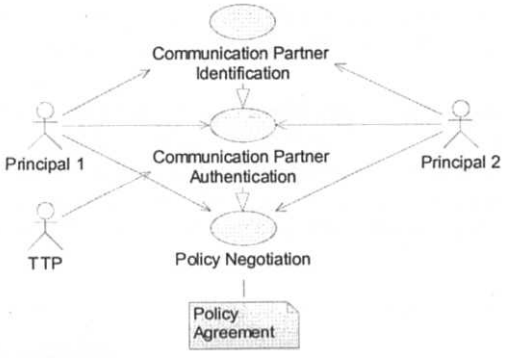


Figure 6.13: Abstract Basic Use Case "CommunicationInitialisation"

6.12.1.5 Abstract Basic Use Case "InformationRequest"

This use case is initiated by an information requestor who requires any kind of information from an information provider. Due to often different terminologies applied and syntax and semantics used, information requested must be specified to be identified and provided afterwards.

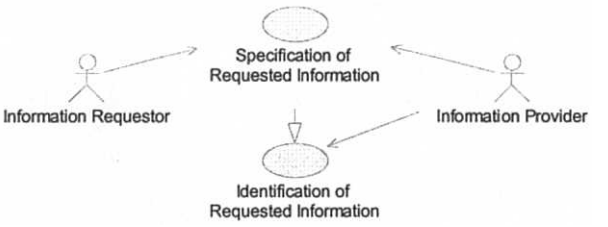


Figure 6.14: Abstract Basic Use Case "InformationRequest"

6.12.1.6 Abstract Basic Use Case "AccessControl"

Because information requestor and information provider may belong to different domains with different specifications and terminologies used, the requested rights must be specified qith the information provider's environment. Fulfilling the need to know principle and the privacy rights of the patient, the access to and the use of patient's information must be restricted and controlled. The control of access rights is based on the role of the principal (e.g., user), who requests access, and the underlying rule for decision according to the policy agreed. In that context, the rather static role definition presented in Chapter 6.12.1.1 must be extended reflecting the conditions within a *shared care* framework, i.e., hospitals, health networks including several GPs as well as other healthcare providers, etc. Additionally to the roles discussed already, the HP has to provide organisation-related as well as function-related roles within that *shared care* framework as explained in more detail in Chapter 6.12.2.1. The access rights follow a mandatory access model or a discretionary access model depending on the underlying process model which distinguishes between the rather static behaviour of the organisational (structure-related) model and the highly dynamic functional model representing the doctor-patient relationships. Both models are influencing the access control model with a legally defined domination of the discretionary procedure according to the principle of patient's transfer (see Chapter 6.12.2.1). On that way, the functional and data access rights of the different user or user groups respectively in correspondence to their functional and organisational (structural) roles are defined and decided according to the rules agreed. In that context, the rights of requester and provider of information must be discussed also regarding specific conditions like emergency cases. The definition of rights, roles and rules as well as the grouping of information and users is out of the scope of this use case, because they have to be specified during the establishment of users, departments, etc. within the"UserManagement" use case (Chapter 6.12.1.1).

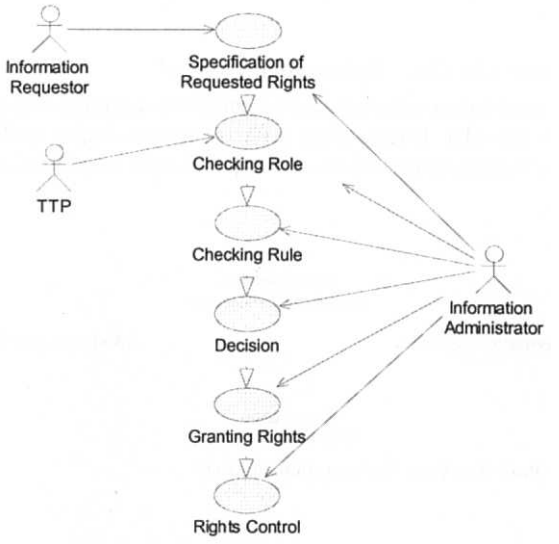


Figure 6.15: Abstract Basic Use Case "AccessControl"

6.12.1.7 Abstract Basic Use Case "InformationProvision"

The provision of information deals only with the selection and delivery of information at the provider's side.

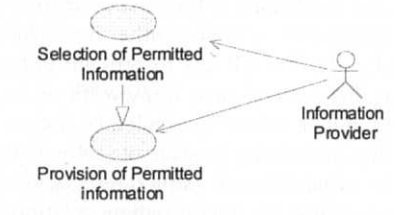


Figure 6.16: Abstract Basic Use Case "InformationProvision"

6.12.1.8 Abstract Basic Use Case "InformationTransfer"

The abstract use case "InformationTransfer" is defined in an very generic way also including the record of information and its transfer between user and system. Therefore, both application and communication security services dealing with integrity, confidentiality, and accountability (in the context of communication security also dealing with non-repudiation of origin and receipt) are reflected in the model presented. To fulfil the policy agreed, beside the users also the information has to be classified and grouped.

Responding to the information request, the information transfer provides the information permitted to the requestor's side.

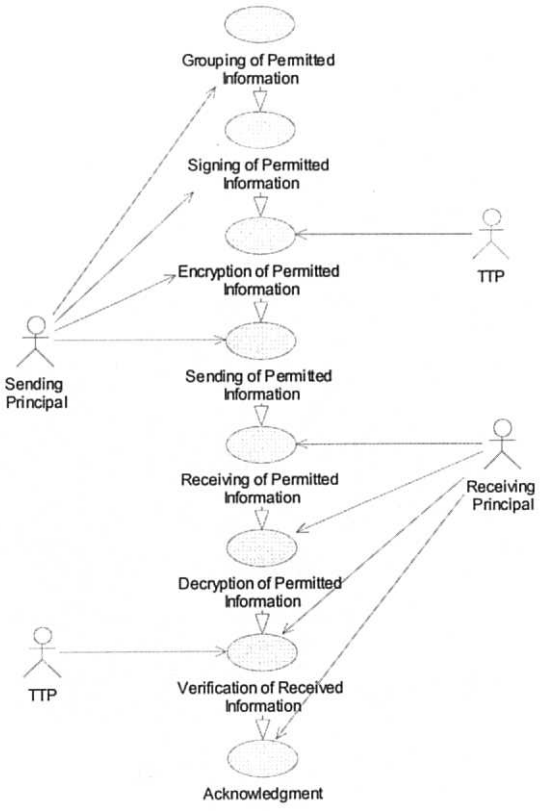


Figure 6.17: Abstract Basic Use Case "InformationTransfer"

6.12.1.9 Refinement of the AccessControl Use Case

Regarding the access decision services discussed in Chapter 6.12.1.6, refinements have to be specified to reflect specific policy requirements ruling access decision in detail.

Figure 6.18 corresponds to the CORBAmed "Resource Access Decision Service" specification. This service enables access decisions to any resource named and attributed, locating and referencing the appropriate policy. The resource controlled might be a CORBA object, a legacy system or something else, differing in the level of interoperability achieved. Therefore, data and operations are included as well, usually separately specified in "classic" policy or access control models.

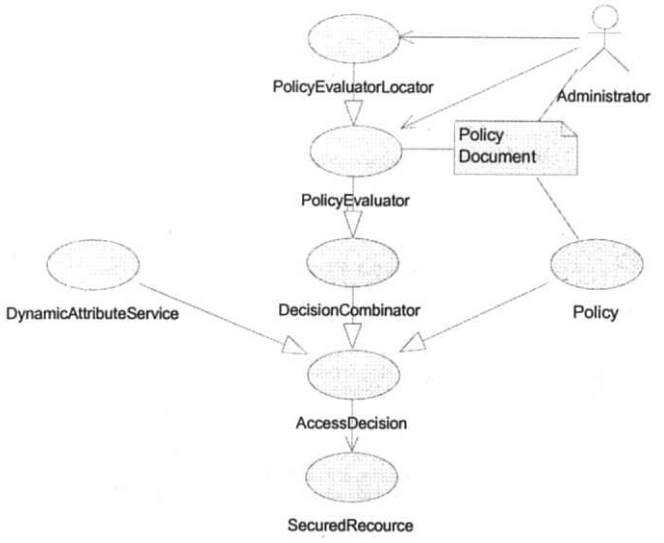


Figure 6.18: Use Case "CORBA RecourceAccessDecisionServices"

Figure 6.19 demonstrates the corresponding information model.

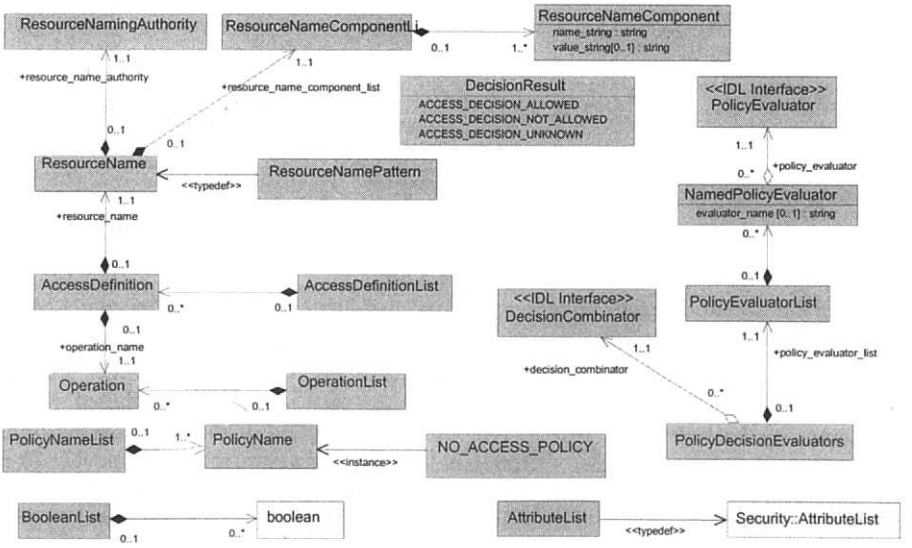


Figure 6.19: Resource Access Decision Information Model (after CORBA [CORBA, 2000])

Using such resource access decision service, the access to medical information held in EHCR systems can be provided using specific access services defining access methods, data types, characteristics, etc. of the data stored or retrieved. Such services have also been specified in the CORBA world such as, e.g., the Clinical Observations Access Service (COAS) and the Clinical Image Access Service (CIAS) [CORBA, 2000].

6.12.2 Derived Issues on Application Security

Establishing a security infrastructure for secure healthcare applications, authentication and certificates provided by the HPC are used to facilitate communication security services but also application security services as authorisation, access control, accountability etc. In that context, the authentication is deployed to derive an identifier and the attribute certificates describe the role(s) of the Health Professional (HP).

6.12.2.1 Health Professional Roles

Regarding the role of HPs in the healthcare business, two different role systems must be distinguished: structure-related roles and function-related roles. The definition and rule-based interpretation of the HPs' roles are explained in Chapter 12.1 (Basic Abstract Use Cases).

The structure-related role of an HP defines his/her position in the organisational hierarchy of the institution reflecting responsibility and competence of the professional. This scheme is a rather static one. With respect to the access control procedures it describes a mandatory model.

Examples for structure-related role in healthcare systems are:

- Medical director
- Director of clinic
- Head of the department
- Senior physician
- Resident physician
- Physician
- Medical assistant
- Trainee
- Medical student
- Head nurse
- Nurse

The function-related role of an HP immediately reflects the position in the healthcare process, i.e., the concrete HP-patient relationship. It represents a highly dynamic relation, which follows discretionary model approaches.

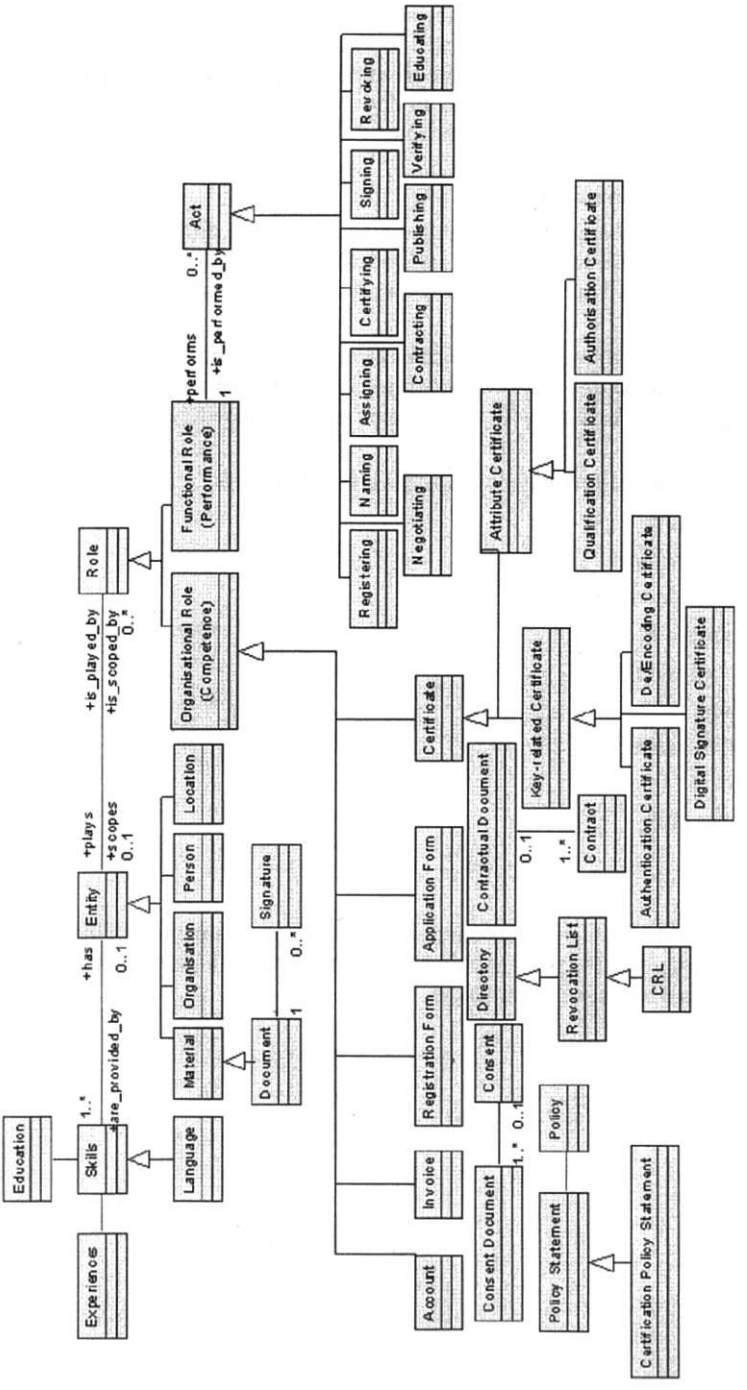


Figure 6.20: Interoperability Summit's Information Model (Class Hierarchy) for Human Resources

Examples for function-related roles in healthcare systems are:

- Caring doctor (reliable doctor[29])
- Member of diagnostic team
- Member of therapeutic team
- Consulting doctor
- Referring doctor
- Attending doctor
- Family doctor
- Attending Nurse

Both roles define the rights and duties of an HP in a Health Care Establishment (HCE). Because HPs fulfil obligations in both the organisational and the functional framework, the resulting access control model combines the two views (Figure 6.21). According to the codes of conduct, the data protection legislation and the European Data Protection Directive, in most of the democracies the function model dominates the access control model in health information systems. Details are given in [Blobel, 1996a,d].

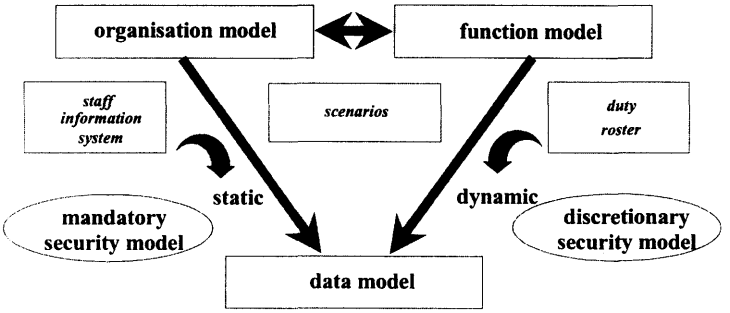


Figure 6.21: Access Control Model in Health Information Systems

6.12.2.2 Authorisation and Access Control

As described before, authorisation and access control is depending on the policy agreed and the roles of the HPs in the context of the medical scenarios. It is embedded in the security framework of the healthcare system or at least the HCE involved in the care process directly or indirectly. This security framework is shortly mentioned in Chapter 12.

The security policy is influenced by the policy of the organisation, its business goals and its organisational and IT concept.

Figure 6.36 later on represents the information model for authorisation and access control in component-based shared care information systems within that general framework mentioned. Considering that general model in the context of EHCR systems, the component must be replaced by"EHCR" with consists of the component data "EHCR_Item" and "EHCR_Item_Complex" [CEN ENV 13606, Part 1]. In Figure 6.36, the patient's rights have been reflected considering the access to his/her personal medical information, however ignoring at the schema the right to trace which HP had or has such access rights. The modelling of implementation details has been discussed in [Blobel, 1996d]. The implemen-

[29] In the healthcare system of several countries (e.g.UK), the family doctor is (or is intended to be, e.g. Germany) the reliable doctor.

tation of authorisation and access control is mainly handled in the context of database security [Castano et al., 1995].

6.13 Management of Principals

6.13.1 Roles

In the context of certification/assignment, PM introduced the concept of principals (e.g. person, organisation, device, application, component, object) which are able to use systems' services.

For managing role-relationships between the entities, organisational and functional roles can be defined. Organisational roles specify relations between entities in the sense of competence (RIM roles) often reflecting organisational or structural relations (hierarchies). Functional roles are bound to an act. Functional roles can be assigned to be performed during an act. They correspond to the RIM participation. Functional roles are an attending doctor, a member of a therapeutic team, etc.

In the context of certification/assignment, a schema of organisational roles has been established in official international standards. The PM-relevant part of this schema is shown in Figure 6.22. Figure 6.23 present the specialisation of the professional class in Figure 6.22. Both schemas can be used to establish a consistent terminology of these entities (principals/authorities) in the context of PM CMETs.

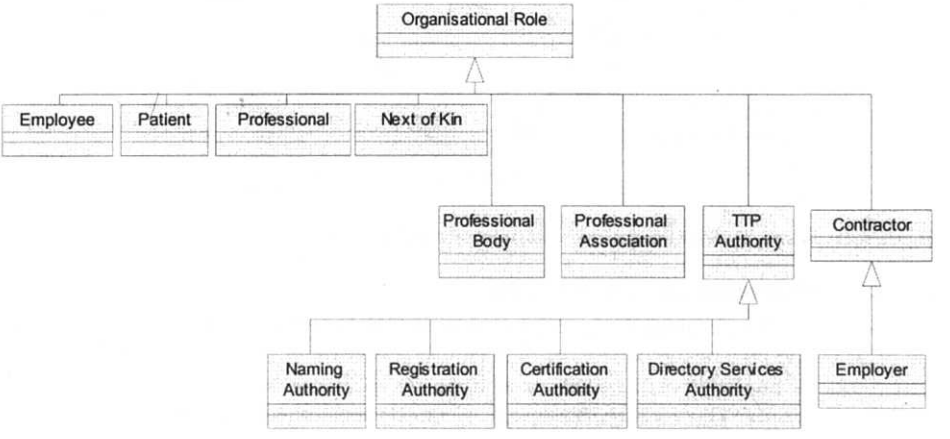


Figure 6.22: Health-Related Organisational Roles Played by the Entities Person or Organisation,

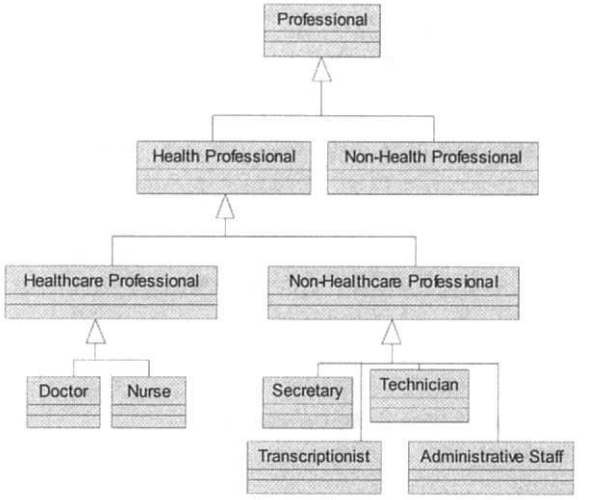


Figure 6.23: Specialisation of the professional class in the health context

Figure 6.23 groups classes of Health Professionals regarding specific types of HC-patient relations which can define certification or assignment.

Regarding the use cases for certification/assignment, the entities device, application, component, and object have been introduced as principals beside person and organisation. These principals can play functional roles, too. An example is a heart-lung-machine dedicated to emergency cases.

A way used by HL7 in its Version 3 development process is the definition of the archetypes related to security services in Story Boards. Figure 6.24 gives an example of a Story Board provided by the HL7 Personnel Management TC which is co-chaired by the author.

Event	Name	Description	Interaction
PRPM_TE004003UV01	Create certificate	Create a certificate token.	
PRPM_TE004004UV01	Issue certificate	Personalise a certificate.	
PRPM_TE004005UV01	Revoke certificate	Revoke cerficate before activated.	
PRPM_TE004006UV01	Terminate certificate	Remove certificate from directory.	
PRPM_TE004007UV01	Expire certificate	Certificate is expired.	
PRPM_TE004008UV01	Renew	Renew an expired role assignment.	
PRPM_TE004009UV01	Terminate certificate	Remove certificate from directory.	
PRPM_TE004011UV01	Purge (delete) information about certificate	Purge/delete the information about a certificate.	
PRPM_TE004012UV01	Activate cerficate	Activate a certificate.	
PRPM_TE004013UV01	Revoke certificate	Revoke active certificate before expired.	
PRPM_TE0040xxUV01	Terminate certificate	Destroy certificate before issuing it.	

Figure 6.24: HL7 Story Board for Certification

For describing the impact of role-related specialisations, State Transition Diagrams are used. Figure 6.25 demonstrates such a diagram specifying possible acts and resulting states of certificates (e.g., permissions to practice, certificate of qualifications, training certificate).

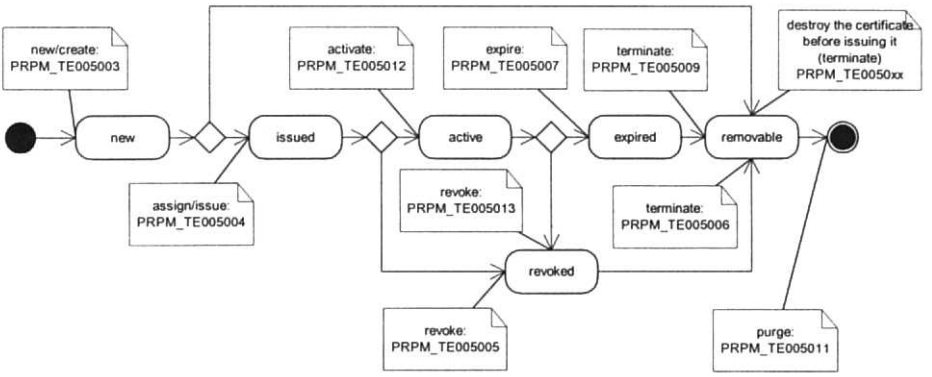


Figure 6.25: HL7 State Transition Diagram for Certificates

6.13.2 Certification Procedure

In the HL7 context, *certification procedures* are not always bound to a comprehensive legislation like the European one. Often, only information of a certain "*certification act*" is provided independently of procedural requirements described, e.g., in Chapter 7. For that reason, the certification procedure and its presentation according to HL7 Version 3 rules is following discussed shortly. This description introduces the way of thinking and discussing within a community developing open standards regardless the applications and conditions this application runs.

The *certification procedure* describes the interaction between a *certification authority* and an *entity* (to be) *certified*. The first step is the request for certification issued by the entity asking for being certified (e.g. the request of a physician for being certified) or by a principal responsible for the certification requested for the principal to be certified (e.g. the request of a technician for certifying a device the technician is responsible for). This request is submitted to a *registration authority* registering all information needed for the certification procedure. Because the certificate must be doubtlessly bound to an identity, this identity must be uniquely named by a *naming authority*. This may be done in the context of the certification procedure (in the case of key-related ID-certificates) or by using naming services of other authorities issuing birth certificates, ID-cards, etc. Now the certificate can be created by the *certification authority* and issued by an *issuing authority*. In the case of identity certificates based on public key algorithm, this step follows the generation of the keys needed within a public key infrastructure (PKI) performed by the *key generation instance*. The publication of the certificate which is signed (and in the paper world eventually separately sealed) by the certification authority is done by a corresponding authority in a proper way, e.g. via *directory services*. The used certificate which contains beside the certified properties also issuing date, (activation date), expiration date can be verified. In the case of *revocation of certificates*, the authority may inform the interested parties. Despite of that act, it publishes this revocation in so called a *Certificate Revocation List* (CRL). Activation, inactivation, renewing of certificates doesn't change the content but only the validity time of certificates.

In the paper world (example of certifying specific education), the registration authority can be the administration of the university or the examining department. This authority will commonly use the service of the naming authority issuing the birth certificate or the ID card. The certificate is created by responsible examiner and issued by the office responsible for administering the examination. The result is the paper certificate. The directory service

can be provided by the university archive, by books registering the examination result, etc. Figure 6.26 shows the HL7 CMET "Certificate or Assignment".

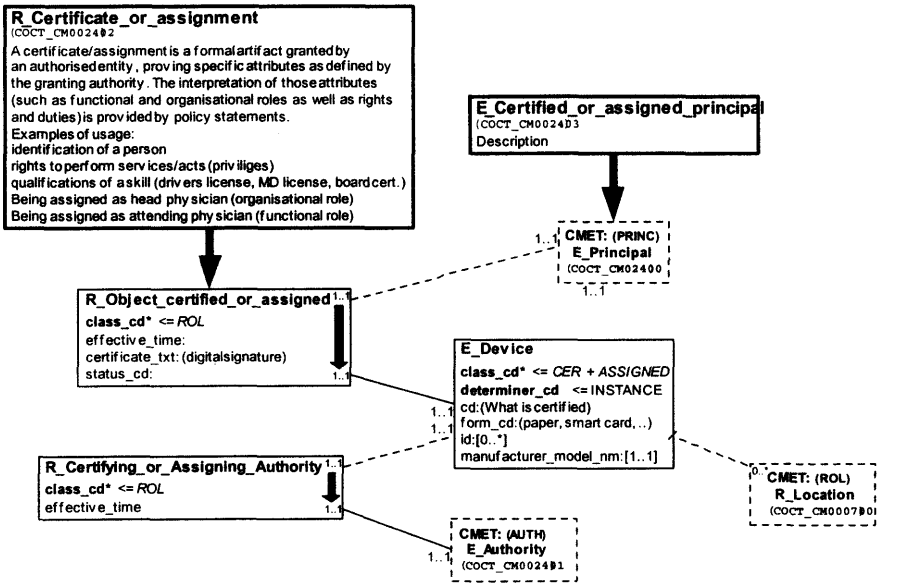


Figure 6.26: Actual HL7 CMET "Certificate_or_Assignment"

An HL7 CMET example for certificate revocation is given in Figure 6.27; the corresponding HMD is shown in Figure 6.28.

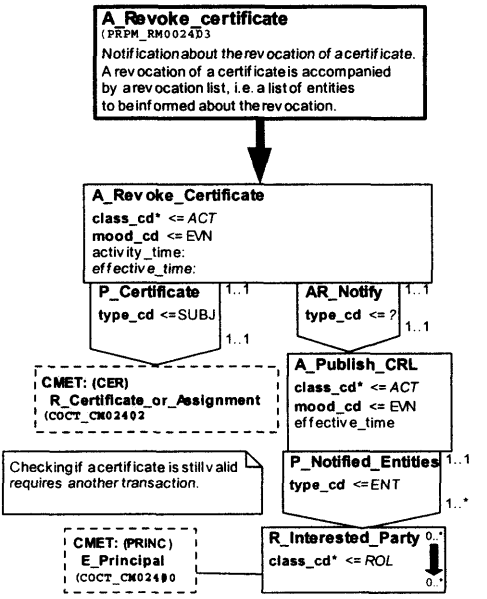


Figure 6.27: HL7 CMET "Revoke_Certificate"

PRPM_MT005013	(Link to Grid View)
A_Revoke_Certificate	Used by: PRPM_HD005013
A_Revoke_Certificate ( )	
[1..1] class_cd , Act , class_cd , D , (CS){ CNE: ActClass }	
[1..1] mood_cd , Act , mood_cd , D , (CS){ CNE: ActMood }	
[0..1] effective_time , Act , effective_time , D , (GTS)	
[0..1] activity_time , Act , activity_time , D , (GTS)	
[1..1] has , Act , has_P_Certificate , N , (P_Certificate)	
[1..1] is_source_for , Act , is_source_for_AR_Notify , N , (AR_Notify)	
has_P_Certificate	Used by: A_Revoke_Certificate
has_P_Certificate (P_Certificate)	
[1..1] type_cd , Participation , type_cd , D , (CS){ CNE: ParticipationType }	
[1..1] has_as_participant , Participation , has_as_participant_CMET_R_Certificate , C , (COCT_HD100100)	
is_source_for_AR_Notify	Used by: A_Revoke_Certificate
is_source_for_AR_Notify (AR_Notify)	
[1..1] type_cd , Act_relationship , type_cd , D , (CS){ CNE: ActRelationship }	
[1..1] has_target , Act_relationship , has_target_A_Publish_CRL , N , (A_Publish_CRL)	
has_target_A_Publish_CRL	Used by: AR_Notify
has_target_A_Publish_CRL (A_Publish_CRL)	
[1..1] class_cd , Act , class_cd , D , (CS){ CNE: ActClass }	
[1..1] mood_cd , Act , mood_cd , D , (CS){ CNE: ActMood }	
[0..1] effective_time , Act , effective_time , D , (GTS)	
[1..1] has , Act , has_P_Notified_Entities , N , (P_Notified_Entities)	
has_P_Notified_Entities	Used by: A_Publish_CRL
has_P_Notified_Entities (P_Notified_Entities)	
[1..1] type_cd , Participation , type_cd , D , (CS){ CNE: ParticipationType }	
[1..1] has_as_participant , Participation , has_as_participant_R_Interested_Party , N , (R_Interested_Party)	
has_as_participant_R_Interested_Party	Used by: P_Notified_Entities
has_as_participant_R_Interested_Party (R_Interested_Party)	
[1..1] class_cd , Role , class_cd , D , (CS){ CNE: RoleClass }	
[1..1] is_played_by , Role , is_played_by_CMET_E_Principal , C , (COCT_HD460100)	

Figure 6.28: HL7 HMD "Revoke_Certificate"

The next figure demonstrates the resulting XML message.

```
<xsd:schema targetNamespace="urn:hl7-org:v3/PRPM_MT005013" elementFormDefault="qualified">
  <xsd:annotation/>
  <xsd:import namespace="urn:hl7-org:v3/dt" schemaLocation="v3dt.xsd"/>
  <xsd:import namespace="urn:hl7-org:v3/COCT_MT100101" schemaLocation="COCT_MT100101.xsd"/>
  <xsd:import namespace="urn:hl7-org:v3/COCT_MT460101" schemaLocation="COCT_MT460101.xsd"/>
  <xsd:complexType name="PRPM_MT005013">
   <xsd:sequence>
    <xsd:element name="class_cd" minOccurs="1" maxOccurs="1" type="dt:CS"/>
    <xsd:element name="mood_cd" minOccurs="1" maxOccurs="1" type="dt:CS"/>
    <xsd:element name="effective_time" minOccurs="0" maxOccurs="1" type="dt:GTS"/>
    <xsd:element name="activity_time" minOccurs="0" maxOccurs="1" type="dt:GTS"/>
    <xsd:element name="has_P_Certificate" nillable="true" minOccurs="1" maxOccurs="1"
type="P_Certificate"/>
    <xsd:element name="is_source_for_AR_Notify" nillable="true" minOccurs="1" maxOccurs="1"
type="AR_Notify"/>
   </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="P_Certificate">
   <xsd:sequence>
    <xsd:element name="type_cd" minOccurs="1" maxOccurs="1" type="dt:CS"/>
    <xsd:element name="has_as_participant_CMET_R_Certificate" nillable="true" minOccurs="1" maxOc-
curs="1" type
="COCT_MT100101:COCT_MT100101"/>
   </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="AR_Notify">
   <xsd:sequence>
    <xsd:element name="type_cd" minOccurs="1" maxOccurs="1" type="dt:CS"/>
```

```
    <xsd:element name="has_target_A_Publish_CRL" nillable="true" minOccurs="1" maxOccurs="1"
type="A_Publish_CRL"/>
   </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="A_Publish_CRL">
   <xsd:sequence>
    <xsd:element name="class_cd" minOccurs="1" maxOccurs="1" type="dt:CS"/>
    <xsd:element name="mood_cd" minOccurs="1" maxOccurs="1" type="dt:CS"/>
    <xsd:element name="effective_time" minOccurs="0" maxOccurs="1" type="dt:GTS"/>
    <xsd:element name="has_P_Notified_Entities" nillable="true" minOccurs="1" maxOccurs="1"
type="P_Notified_Entities"/>
   </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="P_Notified_Entities">
   <xsd:sequence>
    <xsd:element name="type_cd" minOccurs="1" maxOccurs="1" type="dt:CS"/>
    <xsd:element name="has_as_participant_R_Interested_Party" nillable="true" minOccurs="1" maxOc-
curs="1" type="R_Interested_Party"/>
   </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="R_Interested_Party">
   <xsd:sequence>
    <xsd:element name="class_cd" minOccurs="1" maxOccurs="1" type="dt:CS"/>
    <xsd:element name="is_played_by_CMET_E_Principal" nillable="true" minOccurs="1" maxOccurs="1"
type="COCT_MT460101:COCT_MT460101"/>
   </xsd:sequence>
  </xsd:complexType>
 </xsd:schema>
```

Figure 6.29: HL7 XML Message "Revoke_Certificate"

6.13.3 Attestation and Assignment

Attestation and assignment in principle follow the same procedure as certification. Because of the different legal strength, the authorities involved in the procedure may act on a lower level and could be combined.

6.13.4 Qualification and Permission

Qualification and experience belong to skills enabling certain activities in the sense of competence. Skills can be certified or attested only. They cannot be assigned. Permission, authorisation authorise/allow the performance of specific activities. Permissions, authorisations or responsibilities can be certified or simply assigned. In all cases, the interpretation of certificates or assignments in terms of (derived) detailed rights and duties are commonly fixed in policy statements.

6.13.5 Managing Certification, Attestation, and Assignment

If certification and attestation as commonly managed centrally following inter-organisationally, regionally, nationally, or even internationally established requirements, rules, and procedures, the assignment is a specific local task reflecting the local requirements and conditions.

From the policy point of view, certification and attestation follow central policies, if assignment follows decentralised policies.

6.13.6 Authorisation Objects

6.13.6.1 Attribute Certificares According to ISO 17090

Principally, it is possible to use X.509v3 certificates to provide authenticity of directory attributes of a subject at any desired „level". Regarding the certificate specified here, one can indicate the fact of being an authorisation certificate, then the fact of being a professional attribute certificate and finally the professional sector in advance. Especially, this makes it possible to use the same certificate format for other sectors than healthcare, as well. And it makes it possible to use common facilities of reading, writing, updating, and verifying professional attribute certificates and identity certificates. Following, the attribute certificate defined in ISO DTS 17090 "Health informatics – Public Key Infrastructure" [ISO 17090] is presented.

```
AttributeCertificate ::= SIGNED { AttributeCertificateInfo }

AttributeCertificateInfo ::= SEQUENCE {
        version                 Version DEFAULT v1,
        owner                   SEQUENCE {
                baseCertificateId       [0]     IssuerSerial OPTIONAL,
                entityName              [1]     GeneralNames OPTIONAL,
                objectDigestInfo        [2]     ObjectDigestInfo OPTIONAL },
        issuer                  SEQUENCE {
                baseCertificateId               IssuerSerial OPTIONAL,
                issuerName              [0]     GeneralNames OPTIONAL },
        signature               AlgorithmIdentifier,
        serialNumber            CertificateSerialNumber,
        attCertValidityPeriod   AttCertValidityPeriod,
        attributes              SEQUENCE OF Attribute,
        issuerUniqueID          UniqueIdentifier OPTIONAL,
        extensions              Extensions OPTIONAL }

Version ::= INTEGER { v1(0), v2(1) }

IssuerSerial ::= SEQUENCE {
        issuer                  GeneralNames,
        serialNumber            CertificateSerialNumber,
        issuerUID               UniqueIdentifier OPTIONAL }

CertificateSerialNumber ::= INTEGER

UniqueIdentifier ::= BIT STRING

ATTRIBUTE ::= CLASS {
        &id             OBJECT IDENTIFIER UNIQUE,
        &singleValued BOOLEAN DEFAULT FALSE,
        &Syntax }

Attribute ::= SEQUENCE {
        attrType        ATTRIBUTE.&id ({SupportedAttrs}),
        attrValues      ATTRIBUTE.&Syntax ({SupportedAttrs} {@attrType}) }

ObjectDigestInfo ::= SEQUENCE {
        digestAlgorithm         AlgorithmIdentifier,
        objectDigest            OCTET STRING }

AttCertValidityPeriod ::= SEQUENCE {
        notBefore       GeneralizedTime,
        notAfter        GeneralizedTime }
```

Based on this specification, ISO TC 215 "Health Informatics" defined special HcProfessional attributes.

```
hcRole ATTRIBUTE ::= {
        WITH SYNTAX                     HCActorData
        EQUALITY MATCHING RULE          hcActorMatch
        SUBSTRINGS MATCHING RULE        hcActorSubstringsMatch
        ID                              id-at-hcpki-healthcareactor}
```

6.13.6.1.1 Assignment of object identifier values

The following values are assigned in this International Standard:
{iso (1) standard (0) hcpki (17090)}

```
        id-hcpki        OBJECT IDENTIFIER  ::= 1.0.17090

        id-hcpki-at     OBJECT IDENTIFIER  ::= {id-hcpki 0 }
        id-at–hcpki-healthcareactor OBJECT IDENTIFIER ::= 1.0.17090.0

        id-at–hcpki-healthcareactor OBJECT IDENTIFIER  ::= {id-hcpki-at 1}
        id-at–hcpki-healthcareactor OBJECT IDENTIFIER ::= 1.0.17090.0.1

        id-hcpki-cd OBJECT IDENTIFIER  ::= {id-hcpki 1}
        id-hcpki-cd OBJECT IDENTIFIER  ::= 1.0.17090.1
```

6.13.6.1.2 Definition of data types:

```
HCActorData ::= SET OF HCActor

HCActor ::= SEQUENCE {
        codedData                   [0] CodedData OPTIONAL,
        RegionalHCActorData         [1] SEQUENCE OF RegionalData OPTIONAL }

CodedData ::= SET {
        codingSchemeReference       [0] OBJECT IDENTIFIER,
       ----- Contains the ISO coding scheme Reference or local coding scheme reference
       ---- achieving ISO registration) will be OID id-th2
       ---- at least ONE of the following SHALL be present
        codeDataValue               [1] NumericString OPTIONAL,
        codeDataFreeText            [2] DirectoryString OPTIONAL }

RegionalData ::= SEQUENCE {
        type    REGIONALDATA.&id({SupportedRegionalData}),
        value   REGIONALDATA.&Type({SupportedRegionalData}{@type})}
```

6.13.6.1.3 Definition of REGIONALDATA Object Class:

```
REGIONALDATA ::= CLASS {
        &Type,
        &id     OBJECT IDENTIFIER UNIQUE }
        WITH SYNTAX         {
                WITH SYNTAX &Type
                ID          &id }
```

6.13.6.1.4 Definition of SupportedRegionalData Object Class Set

```
SupportedRegionalData REGIONALDATA ::=
        {coded,
        ... —expect additional regional/national objects to be defined
        }
```

6.13.6.1.5 Definition of coded Information Object:

```
coded  ::= REGIONAL-DATA {
        WITH SYNTAX CodedRegionalData
                ID      id-hcpki-cd}

CodedRegionalData ::= SEQUENCE {
        country                 [0] PrintableString (SIZE (2)),
                -- ISO3166 code of country of issuing authority.
        issuingAuthority        [1] DirectoryString,
                -- Identifier of issuing authority as Regional Entity. -- -- Could be implemented
                as a true identifier or a
                -- Directory lookup string (to be determined)
        hcMajorClassCode        [2] CodedData,
        hcMinorClassCode        [3] CodedData OPTIONAL
```

Codes to be used for this field e.g. ASTM E1986-98 Data User Role Name [ASTM E1986-98]

It is RECOMMENDED that the **HcProfessionalData** are taken from the appropriate national coding scheme.

6.13.6.2 Example of a Regulated Health Professional Certificate according to ISO DTS 17090

John Stuart Woolley aka Tink Woolley; license issued by State of California Medical License Board, license number 20A4073, license status code 17 ('01' is 'active and current'), issue date March 22, 2000 – expiration date March 21, 2002.

Version	(2 – decimal code for version 3 certificates)	
SerialNumber	(unique number)	
Signature	(sha-1WithRSAEncryption {1,2,840,113549,1,1,5})	
Issuer		
	countryName	(US=United States of America)
	localityName	(California)
	organizationName	(Name-of-CA-for-California-Health-Care)
	commonName	(Name-of-CA-for-California-Health-Care)
Validity		(validity period coded as UTCTime)
Subject		
	countryName	(US=United States of America)
	localityName	(California)
	organizationName	(CertHolderOrganization)
	commonName	(Woolley, Tink)
	surname	(Woolley)
	givenName	(John Stuart)
subjectPublicKeyInfo		
	algorithm	(public RSA key, 1024 bit {1,2,840,113549,1,1,1})
	subjectPublicKey	(Subject's PUBLIC KEY)

Extensions

authorityKeyIdentifier (unique identifier of CA public key)
subjectKeyIdentifier (unique identifier of subject public key)
keyUsage (digitalSignature or non-repudiation or keyEncipherment)
certificatePolicies (appropriate policy OID)
cRLDistributionPoints (CRL X.500 entry location)
subjectDirectoryAttributes

(**hcRole** OBJECT IDENTIFIER ::= OID-for-ISO-HC-Role-Attribute
hcActorData SET OF {
codedData CodedData ::= {
codingSchemeReference OBJECT IDENTIFIER ::= *ISO-Role-Coding-Scheme-OID*,
codeDataValue NUMERIC STRING ::= *the-code-for-physician-role*,
codeDataFreeText DirectoryString ::= *optional-data}*
regionalHCData Sequence of RegionalData ::= {
type OBJECT IDENTIFIER ::= OID-for-this-regional-encoding,
country PrintableString (SIZE (2) ::= ISO-country-code-for-USA,
issuingAuthority DirectoryString ::= (C=US, L=CA, OU=California Medical License Board),
nameAsIssued DirectoryString ::= (CN= John Stuart Woolley)
hcMajorClassCode CodedData ::= {
codingSchemeReference OBJECT IDENTIFIER ::= ASTM-Coding-Scheme-for-Type-OID,
codeDataValue NUMERIC STRING ::= ASTM-Type-OID-for-physician}
codeDataFreeText UTF8String ::= "license number 20A4073"}
hcMinorClassCode CodedData ::= {
codingSchemeReference OBJECT IDENTIFIER ::= *ASTM-Coding-Scheme-for-License-Status-OID*,
codeDataValue NUMERIC STRING ::= 0 *(unrestricted)*
codeDataFreeText UTF8String ::= "unrestricted"} })

Note that in this example, a license number and license status have been encoded as regional data. Such regional data is optional, and the decision to include or exclude such regional data is left up to the issuing CA.

6.13.6.3 Other Authorisation Objects

As mentioned in above, in other countries also other authorisation objects are deployed depending on the national policy. In that context, privileges and credentials widely used, e.g., in the USA have to be mentioned. Therefore, ASTM recently developed specifications based on a PKI to manage both organisational and functional user roles [ASTM, 2001]. Following, some of them are presented using the ASN.1 notation.

6.13.6.3.1 Role Certificates

Following the ISO DTS 17090 specification on role management, a general role certificate has been specified. The syntax of the role attribute is:

```
Role ATTRIBUTE ::= {
    WITH SYNTAX   RoleSyntax
    ID            id-at-role }

RoleSyntax ::= SEQUENCE {
    roleAuthority        [0]   GeneralNames OPTIONAL,
    roleName             [1]   GeneralName }
```

Not all forms of **GeneralName** are appropriate for use as role names. The most useful choices are object identifiers and distinguished names.

6.13.6.3.2 Credentials

Another common type of privilege is the user credential. This is issued by a trusted authority, and includes an identification string. Examples include licensing of medical professionals by state boards, and assignment of DEA numbers. A credential includes a type, an issuer name, and an identifier. Geographically structured issuer names can be useful to indicate state and other locality information. Credentials are typically matched by type (e.g., "physician") or type and issuer (e.g., "physician licensed in Virginia").

```
Credential ::= SEQUENCE {
        credType        OBJECT IDENTIFIER,
        issuer          GeneralName OPTIONAL,
        identifier      UTF8String }

credentials ATTRIBUTE ::= {
        &id             id-credentials,
        &SEQUENCE OF Credential }
```

If the issuer name is absent, then the issuer name from the enclosing attribute or public key certificate is used. If the certificate issuer name is absent, the credential issuer name must be present. (Note that a certificate may explicitly have more than one credential, from more than one issuer, in order to minimise the number of AAs in a system.)

6.14 XML Digital Signature

The essential services for communicated information as signer authentication, message authentication and message integrity are based on the digital signature mechanism. Considering the XML standard set, this mechanism is defined in the W3C IETF XML-Signature Core Syntax and Processing which reflects the XML-Signature Requirements.

6.14.1 The W3C IETF XML-Signature Core Syntax and Processing

Regarding the common requirements including the legal ones mentioned above,

- authentication of internal and external resources,
- authentication of part or totality of a document,
- signing of composite documents,
- detachment of signatures from document (separation of attribution info, manifest and signature),
- multiple signature (e.g., co-signature, endorsement)

must be supported.

These requirements have been met by the XML Signature element structured as shown in Figure 6.30, where "?" denotes zero or one occurrence; "+" denotes one or more occurrences; and "*" denotes zero or more occurrences.

```
<Signature>
    (<SignedInfo>
        (CanonicalisationMethod)
        (SignatureMethod)
        (<Reference (URI=)?>
            (Transforms)?
            (DigestMethod)
            (DigestValue)
        </Reference>)+
    </SignedInfo>)
    (SignatureValue)
    (KeyInfo)?
    (Object)*
</Signature>
```

Figure 6.30: W3C IETF XML Signature

The signature validation can be realised for core validation or reference validation. In the case of signing multiple data objects, the former requires a complete set of references within SignedInfo. The latter enables the validation of parts of the references. To address these challenge, the additional element type Manifest has been defined which may be referenced by SignedInfo References. The syntax is defined as follows:

```
<Object>
    <Manifest>
        (Reference)
    </Manifest>
</Object>
```

Figure 6.31: W3C IETF XML SignedInfo Reference

Sometimes, additional information is needed to include assertions about how the signature was produced. Therefore, the SignatureProperties element contains information such as time of signing, ID of supporting components, etc.

Defining Core Syntax (Schema Definition, DTD) and Processing, but also the Algorithms required, recommended or optionally, the XML-Signature proposal supports the binding of any kind of information dealing with content, form, and context to the digital signature. Furthermore, it supports healthcare-relevant requirements like the endorsement reflecting the principal's role within the organisational framework.

6.14.2 The ETSI XML Advanced Digital Signatures Standard

Following the European Electronic Signature Directive 1999/93/EC discussed in detail in Chapter 7, additional requirements reflecting the different levels of electronic signature have to be met [CE, 1999]. Technical-organisational specifications necessary for the implementation of the Directive 1999/93/EC have been assigned to third bodies. On EU level such bodies are the Electronic Signature Committee set up under Article 9(1) of that Directive, the European Electronic Signature Standardization Initiative (EESSI), which work is carried out in close co-operation by the European Telecommunications Standards Institut (ETSI) and CEN/ISSS (Information Society Standardisation System).

The ETSI XML Advanced Digital Signatures (XAdES) published in the ETSI Technical Specification (TS) 101 903 introduce different levels of completeness and therefore independence of related services of XML DS. Starting with the XML Advanced Electronic Signature (XAdES) which includes the signature policy ID element, the XML Timestamped Electronic Signatue (XAdES-T) includes the Timestamp over XAdES. This signature is

completed by certificate and revocation references contained in the XML Complete Electronic Signature (XAdES-C) as shown in Figure 6.32.

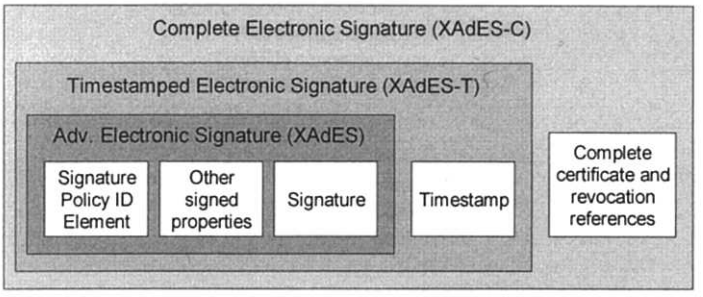


Figure 6.32: Components of the XML Complete Electronic Signature (after [ETSI, 2001])

Timestamping the XML Complete Electronic Signature or even the certification path as well as the revocation status references, the XML Extended Electronic Signature (XAdES-X) is provided, which can be enhanced by the certification path and revocation status data (Figure 6.33).

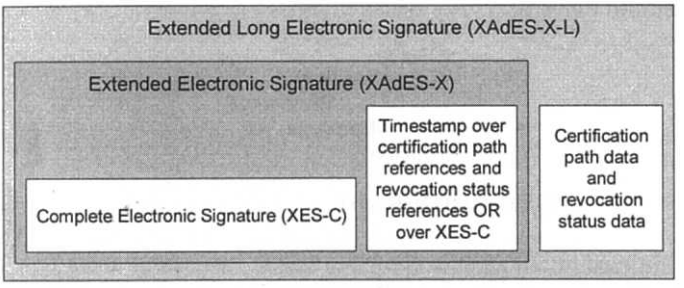


Figure 6.33: Components of the XML Extended Long Electronic Signature (after [ETSI, 2001])

Embedding the sequence over timestamps for archieved documents, the XML Archived Electronic Signature has been specified (Figure 6.34).

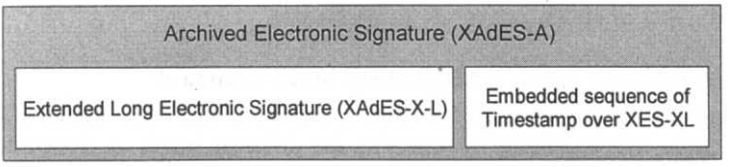


Figure 6.34: Components of the XML Archived Electronic Signature (after [ETSI, 2001])

The next figure presents the compact XML specification of the different levels of the ETSI XML Electronic Signatures.

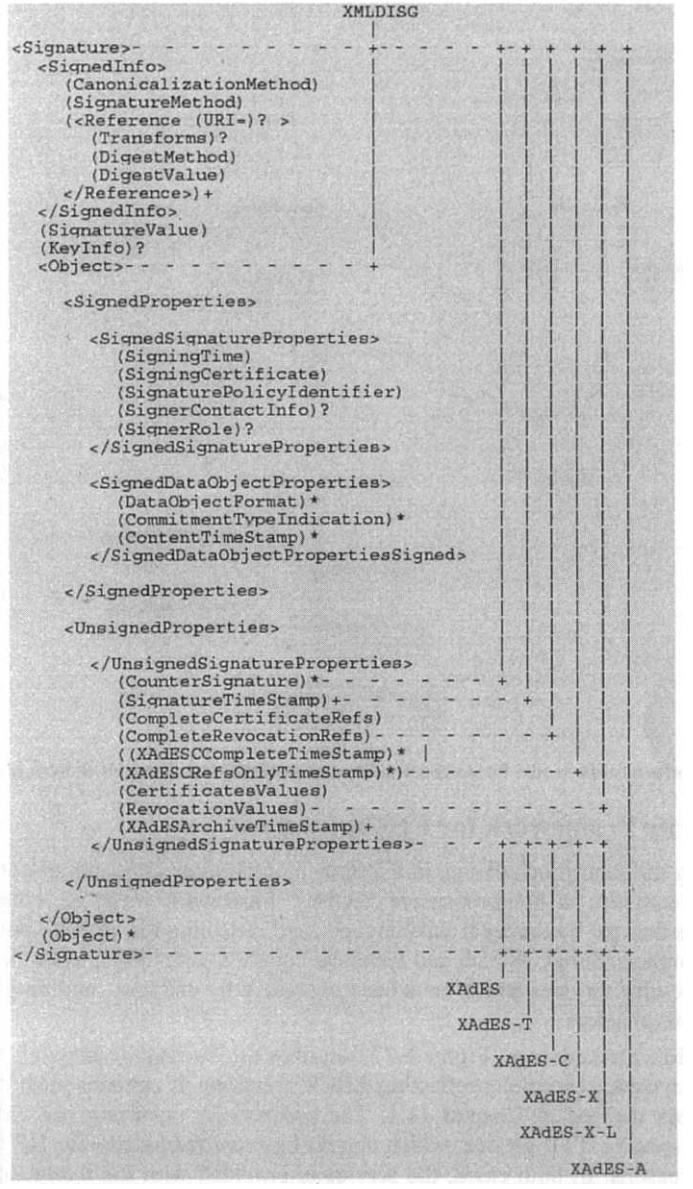


Figure 6.35: XML Specification of ETSI XML Electronic Signatures [ETSI, 2001]

6.15 Alternative Authorisation Models

Similarly to our authorisation and access control concept which has been refined within our ISHTAR work, recently a distribution rules framework has been developed by Robin Hopkins within the CEN EHCR Communications project, i.e. the original CEN ENV 13606 [CEN ENV 13606, Part 3]. However, there are differences in the basic model and therefore in the schema proposed, reflecting the individual security model instantiation we developed on the one side and in the generic frame of delegations in the CEN approach (Figure 6.36) on the other side.

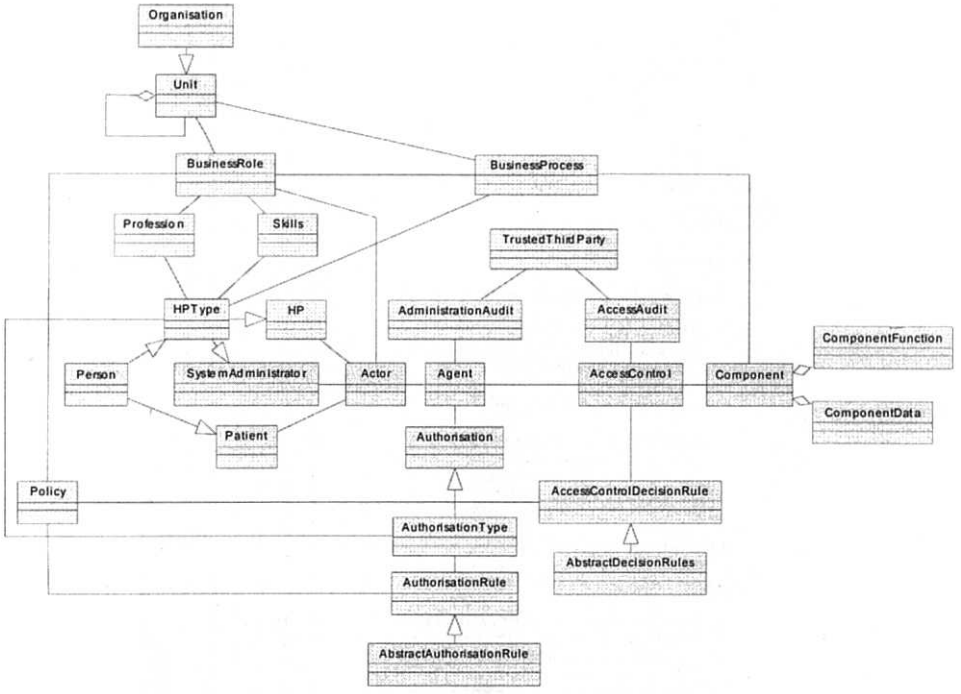


Figure 6.36: Information Model for Authorisation and Access Control in EHCR Systems

6.16 Security Framework for EHCR Systems

Summarising the security challenge in distributed, communicating and co-operating health information systems, a comprehensive security framework must be considered. This framework is defined by the legal fundamentals and reflecting the security policy agreed as well as the organisational, ethical, and technical conditions including rules, decision procedures and security services which have been expressed by use cases and roughly discussed in the previous chapters.

Using the UML methodology, Figure 6.37 describes this security framework for healthcare information systems, especially reflecting EHCR scenarios. It contains most of the abstract use case types defined in Chapter 11.1. The additionally occurring use case "Audit" is originally a specific TTP service, which should be provided locally for HP's privacy and acceptance reasons. In both cases, the service is provided with the interaction of either a TTP or a local system administrator and therefore not being introduced as a basic use case type. The same is happening with the TTP service "Notary's Functions" including, e.g., time stamping.

It must be mentioned, that the security services providing trustworthiness and privacy for patient's information and its communication often influence the HPs' privacy rights. Therefore, security services must be considered considering the often contrary interests of the different parties involved. Thus, information systems and the corresponding security policy can be developed and implemented only, including the HPs and their organisations (e.g., the works committee, which has to be established in German institutions) from the beginning.

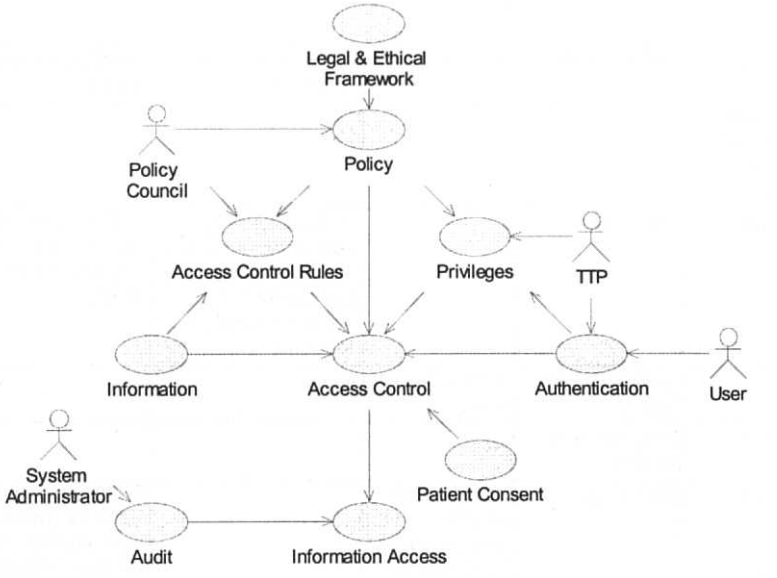


Figure 6.37: Security Framework to be Expressed in the Security Policy

6.16.1 TTP Use Cases

The consideration following describes the basic services for establishing and running a TTP, and is based on the general security scheme widely established. Hereby, a general distinction is formally made between all services belonging to purposes of communication security, and those belonging to application security issues. In some case it is possible that both communication security and application security may contain the same sort of service as, e.g., in the case of providing integrity by hash and digital signature means.

The TTP services may be separated into ***TTP services to establish a security infrastructure*** and ***TTP services to enable security services*** (see Table 6.5). The former ones, e.g. key generation and certificate issuance, are more or less static services (which can be described by use cases) whereas the latter ones, e.g. accessing repositories and checking CRLs, represent dynamic functions (described by process models). Therefore, after having established the infrastructure, the TTP has to provide only the second class of services. In the case of card-based certificates, the derivation of the user identifier needed for the application security services is independent of the TTP services as it is held on the card. Otherwise, the identifier is derived from directory-based certificates. In both cases, the application security services such as authorisation and access control are only dependent of the local use of the identifier. However in order to determine the current validity of the identifier, the TTP services are needed by the application security services.

Therefore, beside the services for establishing a security infrastructure (generating keys, certificates, etc.) only directory services (PK, different certificates), revocation services and notary's (auxiliary) services must be available. The use of the TTP services within the communication security and application security concept is presented in the table below.

Table 6.5: TTP Services

TTP Service	Data Object	Functionality	Communication Security Service	Application Security Service
TTP services to establish a security infrastructure				
Key generation	Key pairs		a) Authentication, accountability, integrity, confidentiality (decryption) of communicated information	e) Accountability, integrity, confidentiality (decryption) of stored information
Naming	Distinguished name	Identify the user		
Registration	Register information			
Certification	Certificates	Certification of user-property (key, role) relationship		f) Derived identifier to be used for (role-based) authorisation, access control, audit (alternative to g)
TTP services to enable security services				
Revocation	Certificate identifier	Revoke user privileges		
DIRECTORY	certificates, CRLs, Cross Certificates	Retrieve current information	b) Verification of authentication, accountability, integrity; c) Confidentiality (encryption) of communicated information	g) Derived identifier to be used for (role-based) authorisation, access control, audit (alternative to f); h) Verification of accountability, integrity of stored information; i) Confidentiality (encryption) of stored information
Notary's	Time stamp, ...		d) Notary's functions	k) Notary's functions

Based on the services mentioned above, the role of a TTP is to provide assurance and evidence about the correctness of information characterising the partners in communication and co-operation as well as to enable the security services.

The following diagrams describe the processes dealing with TTP services. Using the UML sequence diagram, Figure 6.38 demonstrates the sequence of activities for ordering and delivering a Health Professional Card. In that context, the most diversified structure has been assumed. In concrete implementations however, some of the objects involved may be aggregated (e.g. certification authority, key generation instance).

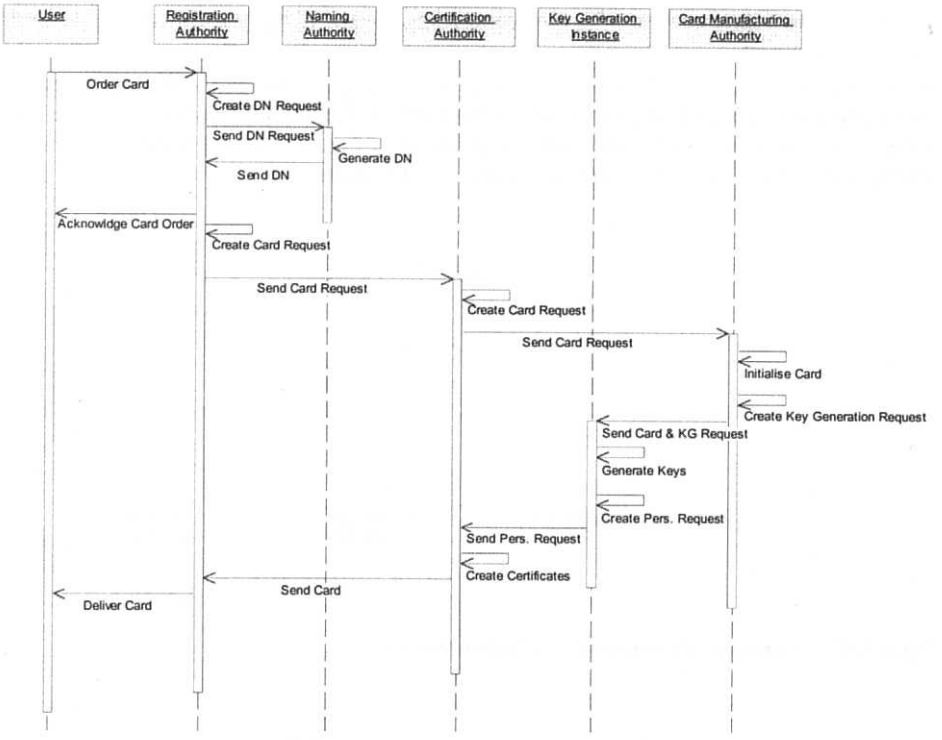


Figure 6.38: Sequence Diagram for Card Order and Delivery

Figure 6.39 presents the sequence diagram of revocation or replacement of cards or certificates respectively. Both diagrams describe the more static workflow of the security infrastructure management. More details on requirements and solutions for TTP services and their organisational set-up are elaborated within the European TrustHealth-2 project results [TRUSTHEALTH_WWW].

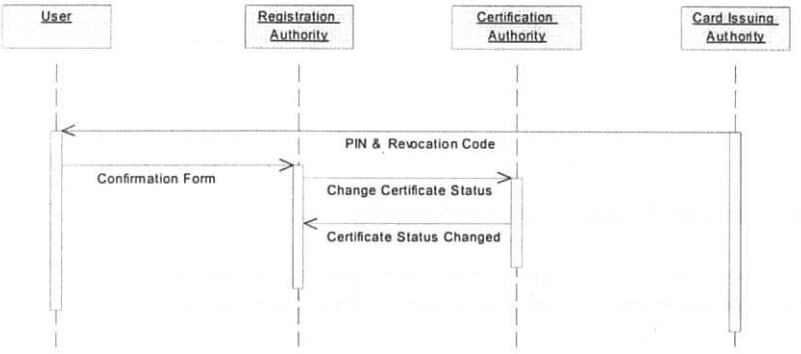


Figure 6.39: Card and Certificate Management

Based on the described management of security tokens and data needed for a secure healthcare application environment, the TTP-related services for communication security and application security may be provided. A basic service for most of the other security func-

tionality is the authentication of a user concerning his/her identity and his/her role in the healthcare business.

Regarding the authentication of user via HPC, two principle cases can be distinguished: the user's authentication to a local or a remote workstation. Figure 6.40 demonstrates the components involved in a local authentication procedure. Figure 6.41 shows the sequence of activities in context of a user authenticating to a local system.

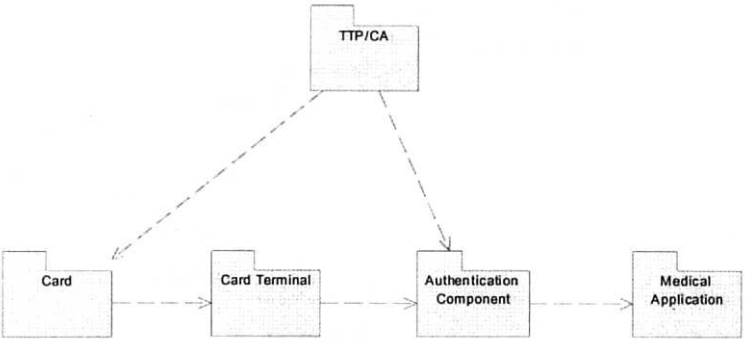


Figure 6.40: Component Diagram for Local Authentication

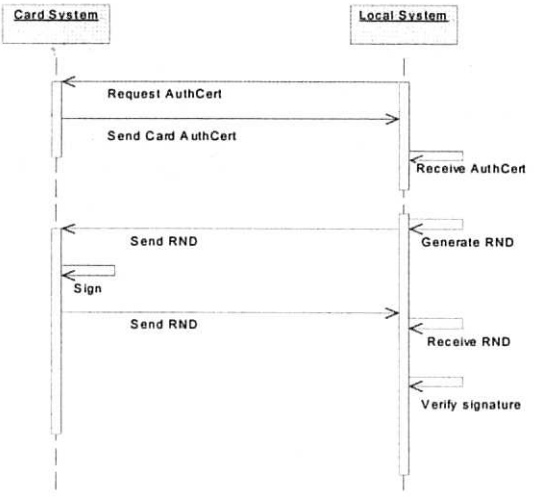


Figure 6.41: Sequence Diagram for Local Authentication

Figure 6.42 and Figure 6.43 represent the corresponding situation in the case of the user's authentication to a remote system like application or data servers etc.

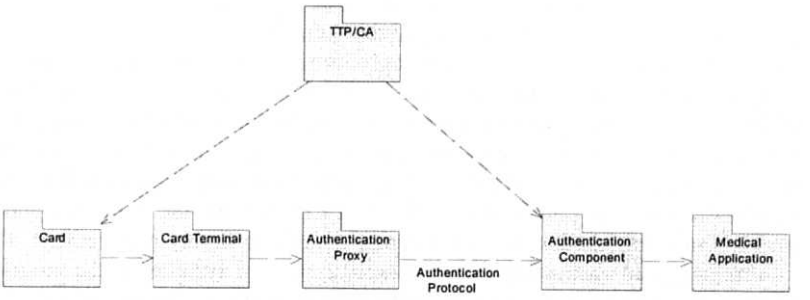


Figure 6.42: Component Diagram for Remote Authentication

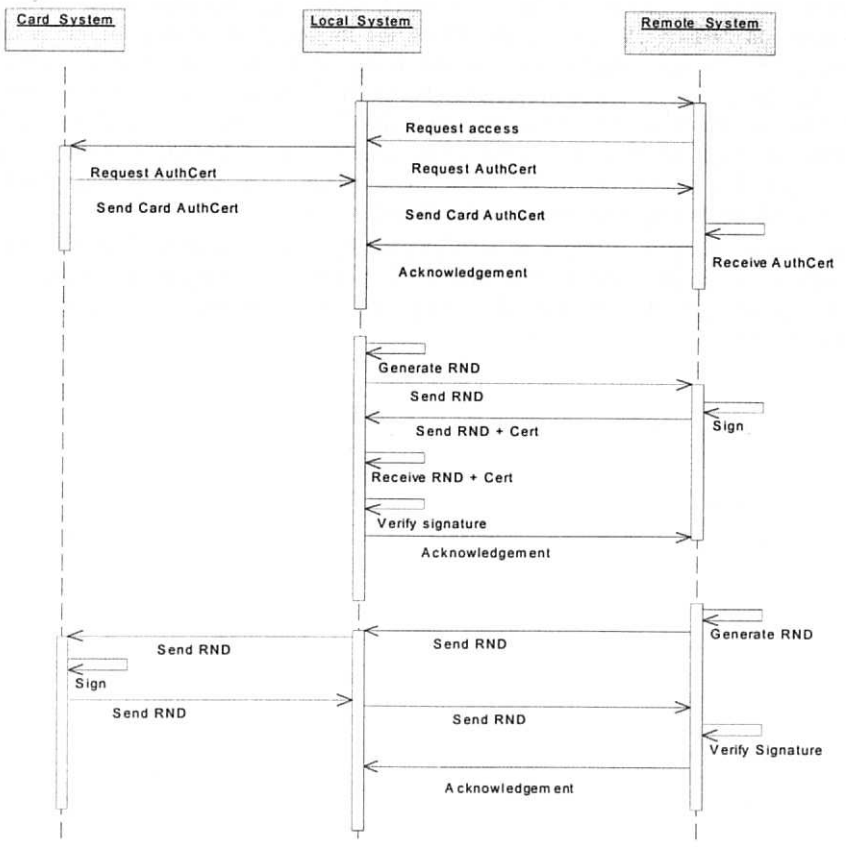


Figure 6.43: Sequence Diagram for Remote Authentication

6.17 Summary and Conclusions

A complete modelling of security for health information systems has been developed, which has not been provided elsewhere, to our knowledge. Based on the paradigms established as object-orientation, component architecture, UML methodology, and domain concepts, but also using their interpretation as well as specific models for health information systems' security, a framework for analysis and design of secure information systems for

health has been developed, implemented and presented. The tool-set facilitates the view of the different user groups involved, as HP, system administrators, and implementers. Analysing the real healthcare environment, a strongly restricted number of classes and of only 6 use case types for medical scenarios as well as 8 use case types for security-related scenarios could be defined enabling any real scenario by combination of these components. Using the generic approaches of a general security model and a layered security scheme, the comprehensive challenge of security enhanced health information systems could be simplified, selecting specific views or partial tasks only. In a LEGO type, the basic components defined can be combined together or with external models to fit any requirement. These efforts in formalising and abstracting from the holistic view of systems on the programmers' level facilitate all the processes from characterisation, analysis, design, specification, and implementation. By that way, not only the component-based thinking but also the component-based, i.e., step by step development and implementation is supported keeping the complex objectives and requirements in mind. Using the Rational Rose environment [Eriksson and Penker, 1998; Quatrani, 1999], the implementation of solutions is directly supported. By that way, security analysis and design for health information systems can be facilitated. In the framework of projects funded by the European Commission, the approach as well as the scenarios and abstract use cases developed are currently used and evaluated within extended pilots of secure interoperable HICS in several European countries' healthcare system. The importance of the users' involvement, their education and training has been derived and is supported by the methodology offered.

Comprehensive lists of activities as well as security services derived from the approach presented are available at the authors' site. UML models describing the security infrastructure needed are provided at other places (e.g. [TRUSTHEALTH_WWW] and are out of scope of the book.

7 Some Legal and Practical Aspects of Assessment and Use of the Results Achieved in Distributed Health Information Systems

7.1 Introduction

The *shared care* paradigm is the dominant paradigm for the health structure in developed countries around the world. Interoperable health information and communication systems are needed to enable this paradigm of caring the same patient by different persons from different organisations exploiting different methods at different time to provide optimal care for the patient's health and welfare. Dealing with personal medical information, distributed Electronic Health Record (EHR) systems fulfil the architectural requirements to support *shared care*. Such information is highly sensitive. Therefore, appropriate services and mechanisms for guaranteeing security and privacy must be provided by legal, organisational and technological means according to the policy agreed. More generally, paradigms and results presented in Chapter 5 will now be discussed in the context of practical aspects of *shared care* information systems. Some special issues as middleware security and security infrastructure, and open secure communication as well as solutions for a specific security environment like chipcard-based health information systems specified, developed and implemented by the Magdeburg Medical Informatics Department within European projects, will be presented in the next chapters.

7.2 Legal Aspects

Dealing with personal medical, i.e., highly sensitive information, *shared care* information systems require appropriate services and mechanisms for guaranteeing security and privacy by legal, organisational and technological means according to the policy agreed. Currently and in the future, the *shared care* paradigm will be realised crossing organisational and regional boundaries. Regarding the mobility of the citizens, the communication and co-operation on a European as well as an international scale, it must be considered even globally.

Considering the legal framework, a huge number of laws and paragraphs concern data protection and data security issues directly or indirectly. Therefore, the following consideration is restricted on some fundamentals of basic security services and related mechanisms needed. In that context, only authentication, integrity, confidentiality, accountability, authorisation, and security infrastructure services are considered. Beside the efforts of the European Parliament and the European Commission for harmonisation the legislation of the European Union member states, many differences may be found regarding the national and the international law. Thus, sometimes a separate discussion must be performed elucidating the German situation. Furthermore, the security framework highly depends on the concrete application scenario. For that reason, often the considerations and interpretations are referred to the specific situation of distributed EHCR systems as, e.g., a regionally distributed clinical cancer registry established at the Magdeburg Medical Informatics Department. Legal aspects establish an important framework for communicating and co-operating health information systems and its security [Blobel, 1996a-c; Blobel, 1997b]. This paper, however, deals with structural, organisational and especially technological issues.

Special publications provide more detailed information (e.g. [Laske, 1995; ISHTAR_WWW; TRUSTHEALTH_WWW; SIREN_WWW]).

As a result of the TrustHealth-2 project Workpackage 1, which has dealt with legal implications of security solutions based on HPC and TTP services and introduced in the healthcare domain, classifications of security issues have been made [Blobel and van Eecke, 1999]. The view on the European legislation in [Blobel and van Eecke, 1999] has been elaborated in responsibility of Patrick van Eecke. Based on the security models discussed in Chapter 5, Table 7.1 identifies legal implications sometimes combined with technical ones. Finally, Table 7.2 demonstrates legal interdependencies of technical measures.

Table 7.1: Legal Issues Classification

Legal issues		**Solution type**
Authentication	The process of reliably identifying a principal by securely associating an identifier and its authenticator who confirms the identity or to verify the eligibility of a station, application, or individual and the related roles	Technical Policies
Data protection	To make sure personal information is not disclosed	Policies Technical Legal
Electronic evidence	To make sure that electronic information has legal value	Technical Legal
Liability	To make sure that a liability scheme exists for malfunction and malpractice	Legal Policies

Table 7.2: Legal / Technical Issues Relation

Legal issues related to technical solutions		**Solution type**
Encryption	To make sure that possible restrictions on the use or export of encryption are being complied with	Legal Policies
Digital signatures	To make sure digital signatures are legally regarded as hand-written signatures	Legal
TTP	To make sure that the issuance of certificates is well organised	Legal Policies

Following, for internal and external security services the basic security services, the legal framework, legal requirements, or challenges for new legilations are reflected on the European legislation.

7.2.1 Peer Entity Authentication

Peer entity authentication provides the corroboration that a peer entity in an association is the one claimed. Authentication is provided identifying a principal by securely associating an identifier and its authenticator, who confirms the identity or to verify the eligibility of a station, application, or individual and the related roles. Peer entity authentication or shortly authentication provides the basis for most of the communication and application security services. The certified user's identity and his/her roles specifies the framework to collect, record, store, process, and transfer sensitive personal information as personal medical data according to the European Data Protection Directive [CE, 1995]. The European technical solution based on HPC needs the legal and organisational framework of TTP services. See also Chapter 7.2.4.

7.2.2 Data Protection

The protection of personal data when collected, processed and/or transferred is a legal requirement within the European Union. The European Union Directive 95/46/EC on the Protection of Individuals with Regard to the Processing of Personal Data and on the Free

Movement of such Data [CE, 1995] is currently implemented into the national legislation of the EU member states. This means that every EU member state has the same level of legal protection of personal data.

7.2.3 Data Confidentiality

Secure information transfer is a necessity when dealing with sensitive personal information such as patient records. Cryptography is a valid instrument to secure information from being disclosed to unauthorised parties.

Until recently, the legal situation of cryptography was unclear hindering the implementation of strong cryptography tools for national or cross-border information transmission. It is noteworthy, though, that since a few years the European Union as well as the diverse member states are explicitly conducting a policy in favour of strong cryptography. Even in the United States of America, a process of rethinking takes place now. In the 1997 Communication "Ensuring security and trust in electronic communication: Towards a European framework for digital Signatures and Encryption" the European Commission states the importance of security based on cryptography for doing business or conducting private communication on the Internet.

7.2.4 Electronic Authentication

The digital signature is commonly accepted as a basic mechanism for securing electronic information. The digital signature provides the mechanisms required for authentication services including peer entity authentication, authentication of data origin and receipt (accountability, non-repudiation, integrity check) in the context of both the communication and the application security concept. The fundamental question is: How meets the digital signature the national legislation of the EU member states as well as the other countries around the world?

There are several possible strategies to adapt the national legislation to the new technologies:

Some of the countries with Germany among others in the first line have introduced, are introducing, or consider the introduction of a specific legislation for ruling the digital signature and the security infrastructure needed (e.g. [Der Deutsche Bundestag, 1997]).

Other countries like Italy define the general equivalence of the digital signature and the hand-written one, if signature is legally required anywhere.

In some countries as Sweden, the digital signature is accepted for specific reasons ruled in specific legislation (sectoral equivalence).

Another group of countries including, e.g., Belgium proclaimed the equivalence in evidence of the digital signature on court.

The simplest way is the adaptation of the legislative framework by interpretation in doctrine (jurisprudence) and case law (jurisdiction), as done in the UK.

Regarding the restrictions, specialisations, exceptions defined in the different countries, the European Commission's initiative publishing a proposal for a European Parliament and Council Directive on a Community Framework for Electronic Signatures [EC, 1998b; CE, 1999] and its refinement was an important effort. Providing the fundamentals for the European Information Society Initiative, this directive establishes the basic principles of

- ensuring technological neutrality,
- avoiding any prior authorisation scheme, and
- recognising the legal validity of an electronic signature

In the same context, a European Parliament and Council Directive on Certain Legal Aspects of Electronic Commerce in the Internal Market (Electronic Commerce Directive) [EC, 1999] has been proposed.

The activities mentioned and others, culminating in the European Electronic Signature Standard Initiative (EESSI) provide the specification of a trustworthy environment needed to push electronic commerce but also specific businesses like *shared care* on a national and international scale.

7.2.5 Authorisation

As agreed in the policy and in strict accordance with the European Data Protection Directive [CE, 1995], authorisation defines rights and duties of an authenticated user (identity and the related role) (see also Chapter 7.2.1) according to the rules applied on identity and roles. Authorisation concerns detailed rights and duties for functions and data to collect, record, store, process, and transfer sensitive information. Authorisation depends on organisation-related as well as function-related user roles. In accordance with the European and the national legislation, the functional roles are dominant.

7.2.6 Access Control

Access control specifies rights to access systems, applications, components, or objects in the application security context according to the European Data Protection Directive [CE, 1995]. Access control is depending on organisational and functional relationships. In accordance with the European and the national legislation, the functional roles are dominant.

7.2.7 TTP Rules

The introduction of digital signature technology linked to a *public key infrastructure* (PKI) and certificate schemes necessitates the use of TTP structures and their services. A TTP is a structure centralised or decentralised, locally or remotely, or a mixture of them to provide the TTP services. These services comprise naming, registration, certification, key management, directory services, certificate revocation list (CRL) management. The services mentioned are provided by corresponding authorities as naming authorities, registration authorities, certification authorities, etc.

In the different countries, TTP structure, environment, working conditions and functionalities are ruled on different ways. In Germany, specific legislation has been established (see Chapter 7.2.8).

The European Commission recently pushed the European Electronic Signature Directive [CE, 1999; EC, 1998a,b] specifying a common legal framework of rules, conditions, and functions of TTP for secure use of electronic signatures and their legal recognition. Contrary to the German legislation, the Directive states that member states may not make the provision of certification services subject to prior licensing. The efforts provide the basics to deploy the new technologies in a commonly acceptable and trustworthy way.

7.2.8 German Organisational and Legal Obligations

Regarding the transfer of patient-related data from the legal point of view, two legal fundamentals must be considered restricting the disclosure of patients' personal, i.e. identifiable information. The first one is the legal request for physicians' secrecy about the patients' information. The second one is the federal German Data Protection Law and the corresponding Data Protection Laws of the German states widely agreeing with the European Data Protection Directive [CE, 1995]. The physicians' secrecy corresponds to the physician's right to refuse testimony in criminal proceedings, Criminal Case Order, § 53. It is

expressively ruled in the German Criminal Law, § 203, as well as in the physicians' professional order template, § 9.

The German Criminal Law, § 203, defines:

Who disclose unauthorised a strange secrecy, especially a secrecy dealing with a person's sphere, an organisational or business secrecy, which was revealed to him/her as a physician or coming to know, will be prosecuted with up to one year jail or fined.

The transfer of patient's data is only allowed

- if this transfer is based on legal requirements,
- if there is a specific reason (e.g., the protection of the data subject's or other parties' life or health) or
- if the informed patient gave his/her (written) consent.

According to the Data Protection Law, recording, processing and transfer of patient's data is defined in the care contract and only allowed if it is needed to provide the care. If the record, processing and transfer of personal information is not really needed for care, the patient's consent is requested.

Furthermore, the patients' rights for information about, correction and restriction of his data stored, processed and transferred are clearly defined.

In many cases, the information management is based on the written consent of the patient registered in our tumour documentation system.

At the moment, there are no legally binding rules for on-line transfer of personal data. However, according to the laws and legislation mentioned, every physician is responsible to guarantee the physician's discretion. Therefore, he must take care for confidentiality, but also for accuracy, integrity of information communicated to authorised users only. Authorised are only Health Professionals directly involved in the patient's care. The information has to be restricted in content and time according to the "Need to Know"-principle.

Using public lines, the confidentiality, integrity, availability, accountability incl. non-repudiation must be provided by appropriate measures.

The Information and Communication Services Law [Der Deutsche Bundestag, 1997] as a framework legislation as well as the corresponding detailed legislation like the German Digital Signature Law [Der Deutsche Bundestag, 1997] provided the background needed to establish an infrastructure for a secure healthcare environment. The transfer of personal medical information across the border however, is currently not allowed by the German law. The European directives on data protection and digital signatures could provide a framework to handle these requirements beside the patient's consent rule.

The use of strong encryption in Germany is unlimited. Contrary to the United States, in Germany the encryption of computer data will not be restricted. The federal parliament decided in June 1999 not to restrict the development and marketing of cryptographic techniques. In a framework catalogue it is mentioned that only the distribution of secure encipherment systems may guarantee the security of business secrets and other sensitive data within computer networks. This guarantee is the crucial basis for development electronic commerce. After two years, the situation should be evaluated again. This has not been done until mid of 2002, however.

Nevertheless, several security agencies demanded the federal government, to avoid the general availability of encryption programs. They fear that the persecution of crime in networks like the Internet would be complicated by such a way.

7.2.9 The European Technical and Legal Security Framework at the Glance

As shown by projects like Trusthealth-2 [Blobel and van Eecke, 1999], the legal framework for trustworthy communication and co-operation is already available or under development in the European Union member states as well as in several other countries around the world. Nevertheless, some agreement on technical, legal, and policy level are missed yet. The TrustHealth project [TRUSTHEALTH_WWW], but also standardisation efforts of CEN TC 251 are specifying structures, protocols, and functions of securely communicating and co-operating health information systems. This includes also security services, e.g., the work on Secure User Identification – Strong Authentication Using Microprocessor Cards [CEN ENV 13729].

In that sense, within the TrustHealth deliverable 1.2 certain recommendations for legal, organisational, and technical requirements, rules, and solutions have been made. This concerns especially the services authentication of principals as well as data protection and confidentiality, but also TTP rules. See [Blobel and van Eecke, 1999] for further information.

In [Blobel and van Eecke, 1999], Patrick van Eecke proposes a security guidelines handbook including amongst others the policies to be followed on data protection, data confidentiality and authentication. This concerns a common security policy specifying, e.g., the legal, organisational and social business framework, the analysed threats, accepted risks and intended organisational and technical solutions, but also the TTP policy. If systems of different organisational and/or policy domain communicate, policy bridging is required. The policy agreed defines legal, organisational and technical security issues and the functionality permitted. Table 7.3 gives an overview on an international framework for security policies regarding both technical and legal aspects.

Table 7.3: A European TTP Policy Legislation Framework (after [Blobel and van Eecke, 1999])

Common TTP policy	
⇓	
Legal issues	**Technical issues**
Based on the Electronic signature directive	Based on the EESSI electronic signature standard
⇓	⇓
Legal coherence with European rules	Technical coherence with European (international) standards
⇓	⇓
Legal coherence with national rules, i.e. legal interoperability	Technical coherence with standards, i.e. technical interoperability

7.3 Alternative Approaches to a Security Concept

The chapters presented dealt with the development of a systematic methodology analysis, design, and implementation of secure distributed open health information systems. This approach overcomes many of the weaknesses other methodologies have. Such weaknesses are, e.g.,

- the inability to facilitate different user groups' views in an easy, simplified way, providing continuous and consistent models for different levels of granularity (from the complex system up to modules, objects, instructions) and different levels of abstraction (business, logical, technical components),
- gaps and breaks in the paradigms and methods deployed,
- inconsistencies in the tools used for analysis, design, development, implementation, and maintenance.

- the orientation on technical requirements and solutions, thereby neglecting of business issues.

Examples for such approaches, the proposed methodology provides alternatives for, are the CRAMM toolset and also some interesting alternatives, which are, unfortunately, not complex and complete enough to meet our challenge, as the SIDERO tool [Flikkenschild et al., 1996]

Another example for developing a security concept has been derived within the MEDICUS-2 research teleradiology project [Baur et al., 1996], strictly following the IT Security Manual [BSI, 1995]. They divided this process in four major steps which can be considered general:

1. Determination of the appropriate level of security: Assessing the value of objects to be protected, by estimation of the damage induced by loss of confidentiality, availability and integrity.
2. Threat analysis: Listing all objects involved in the operation of the system, identifying the threats relevant for each object, determining resulting weaknesses for the system (according to the detailed list given in [EC, 1994]).
3. Risk analysis: Quantify the resulting damage for each object. Estimate the frequency of those damages. Identify the not tolerable risks for all pairs of object value together with damage frequency. Each such pair (object value, damage frequency) is classified as 'tolerable' or 'not tolerable'.
4. Developing the security concept: Appropriate measurements are chosen which either reduce the damage frequency or restrict the damage to tolerable levels. Their consequences due to costs, effect, and operational feasibility are examined. The approach is successful if, after applying acceptable measurements, the remaining risk is on a tolerable level, according to the criteria applied.

Following, a tabulated overview on a number of security solutions is given.

Table 7.4: Threats, Security Services, and Solutions

Threats	Security Services	Solutions	Remarks
unauthorised use of authorised services	identification and authentication	Password	authentication by knowledge
		chip card with PIN or biometrics	authentication by ownership and knowledge or properties, public key security mechanism
manipulation of information	integrity check of information and non-repudiation of origin	Hash algorithm and digital signature of sender	public key security mechanism (e.g. via chip card)
concealment of information origin	non-repudiation of information origin	digital signature of sender	public key security mechanism (e.g. via chip card)
repudiation of receipt	non-repudiation of information receipt	digital signature of recipient	public key security mechanism (e.g. via chip card)
breach of confidentiality	encryption	symmetric or/and public key algorithms	end-to-end encryption (user-related e.g. via chip card) and link-by-link encryption (user-independent between network components)

Threats	Security Services	Solutions	Remarks
unauthorised use, manipulation of infor-mation, breach of confidentiality for unauthorised services	encryption and authentication	Pretty Good Privacy (PGP)	ensures e-mail, certification by super user
		Privacy Enhanced Mail (PEM)	ensures e-mail
	encryption and authentication of server and client	Secure Socket Layer (SSL)	ensures channel-related WWW, FTP, Telnet
unauthorised user, manipulation of infor-mation, breach of confidentiality and repudiation for unauthorised services	encryption, authentication and non-repudiation	Secure Hyper-Text Transfer Protocol (S-HTTP)	ensures document-related WWW
external unauthorised access to internal resources	firewall	several products with different functionalities	inhibits the direct connection between the internal and the external environment (filter), manages addresses and access rights, can be combined with virus scanners

7.4 Categories of Communication and their Security Requirements

In the next sections, the communication services classified in Chapter 2 will be shortly discussed from the security services point of view. An extended presentation and evaluation of these issues can be found in [Blobel et al., 1997; Baum-Waidner et al., 1998].

7.4.1 Simple Communication Services

7.4.1.1 Remote Access

Remote terminal access allows individuals to use a remote system as if it would be a directly attached terminal to a computer system. *Telnet* is the standard for remote terminal access. It provides access to character-based applications only, not including graphics. Another wide-spread protocol with similarities to Telnet is the *Kermit* protocol. In contrast to the *remote login* (*rlogin*) and Telnet procedures the *single sign-on* or *single logon* functionality is even more critical. The single sign-on provides the one-step user access to all system-related applications. After a successful single logon all data and functions of all involved applications are open for both the authorised and the unauthorised user. Therefore, all remote access procedures need both trustworthy identification and authentication procedures. Confidentiality as another challenge can be provided by the channel-based *Secure Socket Layer (SSL)* or its successor, the *Transport Layer Security (TLS)* protocol.

7.4.1.2 File Transfer Protocol

A common method for transferring files is the *File Transfer Protocol* (*FTP*). It allows users to send files and to get files over the network. In healthcare the ftp service may be used to receive, and provide access, to certain documents, annual reports, conference proceedings and promotional material over the Internet. FTP security solutions are available and have been specified as one solution to enhance EDI security [Blobel et al, 1998a,b].

Further lower level services are the common simple procedure and function calls like *remote procedure call* (*rpc*), *remote SQL procedure calls* (*rSQLpc*). Confidentiality as important challenge can be provided by the channel-based *Secure Socket Layer (SSL)* or its successor, the *Transport Layer Security (TLS)* protocol.

7.4.2 Advanced Communication Services

Advanced services for common purposes are *Electronic mail* (*email*), *World Wide Web* (*WWW*), *Gopher*, and *WAIS*.

7.4.2.1 Email

Email is very popular and one of the first networked services, especially also within the Internet. The Internet standard protocol for sending and receiving mail is the *Simple Mail Transfer Protocol* (*SMTP*). An Email transfers data, typically in human-readable form and limited size. Email may be helpful in sharing information internally and externally including world-wide communication. Mail procedures need trustworthy identification, authentication as well as confidential communication. Such security services may be provided by products like *Pretty Good Privacy (PGP)* or *Privacy Enhanced Mail* (PEM) for authentication and by wrapped messages (secure messaging) or by hardware-based encryption on transport layer or by SSL for confidentiality. Within the EU MEDSEC project, EDI security has been investigated, specified and implemented [MEDSEC_WWW]. The results are currently under standardisation.

7.4.2.2 World Wide Web

WWW is a new, entirely Internet-based concept connecting hundreds of thousands of WWW servers world-wide. It is the driving force for the recent explosion of the Internet activities and makes it very popular. The Web uses the *HyperText Transfer Protocol* (*HTTP*) as the hypertext technology to link together a web of text, graphic images, sound, video and other data. In addition, hypertext provides facilities to navigate interactively and world-wide from one document to another, each hosted by an arbitrary server. The Web browser on the client side provides an easy to use graphical user interface to access information from any WWW server interactively.

With *Java*, not only information in the form of Web pages but also small programs called applets can be downloaded from the Internet. Viewing a page with an embedded applet requires a Java-enabled WWW browser that downloads the applet to the local system and executes it. An applet is a piece of code running on the local computer and applets can create animations and interactive programs on the WWW pages. These animations and interactive programs are characterised by a new dimension of interactivity, depending only on the processor power, not on the limited bandwidth of the Internet.

It should be noted that WWW services like Java applets or postscript files can carry Trojan horses or viruses and they should be handled with care because of possible "side-effects" in terms of the possibility of damage to the system services or data. The main security requirements of confidentiality, authentication and accountability may be provided by the document-based *Secure HyperText Transport Protocol (S-HTTP)*. Overviews on security issues for the Internet and WWW are given in [CTR, 1996; Stallings, 1995].

7.5 Application Security Services

The security requirements concern all security services defined within the concepts of both the application security and the communication security. The application security requires authorisation, access control to, as well as integrity, confidentiality and availability of

stored and processed information, accountability for data and procedures, audit, and notary's functions (like certified date/time). The communication security deals with strong authentication of the principals involved, integrity, confidentiality, availability and accountability (including non-repudiation of origin and receipt) of information communicated and notary's functions.

Most of the security services and mechanisms are related to the secure identification of the communicating and co-operating users. Therefore, the secure authentication is the basis for all other services. This authentication concerns the identity but also other important properties of the principals controlling those other services mentioned. Such properties could be the user's profession, qualification, special domains of interest, functional rights, etc.

Within the TrustHealth project funded by the European Commission, a security infrastructure including Health Professional Cards (HPC) and related Trusted Third Party (TTP) services has been specified and and evaluated by large scale test sites in 6 European countries. The HPC serves as authentication token bearing the secret keys for authentication, digital signature and encryption to exchange a session key securely. Furthermore, it contains several certificates according to the X509v3 standard as the ID certificate (authentication certificate), the digital signature certificate, but also some sets of attribute certificates. One set deals with professions, qualifications, capabilities and skills. These certificates may be standardised internationally enabling transborder communication. Another one is related to permission and legitimacy, which are mostly restricted to a country or even to a region, given to the card holder. In Germany, an extended specification of HPC including the certificates needed is now ready for use [HCP-Protocol_WWW].

The certificates are used to verify classified agreed roles of Health Professionals (HP). Based on the certificates including the first set of attribute certificates, the HP roles within the organisational and the functional framework of the Health Care Establishment (HCE) respectively can be specified. The first one is rather static defining the responsibility in the HCE's hierarchy. The latter one is highly dynamic expressing the HP-patient relationship. This role is dominantly influencing the rule-based decision for access control and authorisation according to the EU Data Protection Directive [CE, 1995] and the Council of Europe Recommendation on the Protection of Medical Data [CM, 1997]. For some more details see Chapter 9. Beside the users' roles, rule-based decision support systems also consider the classification of the EHR component needed according to the international classification levels "unclassified", "confidential", "secret" and "top secret".

The rules may be presented by access control or authorisation models respectively [Abrams et al., 1995]. These models are based on the objectives and restrictions covered by the policy of the domain considered. Communication and co-operation between domains require policy bridging.

7.5.1 Basic Access Models

As demonstrated for basic security services facilitating secure communication and co-operation within and between health information systems, which have been specified, developed, and implemented using the Chapter 5 fundamentals and principles, also the implementation of the application security services access control and authorisation requires refinements in granularity and abstraction level of the underlying models. Because these services are not the mainstream for the Magdeburg Medical Informatics Department's research and development, only the related logical models will be mentioned, but not derived based on the methodology proposed. The same is true for the modelling of rules presented in the next section to provide decision mechanisms.

The classic access models based on that or a similar dimensional framework and used in different domains are [Abrams et al., 1995]:

- The Clark-Wilson Model and the Chinese-Wall Model
- The Information Flow Model
- The Role-Based Access Control Model[30]
- The Role-Based Access Control and Information Flow Model

Furthermore, there are several approaches for access control in the Internet/Intranet environment. Examples for such solutions are:

- A Role-Based Access Control for Intranet Security
- A Distributed Authorisation Model for WWW
- A Role-Based Access Control for WWW
- A Lattice-Based Access Control Model for Document Structure.

Beside these solutions dealing with access control or authorisation, further approaches to support application security have been developed, as:

- A Security Model for Co-operative Work
- A Communication Agreement Framework for Access/Action Control
- Rule Set Based Access Control

In general, security services may be managed centrally or de-centrally. They could be organised at a global or at a local scale. Due to the global or at least domain-related character of authentication services, the secure identification/authentication must be realised globally using centralised or hierarchically de-centralised TTP services. On the other hand, the HP's accountability and liability for personal medical information on behalf of the patient requires a local authorisation and access control management using rather decentralised management facilities.

Most of the access control and authorisation mechanisms especially for distributed nodes are based on Access Control Lists (ACL). In larger populations, such an ACL is hardly to define and to maintain. Often, specific authorisation servers are applied, however mostly serving as a bottleneck of the system. An appropriate solution could be a distributed authorisation scheme using the capabilities specified for the documents and their contents in the sense of a Capability-Based Authorisation Model. An example of how to deal with exclusive roles in such authorisation schemes is given in Chapter 6.

7.5.2 Security Rules

Considering the security services authorisation and access control, some investigations and modelling efforts have been provided [Blobel, 1996d] which are based on the related basic scheme in Figure 7.1. In simple cases, the underlying security rules can be expressed as quadruple [Abrams et al., 1995]

(s, o, t, p) (17)

where s = subject (principal), o = object (information), t = type of access right, and p = (optional) predicate. Often, however, additional dimensions as a = authoriser subject and f = flag for the subject's role (function) must be considered expanding the quadruple to

(s, o, t, p, a, f) (18)

[30] Because being a member of a team could be a principal's role, RBAC covers the Team-Based Access Control (TBAC) model sometimes used too.

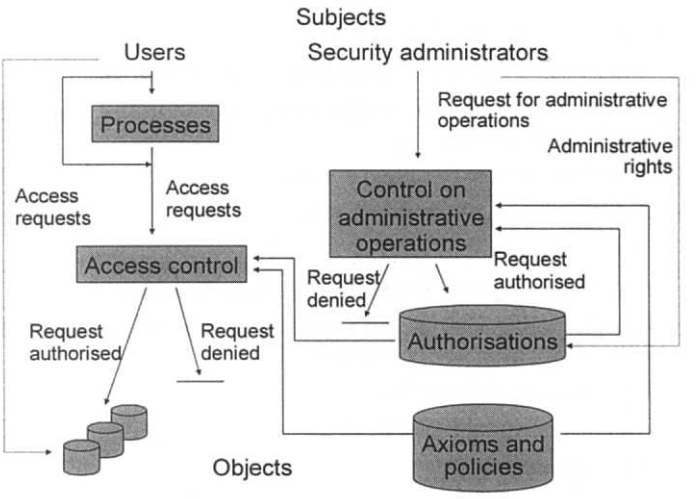


Figure 7.1: Basic Scheme of Authorisation and Access Control (after [Castano et al., 1995])

Combining and generalising both the mandatory and the discretionary security model, the role-based access control (RBAC) is an alternative to traditional discretionary (DAC) and mandatory access control (MAC) policies also manageable for larger user communities not knowing each other as in the *shared care* or the electronic commerce world.

A real world problem is the circumstance, that a user can be authorised for two or even more roles excluding each other. In RBAC, a static and a dynamic separation of duties can be defined to administer this challenge by excluding the case of exclusive roles of a user or by constraining the simultaneous activation of such roles, respectively. Both cases may be expressed logically through the following relations [Barkley et al.]:

Case 1: Exclusion of competitive exclusive user roles

$$\forall u{:}user,\ r_{ij}{:}roles,\ i \neq j$$
$$u \in role\text{-}memberships(r_i) \wedge u \in role\text{-}memberships(r_j)$$
$$\rightarrow r_j \notin mutually\text{-}exclusive\text{-}authorisation\ (r_j)$$

Case 2: Constraining the simultaneous activation of competitive exclusive user roles

$$\forall s_x, s_y{:}subject,\ ri, rj{:}role$$
$$r_i \in active\text{-}roles(s_x) \wedge u \in active\text{-}roles(s_y)$$
$$\wedge\ r_j \in mutually\text{-}exclusive\text{-}activation(r_j)$$
$$\rightarrow subject\text{-}user(s_x) \neq subject\text{-}user(s_y)$$

Figure 7.2: Management of Exclusive Roles

Another interesting approach recently published looks for a more detailed consideration of authorisation defining both role-based and content-dependent authorisation [O, 1999]. It is based on the assumption of rather static organisational roles and highly dynamic functional roles alike as developed in investigations of the monograph's author in the early nineties (see for reference Chapter 0). O's model specifies an organisational authorisation reflecting the user's role which depends on his/her position in the unit and the unit's role. The functional authorisation deals with the function-dependent as well as content-dependent authorisation comprising functional and issue-related roles, the objects managed, and the transaction performed as well.

Analogous to Figure 7.2, role-based and user-based authorisation must be enabled, both demanding strong user authentication provided by an appropriate security infrastructure (see Chapter 4). This authorisation granted must be controlled, sometimes including content-depending mechanisms.

According to their roles, organisational authorisation is assigned to a Health Professional, which is rather stable for a long period of time and defined by

User_role (user, role, unit, validity_flag) ←
User_position (user, position, unit) ∧
Unit_role (role, position, unit, object_type, actor)

According to the functional role, a role-based authorisation can be specified as

Role_authorisation (role, unit, object_type, transaction, state) ←
Unit_role (role, position, unit, object_type, actor) ∧
Transaction_mode (object_type, transaction, state, action)

Finally, OMG offered in its newest Security Service Specification V 1.7 [CORBA_SSS, 2001] a schema adapted to OMG's open architecture which is shortly described now.

If there is given a list of granted rights, G, and a list of required rights, R, the definition of the SecAllRights combinator forms the following predicate:

$$\forall r \ni r \in R \Rightarrow r \in G$$

The definition of the SecAnyRights combinator forms the following predicate:

$$\exists r \ni r \in R \wedge r \in G$$

These definitions have important ramifications when an empty list of required rights is specified with each combinator. Regardless of the granted rights, if the required rights, R, is empty, then the predicate formed with the SecAllRights combinator results in *true*, and the predicate formed with the SecAnyRights combinator results in *false*.

Note that the following behaviors of systems conforming to CORBA Security

- Assignment of initial required rights to newly created interfaces
- Inheritance of required rights by newly created derived interfaces

are unspecified and therefore may be implementation-dependent.

For more details, see [CORBA_SSS, 2001]

7.6 Summary and Conclusions

Supporting the *shared care* paradigm, interoperable and distributed health information systems which deal with sensitive personal health information require appropriate security services guaranteeing both communication and application security. Thereby, authentication as a basic service needed for most of the other security services and mechanisms can be provided by the European security infrastructure based on HPC and TTP services. Electronic Health Care Record (EHCR) systems are a central application scenario for such information systems. Contrary to authentication and other communication security services centrally controlled, authorisation and access control must be decided and managed locally. Based on specified roles and rules according to the policy agreed, de-centralised approaches are more and more used. This trend is supported by the Internet technology. Practical examples for specifying and implementing appropriate solutions are presented in the next chapter.

The legal framework for secure health information systems has been discussed and refined access control models have been presented within the proposed systematic and generic methodology.

8 Security Models for Open Architecture Concepts

8.1 CORBA Conceptual Scheme in the Context of Security Concepts

In distributed co-operating information systems, the underlying middleware provides also some integrative functions. For example, the envisaged CORBA vertical facilities *Patient Identification Service* [OMG, 1996d], harmonising the patient identification in different applications and meanwhile generalised to the *CORBA Person Identification Service* (PIDS) [CORBA_PIDS, 2001], and *Lexicon Query Service* [OMG, 1997b], supporting and managing terminology and semantics between different systems, provide functionalities supporting the intraorganisational or interorganisational interoperability of different information system components, renewed in 2000 [CORBA_LQS, 2000]. Another important CORBA specification related to health is the *CORBA Clinical Observations Access Service* (COAS) Specification, which is dealing with any information that has been captured about a single patient's medical/physical state and relevant context information renewed in 2001 [CORBA_COAS, 2001]. Context information relevant to health professionals such as patient demographics, observation types and data formats, etc., is provided via an Access-Component interface. Since those facilities will support such essential medical functions as electronic health records, archiving systems, clinical or epidemiological registries, they must ensure an adequate level of security.

In Chapter 3, the concepts of the mostly implemented middleware approaches have been analysed and interpreted. Integrating security services considered from the view point of the concepts of application and communication security, Figure 3.1 converts to Figure 8.1. Application security services interact with the objects via the interceptor mechanism. In comparison, communication security services interact with the interfaces using the same interceptor mechanism.

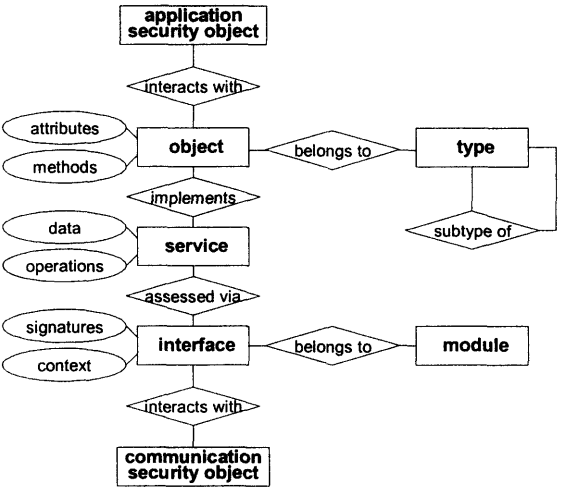


Figure 8.1: Security Services in the Basic Concepts of CORBA

8.2 Security Features Available in CORBA

Security protects information systems from unauthorised attempts to access information or to interfere with their operation. Though the need for incorporating security services into CORBA has been recognised rather early [OMG, 1991; OMG, 1994], it was only in 1996

that a comprehensive specification of the proposed security services has become part of the adapted *Common Object Services Specification* (COSS) [OMG, 1997a]. This *CORBA Security Services Specification* (SSS) has been published in the latest version 1.7 in March 2001 [CORBA_SSS, 2001]. Containing an own chapter for *Common Secure Interoperability* (CSI) features, it refers to the *CORBA Common Secure Interoperability V2 Specification* published in July of the same year [CORBA_SIS, 2001]. Simplicity, consistency across the distributed co-operating systems, scalability and usability (transparency), flexibility of security policies, independence of security technology, application portability, interoperability, and sufficient performance were defined as goals for an object-oriented security architecture within CORBA. Specific security goals the CORBA SSS has to meet are the reflection of regulatory requirements as well as of evaluation criteria for assurance (e.g. ITSEC, TCSEC, Common Criteria). Fulfilling these goals, CORBA transparently provides the required security to users and applications at least on the level of their own environment. In addition, the CORBA security services are also available to security unaware applications. The CORBA security specification framework meets the general security model and its refinements separating the communication security concept mainly represented as secure invocation and the application security concept focussing the object's behaviour (e.g. access control decision). Secure invocation comprises establishing security association (authentication, credentials, secure communication) and message protection (integrity, confidentiality). Also the other services mentioned above are placed in the conceptual framework.

Talking about CORBA security services, COSS *Basic security objects*, *Domain-unspecific security objects*, *Domain-specific security objects*, and CORBA *External security objects* have to be distinguished.

CORBA provides all important security services, such as identification and authentication, authorisation and access control, security auditing, security of communication including mutual authentication of clients and targets, integrity protection and confidentiality protection, non-repudiation, and administration of security [OMG, 1997c; CORBA_SSS, 2001]. Security is defined for domains differing from the point of view of organisational or legal conditions (security policy domains), institutional boundaries (security environment domains), or the technology platforms (security technology domains). Security is pervasive; it pertains to various components of the CORBA architecture. A considerable part of security functions is implemented directly through the ORB or through their bridging mechanisms. Others are confined to transaction services or to additional security services, implemented through specific security-related objects. Finally, security services are also provided by the underlying operation systems and communication services.

Each object service is ultimately requested on behalf of a principal, i.e. an end-user known to the system and separately accountable for the requests it initiates. Unless the principal has been already trustworthy authenticated outside the system (see next section), its authentication is performed by the *Principal authenticator* object, associated to each ORB providing a higher level of security. The *Principal authenticator* creates for each principal a *Credentials object*, containing the *Principal's privilege attributes*, e.g. the access identity, groups to which the principal belongs, roles, security clearance, and capabilities concerning various groups of objects. A security aware target application may obtain attributes of the principal responsible for the incoming request, to make its own authentication-depending access decisions. The information contained in *Credentials* can be obtained either directly or through the *Current*, an interface of the *Transaction services*, which holds reference to the current execution context at both client and target objects.

The privilege attributes are first needed for making a secure invocation, which is mediated by the ORB. Whether the invocation can take place, as well as the way in which it is medi-

ated, depends on the client and target security policies. Security policies concern such issues as access control, establishing trust in client/target, protection of messages for integrity/confidentiality, time restrictions, or delegation of privileges. If a request initiates a chain of invocations, then the security policies of all objects in the chain are taken into consideration by delegation mechanisms, including all intermediate objects.

As far as access control is concerned, applications can enforce their own access policies. Typically, details of access control are isolated from the application itself, and are implemented through an *Access decision object*, specific to the access policy. In addition, there is an *Access decision object* associated with the ORB and used for the *Invocation access policy*, which is enforced internally by the ORB. The decision whether to allow access to a given function or data depends on the privilege attributes of the initiator of the request, control attributes of the target, and on the execution context. Access policy can be actually shared by a whole domain of objects with similar security requirements. In that case, reference to the corresponding *Access decision object* is available via the *Current interface*.

Similarly, applications can also enforce their own audit policies, which can be again managed via a domain structure. Each application writes its audit records to an Audit Channel object. One such object is created at ORB initialisation time and is used for all system auditing. Application can use different *Audit channel objects*.

Finally, CORBA supports optional *Non-repudiation services*, providing generation and later verification of evidence concerning performed actions and data associated with those actions. The evidence can be generated using either symmetric cryptographic algorithms requiring a trusted third party as the evidence generating authority, or asymmetric cryptographic algorithms assured by public key certificates issued by a certification authority. Keys or other information needed for generating or checking the evidence are available via Credentials.

CORBA differentiates two levels of conformance with the security specification. Level 1 is intended for security unaware applications and for applications with limited requirements to enforce their application security in terms of access control and auditing. Level 2, on the other hand, covers all the security functionality needed to allow an application to control the security provided at object invocation. In addition, it also includes application-specific security administration.

For legal reasons, security unaware applications are not acceptable in patient-related communication and co-operation [CE, 1995; CM, 1997].

Regarding the *Common Secure Interoperability* features, three levels have been specified [CORBA_SSS, 2001]:

1. *Identity based policies without delegation (CSI level 0)*: At this level, only the identity (no other attributes) of the initiating principal is transmitted from the client to the target, and this cannot be delegated to further objects. If further objects are called, the identity will be that of the intermediate object, not the initiator of the chain of object calls.
2. *Identity based policies with unrestricted delegation (CSI level 1)*: At this level, only the identity (no other attributes) of the initiating principal is transmitted from the client to the target. The identity can be delegated to other objects on further object invocations, and there are no restrictions on its delegation, so intermediate objects can impersonate the user.
3. *Identity & privilege based policies with controlled delegation (CSI level 2)*: At this level, attributes of initiating principals passed from client to target can include separate access and audit identities and a range of privileges such as roles and groups. Delegation of these attributes to other objects is possible, but is subject to restrictions, so the initiating principal can control their use. Optionally, composite delegation is supported,

so the attributes of more than one principal can be transmitted. Therefore, it provides interoperability for ORBs conforming to all CORBA Security functionality.

In the interoperability context, GIOP/IIOP protocol with the security (SECIOP) enhancements is especially important.

8.3 CORBA Security Services in the Healthcare Context

In the *shared care* context, in general, a client requests a service from a server. Client and/or server could be a user and/or an application. The guarantee of data security as well as the reliability and obligation of certain activities are basic conditions for health information systems supporting trustworthiness between physicians and patients, but also between different care providers. To design and to implement trustworthy information systems, the business objectives, the IT framework, and policies must be defined. Legislation, rules, roles, duties, rights, conditions, and penalties are defined by the security policy. Security threats and risks have to be analysed and assessed within the policies agreed. Countermeasures must be evaluated and implemented. These steps must be regularly repeated [Baur et al., 1996]. For the secure invocation of a service or the secure use of an application, two kinds of security are needed (see also Chapter 6.4):

- the communication security, ensuring integrity, reliability, and confidentiality of communication between authenticated partners, and
- the application security, controlling access rights to the application (functional and data access rights) as well as the reliability of the application functions and data

The access rights depend on the organisational structure of the healthcare institution (mandatory access rights), on the role of the principal within the care process (e.g. caring doctor, therapeutic team, consulting doctor, nurse, administrative clerk), and finally, on the patient's consent as shown in Chapter 5. The case of emergency care with roles of particular principals not being known in advance can be essentially covered using the CORBA *Identity domain*, a special case of *Security environment domains*.

To ensure integrity, reliability, accountability including non-repudiation and authentication as described in Chapter 5, strong authentication mechanisms must be used, relying on user-specific knowledge (password, PIN), ownership (electronic identity cards or other tokens with keys and certificates), or physical properties (such as fingerprint, voice analysis, retina analysis, face analysis). The strong authentication is a basic service for several related security services. It is joint with professional roles (e.g. as a Health Professional) verifiable through certified credentials. Therefore, the European HPC as preferred authentication token contains certificates for identity-role relationships. To avoid misinterpretations, the European term professional certificates should be preferred for credentials which are COSS objects with properties (methods) unacceptable for professional certificates as user or administrator manipulations etc. The *Professional Certificate Services* as well as the *Smartcard Services* framework will be amongst the next RFP issued by the CORBAmed TF. Confidentiality can be provided using symmetric and/or public key cryptographic algorithms. Nowadays, availability, feasibility and cost-benefit relation are promoting chipcards for security mechanisms in healthcare. Those cards will contain the user's identity, private keys for digital signatures (ensuring integrity and non-repudiation of origin and receipt), as well as, if necessary, class keys for group authentication. The latter function could also be provided using the individual authentication, together with directories of group members, their roles and rights. Finally, a trust authority (trusted third party = TTP) is needed, to ensure the correctness and validity of keys by certificates, and to provide directory services (public keys for encryption and proof of digital signatures), as well as notary functions. To fulfil trusty conditions, this TTP must be independent from the middleware

infrastructure, i.e. it must be provided externally to the system's architecture. For more details, see [Blobel and Pharow, 1997a; TRUSTHEALTH_WWW].

Functions related to the communication security can be globally organised, whereas the application security related to detailed access rights concerning a particular application can be controlled only locally, by the owner of the data or by the application administrator. In this context, the delegation mechanisms available in CORBA support the above described authentication procedures of security aware healthcare environment. The highly dynamic access rights underlying the access decisions are, in general, enforced by the application via access decision objects and additional services (like time services, account management). Using the various delegation options (simple delegation for end-to-end interactions, composite, combined and/or traced delegation) the middleware can adapt to requirements of different users and establishments. Figure 8.2 summarises the security objects defined in the CORBA Security Specification (after [OMG, 1997c]).

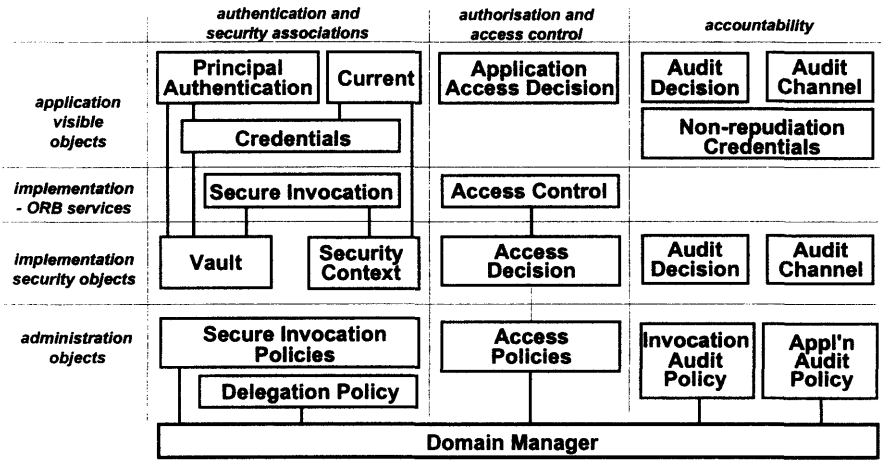


Figure 8.2: CORBA Security Objects – Architectural and Functional Relationships

8.3.1 CORBA Person Identification Service (formerly Patient Identification Service)

The *CORBA Person Identification Service* (PIDS) defines an object interface that organises person ID management functionality to meet healthcare needs. It is a generalisation of the CORBAmed Patient Identification Service from 1996. Since April 2001, the latest version 1.1 is available. After OMG [CORBA_PIDS, 2001], the PIDS is designed to

- support both the assignment of IDs within a particular ID Domain and the correlation of IDs among multiple ID domains,
- facilitate searching and matching of people in both attended-interactive and messagedriven-unattended modes, independent of matching algorithm,
- support federation of PIDS services in a topology-independent fashion,
- permit PIDS implementations to protect person confidentiality under the broadest variety of confidentiality policies and security mechanisms,

- enable plug-and-play PIDS interoperbility by means of a "core" set of profile elements, yet still support site-specific and implementation-specific extensions and customization of profile elements,
- define the appropriate meaningful compliance levels for several degrees of sophistication, ranging from small, query-only single ID Domains to large federated correlating ID Domains.

Figure 8.3 presents the PIDS conceptual schema.

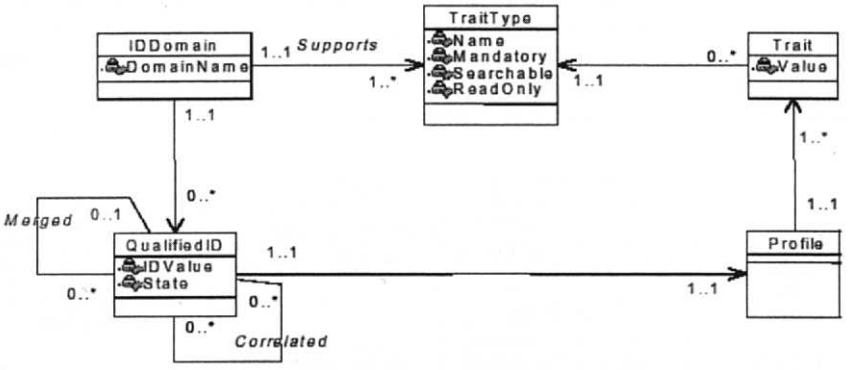


Figure 8.3: The CORBA PIDS Conceptual Schema [CORBA_PIDS, 2001]

Two main interfaces have been specified: IdentifyPerson interface concerning the matching of candidates and ProfileAccess interface for receiving a specific person's profile. The latter interface is specialised into IdentityAccess interface and SequentialAccess interface. For managing IDs in correlating domains, the two managers CorrelationManager interface and IDManager interface are needed.

8.3.2 CORBA Resource Access Decision Service

In April 2001, the *CORBA Recource Access Decision Service* (RADS) Version 1.0 has been published [CORBA_RADS, 2001]. The CORBA RADS manages authorisation decisions in the sense of requesting and receiving such decisions. It administers access decision policies. The facility is intended to be used by security-aware applications (see for reference Chapter 8.1). The authorisation logic is encapsulated within an authorisation facility that is external to the application. In order to perform an application-level access control, an application requests an authorisation decision from such a facility and enforces that decision as shown in Figure 8.4 for a healthcare environment. Because the specification has been developed for this environment, the next figures refer to healthcare resource access control (HRAC).

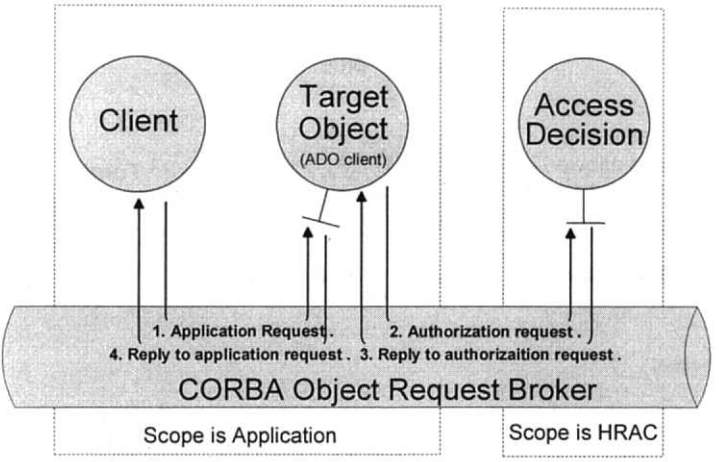


Figure 8.4: Interaction Sequence for an Access Request and Decision [CORBA_RADS, 2001]

This sequence is defined by CORBA RADS as follows [CORBA_RADS, 2001]:

1. An application client invokes an operation of the interface provided by the target object. The object request broker transfers this request to the target object and causes invocation of the appropriate method in the target object.
2. While processing the request, the target object requests authorisation decision(s) from the Access Decision object (ADO) by invoking the *access_allowed()* method of the ADO.
3. The Access Decision object consults other objects that are internal to the RAD (described in this specification) to make an access decision. The access decision is returned to the Target Object (ADO client) as a boolean.
4. The target object, after receiving an authorisation decision, is responsible for enforcing the decision. If access was granted by the ADO, the target object performs the requested operation and returns the results. If access to secured resources was denied, the target object may return partial results or raise an exception to the Client.

The RADS underlying access decision model is given in Figure 8.6. Using the general CORBA authorisation model (Figure 8.5) specified in the CORBA SSS [CORBA_SSS, 2001], it describes the interaction of the objects involved in that service as specified by OMG:

An Access Decision is requested by a client by invoking the *access_allowed()* method of the *AccessDecision* object (ADO) passing a *ResourceName*, Operation, and *SecAttributes*. The ADO consults a *DynamicAttributeService* to obtain an updated list of *SecAttributes* that include any dynamic attributes currently applicable for this access decision. The *DynamicAttributeService* may consult externally provided dynamic attribute evaluators as part of its implementation. The *AccessDecision* object also consults the *PolicyEvaluatorLocator* to obtain object references for the *PolicyEvaluator(s)* and the *DecisionCombinator* that are required for an access decision. The *AccessDecision* object consults the *DecisionCombinator* that consults with any *PolicyEvaluators* responsible for interpreting access policy that controls access to the *ResourceName/operation*. The *DecisionCombinator* encapsulates policy combination logic and is responsible for understanding the policy that controls how a series of results from *PolicyEvaluators* are combined including any precedence rules that may apply. It is the response from the *DecisionCombinator* that is returned to the client. This combinator is responsible for taking the results of the *PolicyEvaluators_evaluate*() method and making a final access decision.

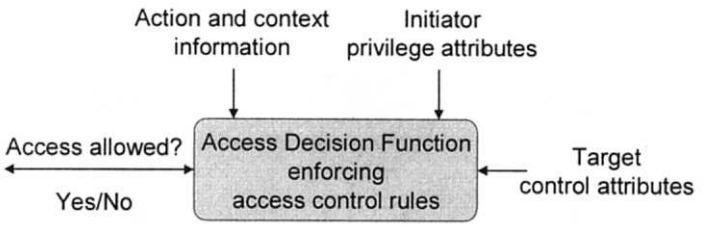


Figure 8.5: CORBA Authorisation Model, after [CORBA_SSS, 2001]

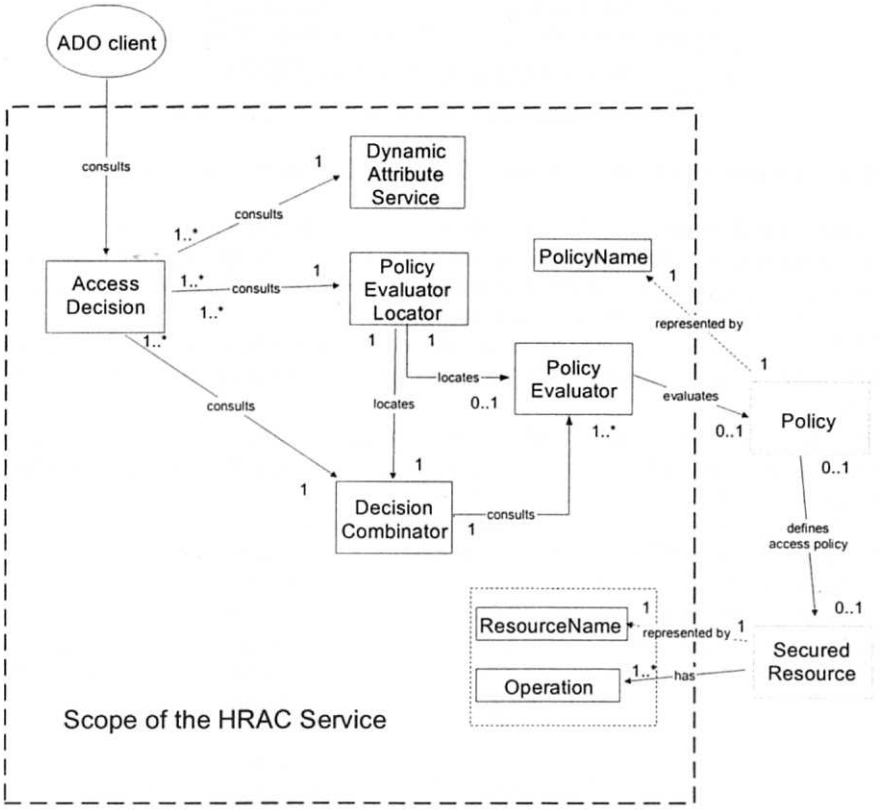


Figure 8.6: CORBA RADS Access Decision Model [CORBA_RADS, 2001]

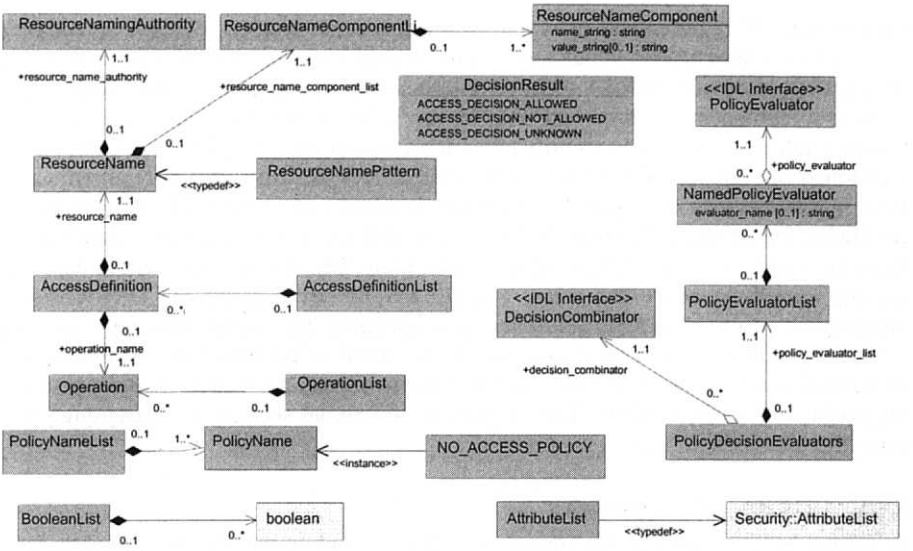


Figure 8.7: The CORBA RADS Information Model [CORBA_RADS, 2001]

This CORBA RADS has to be revised in the CORBA 3 framework as performed and currently enhanced within the HARP approach (see Chapter 12.7.2).

8.3.3 CORBA Terminology Query Service (formerly Lexicon Query Service)

In June 2000, the new edition of the *CORBA Lexicon Query Service* (LQS) V 1.0 has been published [CORBA_LQS, 2000]. Among others, the LQS deals with the following scenarios:

1. *Information Acquisition* using terminology services to aid in the process of entering coded data.
2. *Information Display* using terminology services to translate coded data elements into human or machine-readable external forms.
3. *Mediation* using terminology services to transform messages or data records from one form or representation into another.
4. *Indexing and Inference* using terminology services to inquire about associations which may or may not pertain between various data elements and to assist in the location of various data record sets, which may contain information relevant to the specific topic or entity.
5. *Browsing* using the terminology services to determine the structure and meaning of a terminology system.
6. *Composite Concept Manipulation* using the terminology services to aid in the entry, validation, translation, and simplification of composite concepts.

Regarding naming services for qualified names, registration services, etc., authorities are needed which are administratively similar to TTP services mentioned in the security context. Because terminology issues are out of scope of this book, the reader is kindly referred to the original documents.

8.3.4 Recommendations for Security Objects

Summarising all requirements of *shared care* information systems, we have agreed on a set of security services needed which are only mentioned in the current specification or have to be specified as future CORBA security objects at all. In that context, the secure time services as a fundamental notary's services for any other security services as digital signature and accountability as well as security token formats and message protection details are only mentioned but not specified in the detail needed. New security objects requirements arise from interoperability of different security mechanisms within one ORB or between different ORBs. Furthermore, the specification of external security objects and methods to access them are required. This is true for key generation, key distribution, certificates, cryptographic algorithms, and TTP services. Because middleware concepts as CORBA are widely independent of concrete application and environmental conditions, especially user-related services as authentication or credentials in the sense of professional certificates are not mentioned but of great importance. These security objects have been analysed and are being prepared for specification. The integration of external security services could be done using the interceptor object.

8.3.5 CORBA TTP Approach

In current security models, the service providers, including middleware services, are considered untrustworthy, following the basic concept to trust nobody and to organise security mainly by the communicating and co-operating partners in the sense of the distributed security paradigm [Blobel et al, 1997]. Especially for distributed middleware architectures involving a number of hosts, Varadharajan proposed to install, on each of them, security functions (e.g., encryption/decryption, signatures), a security information base, secure factory objects (objects responsible for creation and deletion of other objects), and secure interfaces [Varadharajan and Hardjono, 1996]. Most of these services can also be provided by functionalities specified in CORBA [OMG, 1997c].

Looking for further integration of open systems architecture, concepts like HL7 which can be mapped into the CORBA approach can make use of the security objects services immediately. Beside the current HL7 security solutions based on secure communication protocols wrapping HL7 messages as S/MIME (secure MIME) or sFTP (secure FTP) using the multipart content type, therefore the future HL7 security conception is at least partially directly consuming the specified CORBA services. Details describing a generic and open TTP for security enhanced EDI communication are given in Chapter 10. For more information see also the HL7 Web site http://www.hl7.org.

8.4 Summary and Conclusions

The CORBA middleware architecture, as it has been specified so far, as well as the proposed extensions provides advanced security services that allow the integration of both security unaware and security aware applications typical for the healthcare area. Special conditions defined in security policies of departments, institutions, organisations, regions, countries, or even the European Union can be specified, to control the middleware security services. The CORBA security solutions are suitable to integrate external security services in healthcare proposed within the TRUSTHEALTH project funded by the telematics programme of the EU. Moreover, the integration of such external security services is also possible in coexistence with other middleware approaches, such as DHE and HL7.

The breakthrough has been performed by the HARP Cross Security Platform, however. Facilitating different technical views of the RM-ODP, HCSP can be implemented in every environement even if it has been demonstrated for the Internet and a Java environment only.

9 Security Infrastructure Principles and Solutions

9.1 Introduction

As demonstrated in the chapters above, secure information systems need a framework of security services provided locally or remotely, decentralised or centralised, internally or externally. The classification depends on the domain considered. For example, a centralised directory service from the organisation point of view could be interpreted as decentralised from a federal institution point of view. The services are influenced by factors concerning legal, organisational, logical, and technical aspects. Most of the services securing information systems in sensitive domains as the health sector are based on cryptography applied. The detailed discussion of the cryptographic algorithms is out of scope of this book. For reference, see e.g. [Stallings, 1995].

Using such algorithms to provide trustworthy conditions and procedures for communication and co-operation in *shared care* information systems, a security infrastructure must be established the principals involved can trust. Especially in a heterogeneous environment including different HCE with often unknown partners or specific requirements on the legal basis for communication and co-operation (contractual relationships, liability, auditing, etc.), such a security infrastructure has to be trustworthy too. Therefore in most practical circumstances and certainly in a pan-European context, it is a requirement that the security services are provided by certain parties which are not formally attached to any of the communicating parties, but are in some sense trusted by these parties to fulfil all requested services in a secure and trustworthy way. Such independent security service providers are also called Trusted Third Parties (TTP). The TTP itself might be separated into different part belonging to different organisation and providing specific services needed. Using strong asymmetric cryptographic algorithms, e.g., for authentication and digital signature, the leading industrial companies in the world are nowadays able to provide a high security level meeting the requirements mentioned above. The integration and implementation of related technical and organisational means fulfilling also the new European legal initiatives' requirements support communication security and application security not only in the healthcare sector [CE, 1995; CE, 1999; CM, 1997; EC, 1998; EC, 1999].

9.2 Security Services Categorisation

Another scheme for classification of security services, a layered one, has been elaborated in the TrustHealth project [TRUSTHEALTH_WWW], separating basic security services, infrastructural security services, and value added security services. From the users' point of view, the basic services are provided locally and decentralised (Figure 9.1).

9.2.1 Basic Security Services

Basic security services concern fundamental security services and functions directly related to the secure communication between two parties. The services may also be applied in other circumstances such as for the authentication of the end user towards his or her workstation. The basic services compare to the security services described in the security framework of ISO OSI, and thus constitute a necessary basis for both the infrastructural and value added services.

9.2.2 Infrastructural Services

Infrastructural services facilitate secure, open communications in large scale, i.e., between a large number of users affiliated in various enterprises and belonging to various sectors even

in various countries. As mentioned already, in such environment it is rather unlikely or legally impossible that all users can know or trust each other. There even exist different security policies. Therefore an independent and, related to the principals, central TTP is needed to provide certainty and trustworthiness by organisation, methods, and services served. The handling of unique names, keys, certificates and cards is a typical example of services which is not necessary in a world where only a few parties known to each other communicate. However, as far as an infrastructure for large scale open communication is established, the infrastructural security services will become necessary; even a prerequisite to establish trustworthy health telematics in a large pan-European context. When leaving a local healthcare establishment, a TTP is needed to provide some of these services. Note that both basic and infrastructural services should be more or less transparent to the users; the users should not be involved more than absolutely necessary when using these services.

9.2.3 Value Added Security Services

Value added security services are related to the business functions of the user or the communication of documents and messages, fulfilling legal, organisational, methodological, or ethical requirements. Examples of such services, relevant to healthcare, are the registration of Health Professionals, the issuing of professional certificates, the secure storage of documents, anonymisation, pseudonymisation, and other.

If in the patient's care context a personal trustworthy relationship exists, which is supported by identified interaction between patient and Health Professionals, communications and Health Professionals' collaboration in epidemiological context do not need the identification of the patient. The gold standard is anonymisation or pseudonymisation of personal information. In both cases, the usually cryptographic algorithm used has to disable any re-identification of the person with reasonable effort. Be aware that not only demographic data such as name, address, or date of birth have identifying properties. If the patient's profession is student, this does not cause problems. If the patient's profession is being the Chancellor of the German Federal Republic, the person is identified, however.

Also specific rare diseases can have identifying character. Generally speaking, biological properties such as human's genetic information or the sum of medical data are unique and therefore identifying a person.

Anonymisation constitutes the deletion of any identifying information or its one-way encoding, which exclude any opportunity to re-identify the information's subject. Pseudonymisation constitutes an encoding, which enables a controlled re-identification for authorised users only. Therefore, pseudonymisation provides a compromise between anonymisation and person-relation. Furthermore, a pseudonym shall also hide time and order of its creation.

A simple way for pseudonymisation is the coding of identifying data into a continuous numbering schema. For hiding time and order of the pseudonym, a cryptographic algorithm is required. As discussed already in the EHR context, the long-term stability of the solution is challenging. More detailed considerations on the topic as well as practical solutions have been elaborated, e.g., by Pommerening [Pommerening_WWW].

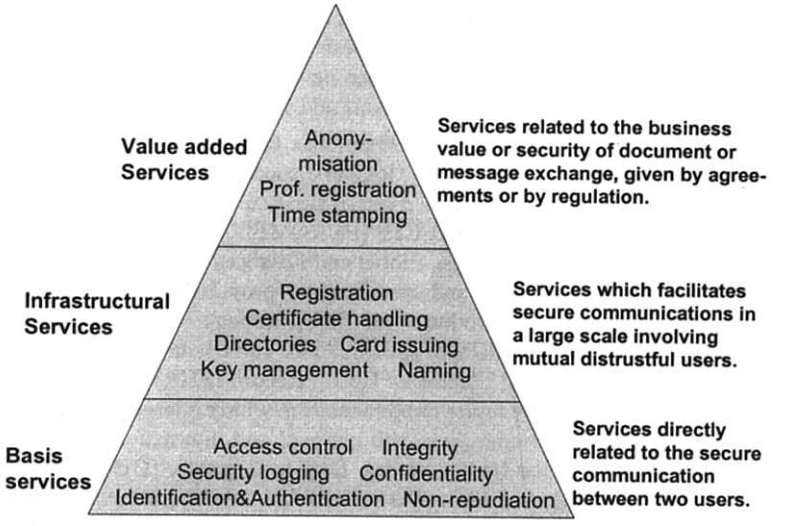


Figure 9.1: Security Services Categorisation [TrustHealth_WWW]

9.3 Basics of the Security Infrastructure

To overcome the weakness of existing solutions (e.g. the widely-used password for authentication), additional properties or even new tools are required. In Europe but increasingly also in other regions of the world, the combination of ownership and knowledge is used for strong authentication consisting of smartcards as token and the PIN identifying the card user as the card holder. So the ideal format for storing personal information items and secret keys is a processor smartcard with cryptographic functions. In the future, biometric procedures will be introduced such as fingerprint, voice analysis, retina analysis, face analysis, characteristics of typing or writing additionally to or instead of the PIN. The smartcard provides private keys of the PKI key pairs and corresponding symmetric and asymmetric cryptographic algorithms applied to them for identification and authentication. The identity certificate providing the trustworthy relationship between the public key and the card holder is a TTP services stored and managed in directories. Furthermore, the card is able to bear the cryptographic keys and mechanisms needed for other security services as e.g. integrity check by digitally signed hash values, and the protection of confidentiality by specific encipherment / decipherment algorithms. To technically enable the off-line use of such cards, corresponding (card verifiable) certificates can be stored on the card, functionality especially valuable for card-card interactions (see also Chapter 11). Relevant items including public keys have to be stored in and provided by certificates. In that context, information items about the physicians themselves (name, address, employers' or office's address respectively) are available. The smartcard and the card-related infrastructure are able to handle the access to public directories as well.

Additionally to identity certificates, attribute certificates describing professions and professional roles expand the identity card functionality (access card, token for strong authentication, integrity check, confidentiality, accountability using digital signatures) by the professional licence functionality. Such a card specially developed for Health Professionals' needs is called Health Professional Card (HPC), as mentioned already before in several chapters of this volume.

All these items belong to a system of security components within domains, and have thus to be considered for a domain policy. Aspects of these components and the secure communication and co-operation between them using open networks are also mentioned in detail in other chapters more focusing on issues of domains and policies.

The Magdeburg Medical Informatics Department has been involved in several security-related European projects and their logical successors as, e.g., TrustHealth [TRUSTHEALTH_WWW] dealing with the use of smartcards as well as the principles and the establishment of TTP, or EUROMED-ETS [EUROMED_WWW] dealing with Internet security and international TTP structures [Blobel and Katsikas, 1998; Katsikas et al., 1998]. Based on the experiences, definitions and specifications provided by these projects, in May 1997, the Magdeburg group has introduced the first European HPC and related TTP services in accordance with the TrustHealth specification. Joint with the long-term activities on a secure distributed oncological Electronic Health Record (EHR) [Blobel, 1996b,c], this security infrastructure is currently under implementation within a large scale pilot establishing a secure ONCONET. In co-operation with national and international initiatives in the area and close to standardisation bodies as, e.g. DIN in Germany [DIN_WWW], CEN in Europe [CENTC251_WWW], and ISO [ISOTC215WG4_WWW] as an international one, the pilot will support the improvement of the communication security as well as the application security in the context of a real medical application.

9.4 Health Professional Cards

The HPC can be implemented on several existing card operating systems. It is also designed to be implemented in several existing card configurations avoiding the need to create new masks. The card follows available and applicable ISO 7816 standards. The HPC supports asymmetric (RSA) as well as symmetric (DES) cryptographic algorithms. It has to keep at least three separate secret keys separating the signature key function from authentication and services, e.g. to manage needed availability of information locally. One key serves as class key (profession group) enabling secure communication or authorised access to the information in cases of impersonal referral or for emergency cases. Optionally, the card may contain a symmetric key used for interaction with a patient card and contain data for other applications, e.g. encryption keys or access privileges used for local computer protection or file encryption. To read the HPC, a T=1 Multifunctional Card Terminal (MCT) has been specified equipped with an ISO/IEC 9564 keypad and an LCD display. A more detailed description can be found in [The TRUSTHEALTH Consortium, 1997]. Figure 9.2 presents the scheme of the HPC infrastructure in the context of functional layers according to the TrustHealth-1 and the MCT specification. Integrating the smart card technology in the environment of Microsoft's Windows™ operating system (OS) and the standard PC, a consortium around Microsoft and hardware big players has specified an appropriate open PC/SC (personal computer / smart card) interface. Figure 9.3 shows the corresponding scheme of the HPC infrastructure in the context of functional layers according to the TrustHealth-2 and the PC/SC specification. Regarding the functional requirements on the smart card in the framework of the PC OS as well as Microsoft's or other provider's applications, the PC/SC specification defines interfaces to (and through) different layers up to the application layer enabling interoperation between the card (including the card infrastructure) and its services on the hand and system components and their services on the other hand.

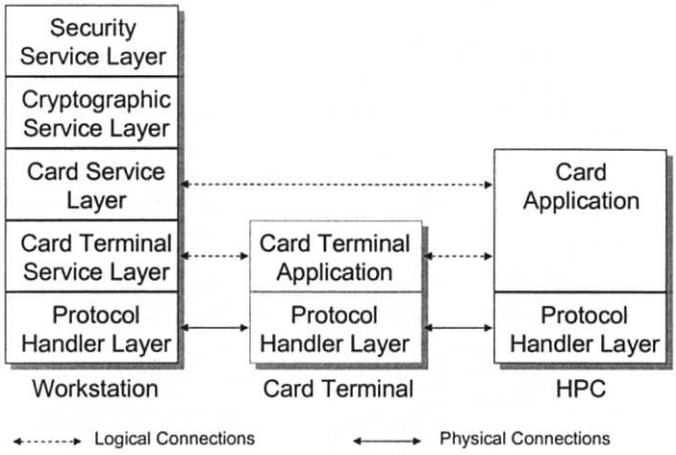


Figure 9.2: TH1.HPC (MCT-API) in the Context of the Functional Layers

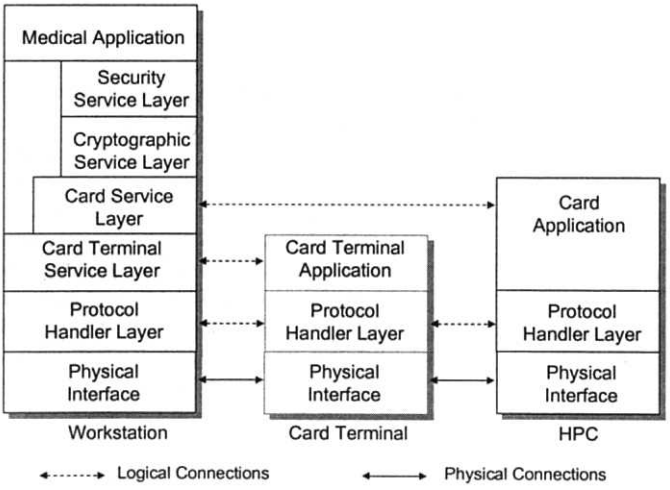


Figure 9.3: TH2.HPC (PC/SC-API) in the Context of the Functional Layers

The card is opened with a personal PIN code. Only with a correct PIN the host computer requests the user's authentication certificate from the directory service. It verifies the certificate, extracts the user's public key, and sends a challenge including a random number and preferably a time stamp or predefined pattern to the card. The card transforms the random number with the secret key and sends it back for verification. The host computer verifies the response with the public key stored in the certificate. The public keys of a user, together with some other information, rendered unforgeable by encipherment with the secret key of the certification authority which issued it. The authentication process may be repeated at regular intervals, or every time a critical operation is to be performed.

Figure 9.4 shows the file structure used. The file EF_{ICCSN} contains a unique ICC serial number presented as Tag-Length-Value data object. The file EF_{DIR} is only present if the card supports the indirect application selection method. The file EF_{CHN} contains the cardholder's name presented as Tag-Length-Value data object.

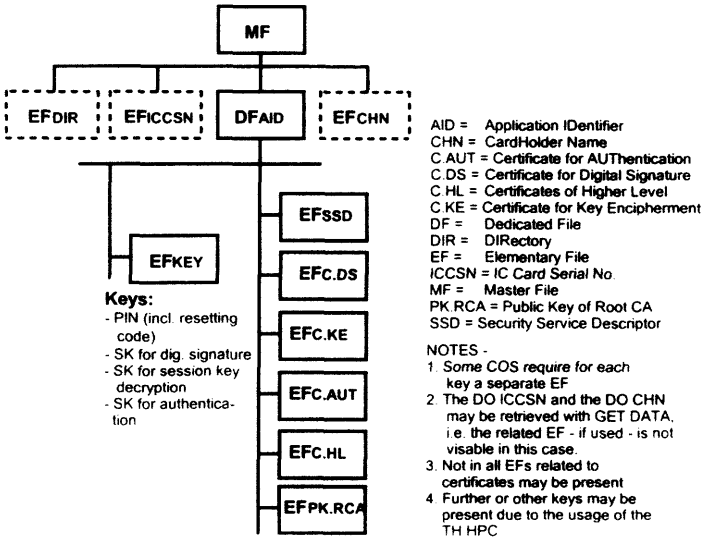


Figure 9.4: File Structure of a TH.HPC

As mentioned already in Chapter 6.7, in future the card will bear attribute certificates trustworthy characterising the card holder's profession and specific professional roles. Beside the profession certificate, the German HPC specification [HCP-Protocol, 1999] establishes two sets of attribute certificates. One set is dealing with physicians' training (e.g. education, approbation, qualification, profession, specialities, examinations), issued by the Physician Chambers of the German Federal States each physician is obliged to be registered in. The Physician Chamber is also responsible to register the card-stored personal items about the card holder. The second set is dealing with permission for GP to practice in specific regions for specific specialities, issued by the Federal States Physician's Statutory Bodies. Figure 9.5 demonstrates the corresponding card file structure schema.

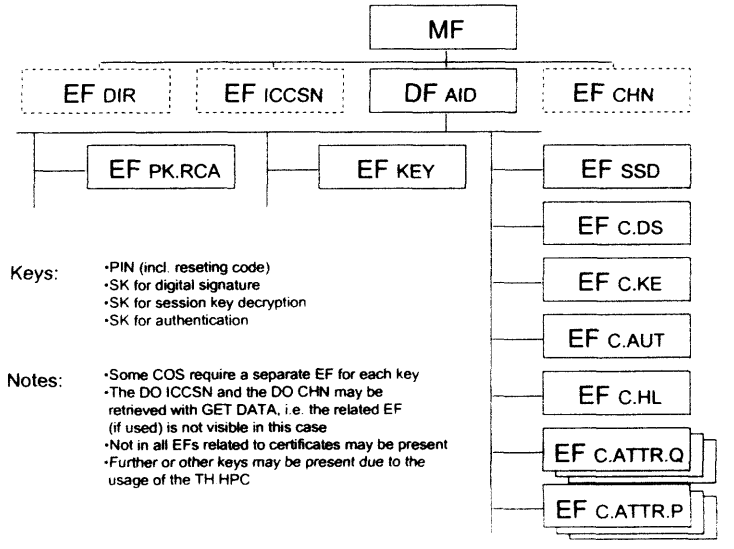


Figure 9.5: File Structure of a TH.HPC Containing Sets of Attribute Certificates

9.5 Security Toolkits

A special security toolkit called Security Development Environment for Open Systems (SECUDE™) provides certain security functions. Among others, SECUDE™ is able to handle X.509 public key certification functionalities, certification paths, cross certification, and certificate revocation. It provides utilities and library functions for the operation of certification authorities (CA) and interaction between a certifying CA and its certified users. Assumed that specifications (certificates, directory services, protocols, etc.) as well as applied tools of CAs involved comply with internatoinal standards, cross certification is not a technical matter but a concern of policies (policy practices statements, etc.) (see also Chapter 9.6.2 introducing into the corresponding ISO specification).

As an intelligent interface between the card infrastructure and the application, the SECUDE™ is a versatile security library and tool-kit developed by the GMD Darmstadt. It offers a broad range of application programming interfaces (APIs) for accessing well-known and established cryptographic algorithms, methods and techniques including symmetric and public key cryptography. So it provides implementation support and security services like origin authentication, data integrity, non-repudiation of origin and data confidentiality using symmetric (e.g. DES, Triple DES, IDEA) and asymmetric (e.g. RSA, DSA, DSS) cryptographic algorithms as well as various hash functions (e.g. MD2, MD4, MD5, SHA, Sqmodn), by this way realising a Personal Security Environment (PSE).

SECUDE™ has been adjusted by the GMD Darmstadt (now Fraunhofer Gesellschaft) for usage of the TH.HPC.

The high-level API **PKCS#7** implements the Public key Cryptography Standard No. 7 [RFC_2315]. All necessary keys used, security operations to perform (like encryption/decryption, digital signature generation/verification), handling of distinguished names and aliases, certificate management (featuring X.500/LDAP directory server support) as well as many other auxiliary functions (as ASN.1-encoding/decoding, for instance) are provided by the SECUDE™ standard API called **AF-API** (Authentication Framework and Certification). Access to keys and low-level usage of cryptographic algorithms (just as key generation, digital signature generation/verification, data encryption/decryption) is supplied by the SECUDE™ standard API named **SECURE-API**.

The lower level APIs of SECUDE™ are calling several functions of the Card-API and CardTerminal-API to get access to the smartcards and card terminals. Further details about SECUDE™ are discussed in the practical Magdeburg ONCONET context in Chapter 9.8.2.

9.6 Trusted Third Party Services

This section describes the overall functional aspects of Trusted Third Party (TTP) services required for trustworthy health telematics infrastructure. A detailed model consisting of functional roles and their interaction in a TTP infrastructure is described in [TRUSTHEALTH_WWW].

The TrustHealth-1 project[31] (TRUSTHEALTH_WWW) aims to facilitate the establishment of trustworthy information systems in healthcare, providing a set of specifications for security services and interfaces as well as a trusted third party service infrastructure with operational systems in some countries and publicly available specifications. The *shared care* requirements mentioned above must be fulfilled, accepting PC type workstations as dominating clients. The need to control key and certificate distribution makes the smart card

[31] The TrustHealth project is strongly co-ordinated with the German Model Trial "Health Professional Cards" (HPC) employing HPC for strong and certified authentication and TTP communication security services (Arbeitskreis, 1996). The demonstrated solution is part of this project.

format ideal. The framework, security services, TTP services, and interfaces described in the TrustHealth-1 project are thus based on the usage of a Health Professional Card (HPC) smart card, in the paper also called the TrustHealth Health Professional Card (TH.HPC).

The European TrustHealth-1 project has started to describe the processes within the real world and the electronic world in terms of security services and their service specification [Blobel and Pharow, 1997b]. TTP organisations have to provide different services as described in Table 9.1 as well as in the scheme of Figure 9.6. The services will be discussed in more detail in the next sections.

Table 9.1: Roles and Activities in the TTP Services' Context

Role	**Activities performed**
User	An individual or organisational entity
Public key registration authority (PK-RA)	An entity which uniquely identifies and registers users applying for the DS services provided
Professional registration authority (Pr-RA)	An entity which registers (and possibly authorises) individuals as Health Professionals
Naming authority (NA)	An entity which appoints unique certificate names to users. The naming authority may also handle the naming of Health Professional classes (e.g. physician), specialities (e.g., internal medicine) and possibly sub-specialities (e.g., nephrology)
Public key certification authority (PK-CA)	An entity which certifies the linkage between the unique certificate name and the users public signature or decryption key by issuing public key certificates digitally signed by the PK-CA. PK-CA is also responsible for the revocation and re-issuing of public key certificates
Professional certification authority (Pr-CA)	An entity which certifies the linkage between the unique certificate name and the users professional status by issuing professional certificates digitally signed by the Pr-CA. Pr-CA is also responsible for the revocation and re-issuing of professional certificates
Card issuing system (CIS)	An entity which issue signature/decryption chipcards containing (at least) the private keys of the users (card owners)
Local / central key generator (LKG/CKG)	An entity either located locally (by the user or PKRA) or centrally (by the PKCA or CIS) which generates the required public key pairs
Certificate directory (DIR)	An entity which provides the public key certificates, professional certificates, certificate revocation lists and possibly other information about users to other users at request

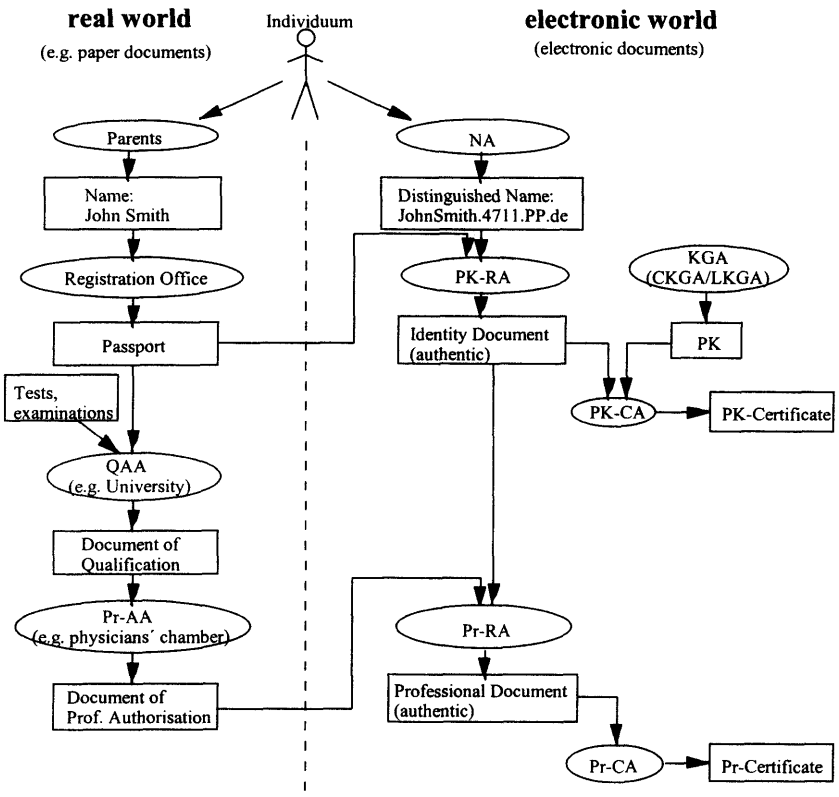


Figure 9.6: Real World and Electronic World Authorities

On the left hand side (the "real world", that means the world of papers we all know) one will find the authorities responsible for issuing authentic documents of an individual. That includes e.g. a registration office for inland and travel passports and a qualification authentication authority (QAA) for diploma etc. On the right hand side the authorities of the so-called "electronic world" (the world of bits and bytes) are mentioned. All the authorities of the electronic world are components of a Trusted Third Party structure. Any kind of information or certain data items are processed and transmitted from the real world to the electronic world by specific interfaces.

A TTP should not be formally connected with any of the communicating parties but independent, thus can be trusted to provide the infrastructural and value added security services in a secure and trustworthy manner. Responsible for a defined service, it comprises all of the independent organisations offering security services. To fulfil its basic objective offering security services with the necessary degree of (technical and business) functionality and assurance, the TTP has to provide a secure IT and communication system. Its formal or legal position within its service domain might be equally important.

Based on the formerly real world data items mentioned above, and connected to a unique distinguished name (DN) created by a Naming Authority (NA), a Registration Authority (RA) within the electronic world issues authentic documents (paper or database) of identity (Public key Registration Authority - PK-RA) of profession (Professional Registration Authority - Pr-RA). Besides that, specific key pairs (see above) are generated by a Key Generation Authority or Instance (KGA). This could be done as a centralised process within the

TTP (CKGA), or it could be done locally within the user's environment (LKGA). The decision whether it is allowed to generate keys outside a TTP environment is more a political than a technical one.

Authentic links between an individual's DN, his or her authentic ID documents and his or her Public key are used to issue a Public key Certificate (PK-Certificate) by a public key Certification Authority (PK-CA). The same is done by a Professional Certification Authority (Pr-CA) linking professional information items without any key to issue a Professional Certificate (Pr-Certificate). All these different data items, keys, and related certificates are necessary to establish the security services of identification and authentication, integrity, confidentiality, availability, and accountability. The public keys of the principals as well as their certificates must be published in a convenient way by Directory Services (DS). To join the directories with the PSE provided by the security toolkit, an interface must be used. In our environment the LDAP (Lightweight Directory Access Protocol, a shareware of the University of Michigan) is used. Finally, the HPC has to be issued. Figure 9.7 presents the TTP roles or services and a possible interaction model. The next chapters will describe the relevant services more detailed.

For legal reasons (responsibility) and for reasons of trust (professional bodies), different organisations become responsible for the different steps of the registration and certification processes.

9.6.1 General Description

To describe the structure of the relevant Trusted Third Party services one must again emphasise that a TTP comprises all the independent organisations which offers and is responsible for a defined TTP service. One girder of such an organisation should be a secure IT and communication system, which as a whole or in parts might be outsourced to another organisation. However, this is not the only or even the most important girder for a TTP to fulfil its basic objective: to offer security services with the necessary degree of (technical and business) functionality and assurance. Its formal or legal position within its service domain might be equally important.

Further, a TTP service structure is not meaningful unless we define a set of roles and describe the objectives and tasks of these roles are and how the various roles interact. Figure 9.7 pictures the relevant roles and how the various roles might interact in a general TTP infrastructure.

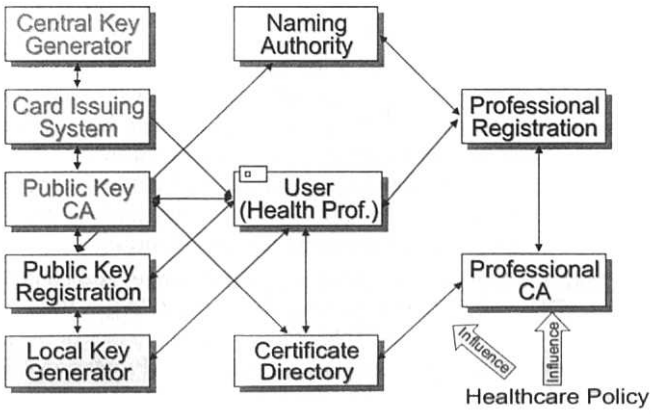


Figure 9.7: TTP Roles and Possible Interaction Model

There are common view and requirements to TTP infrastructure as key generation, card issuing, and public key certificate services, but also healthcare specific needs of partial functionalities and security requirements depending on the security policy in question. Nevertheless, looking for functional specifications of TTP services relevant for the healthcare sector, a few elements of relevant security policies in the European healthcare could be specified within the TrustHealth project. Examples of relevant issues would be the organisation and functioning of the public key and Pr-RAs and the interaction with directory services directly attaching to and influencing the acceptance of the users.

9.6.1.1 Naming

The *Naming authority (NA)* is an entity which appoints unique certificate names to users.

The naming authority may also handle the naming of Health Professional classes (e.g. physician), specialities (e.g., internal medicine) and possibly sub-specialities (e.g., nephrology).

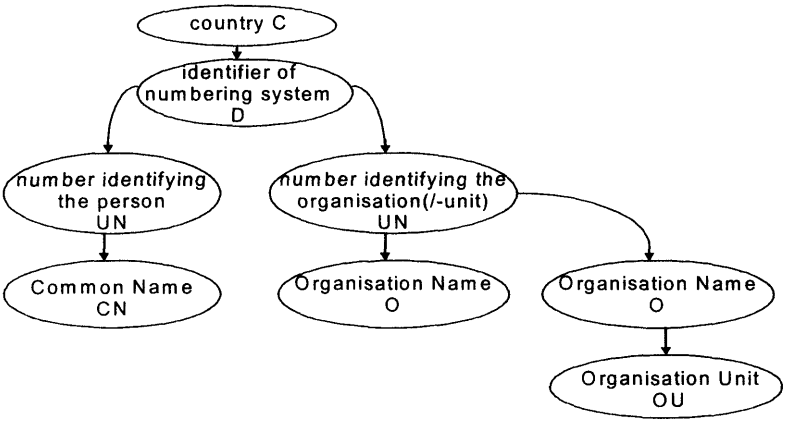


Figure 9.8: Naming Scheme

The close relations between naming schemes (Figure 9.8) of public key and professional certificates also gives specific requirements for the organisation and functioning of the naming authority. The concatenation of the attributes of each level results in the structure of a unique name:

case 1: CN||UN||D||C,

case 2: O||UN||D||C or OU||O||UN||D||C .

In both cases the uniqueness of the name is achieved by the attributes UN, D and C. Attribute CN, O and OU, respectively, are added for the readability for humans.

9.6.1.2 TTP Roles and Interactions

A *User* is an individual entity. A *Public key registration authority (PK-RA)* is an entity which uniquely identifies and registers users applying for the *Directory service* (DS) provided, whereas a *Professional registration authority (Pr-RA)* is an entity which registers (and possibly authorises) individuals as Health Professionals. The *Public key certification authority (PK-CA)* is an entity which certifies the linkage between the unique certificate name and the users public signature or decryption key by issuing public key certificates digitally signed by the PK-CA. PK-CA is also responsible for the revocation and re-issuing of public key certificates, whereas a *Professional certification authority (Pr-CA)* is an en-

tity which certifies the linkage between the unique certificate name and the users professional status by issuing professional certificates digitally signed by the Pr-CA. Pr-CA is also responsible for the revocation and re-issuing of professional certificates. And last but not least the *Card issuing system (CIS)* is an entity which issue signature/decryption chipcards containing (at least) the private keys of the users (card owners). The generation of keys could be done by a *Local / central key generator (LKG/CKG)* as an entity either located locally (by the user or PK-RA) or centrally (by the PK-CA or CIS) which generates the required key pairs. The certificates have to be stored in a *Certificate directory (DIR)*. It is an entity which provides the public key certificates, professional certificates, certificate revocation lists and possibly other information about users to other users at request.

9.6.1.3 Professional Part

It is desirable to have the professional certificate created by the same professional authority who delivers the actual professional information to be certified. Such professional or governmental organisations should represent the profession (including specialities, etc.) and have to have a complete overview on all members of that profession in the TTP catchment area. They should be competent to judge the status of each professional and have to dispose the unique registration number of each professional, if such a number exists. The TTP must have an appropriate legal status. The professionals should be obliged to keep this organisation up-to-date concerning their professional status.

It is important to distinguish the general registration or license to practice from the qualifications given by educational degrees. It is mainly the professional registration that can be of importance for Health Care information systems.

Most of the security services and mechanisms are related to the secure identification of the communicating and co-operating users. Therefore, the secure authentication is the basis for all other services. This authentication concerns the identity but also other important properties of the principals controlling those other services mentioned. Such properties could be the user's profession, qualification, special domains of interest, functional rights, etc.

Within the TrustHealth project funded by the European Commission, a security infrastructure including Health Professional Cards (HPC) and related Trusted Third Party (TTP) services has been specified and is currently under evaluation by large scale test sites in 6 European countries. The HPC serves as authentication token bearing the secret keys for authentication, digital signature and encryption to exchange a session key securely. Furthermore, it contains several certificates according to the X509v3 standard as the ID certificate (authentication certificate), the digital signature certificate, but also some sets of attribute certificates. One set deals with professions, qualifications, capabilities and skills. These certificates may be standardised internationally enabling transborder communication. Other ones is related to permission and legitimacy given to the card holder which are mostly restricted to a country or even to a region. In Germany, an extended specification of HPC including the certificates needed is now ready for use [HCP-Protocol_WWW].

9.6.1.4 Directory Part

The technical structure of the X.500 Directory Service is based on the modern "client-server principle". The user's "assistant" is the "directory user agent" (DUA) – an application software component in the workplace computer (client computer) – which facilitates user access to the Directory Service. The actual service itself is provided by the "directory system agent" (DSA) on a server computer. Since users must be shielded against changes in the network (e.g., a change of computer or site should be transparent to users), these two components must exist separately (Figure 9.9).

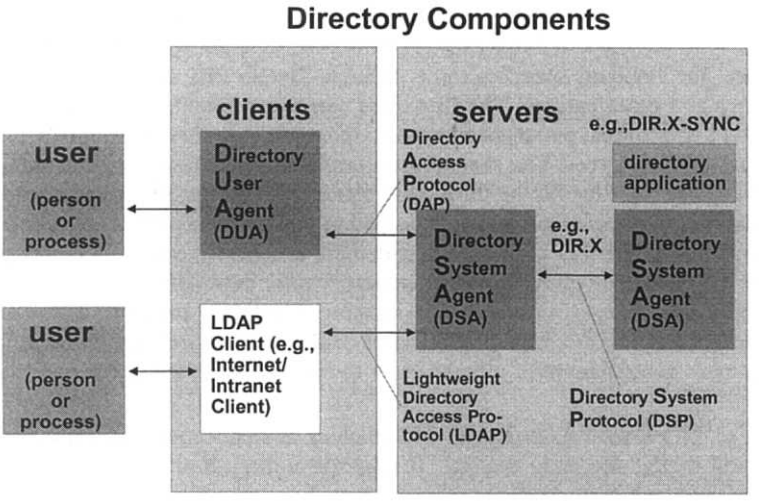


Figure 9.9: Directory Service Structure

9.6.2 The ISO Public Key Infrastructure Technical Specification

For harmonising and standardising health informatics issues internationally, ISO has founded its new ISO TC 215 "Health Informatics" consisting of the Working Groups WG1 "Health Records and Modelling Co-ordination", WG2 "Messaging and Communication", WG3 "Health Concepts Representation", WG4 "Security", and WG5 "Health Cards". The Working Groups have to provide New Work Item Proposals (NWIP), New Work Items (NWI), Technical Specifications (TS) and Standards ruling health informatics issues. Regarding the security and privacy domain, WG4 "Security" has deal with security, safety, and quality in health.

In order to establish TTP systems that are interoperable and able to cross-certify each other, relevant technical and policy-related standards are required. Among other standardsiation bodies, ISO TC 251 has recently provided a PKI standard framework. The resulting ISO Draft Technical Specification (DTS) 17090 "Health Informatics – Public Key Infrastructure" consists of three parts.

Widely approved meanswhile, DTS 17090 describes the scope as well as a glossary of terms used in all three parts, enabling the independent use of the documents by parties only interested in a certain content. Influenced by several European experts including this book's author, DTS 17090 reflects the European legislation and standardisation as well as the already more specialised German specification. Therefore, ISO DTS 17090 will be an important basis for any specification and implementation of health-related PKI in Germany and abroad. It will be also used as the PKI specification for the Bridge - CA in Germany.

ISO DTS 17090-1 "Health Informatics – Public Key Infrastructure – Part 1: Framework and overview" defines the basic concepts needed to describe a healthcare PKI and to provide a scheme of interoperability requirements to establish a PKI enabled secure communication of health information. ISO DTS 17090-1 introduces different types of certificates such as public key identity certificates, associated attribute certificates, self-signed CA certificates, and CA structures like CA hierarchies and bridging structures. The CA structures established depend on the legal, organisational and technical framework given. CA hierarchies normally require governmentally ruled national or even international single CA hier-

archy schemes (a challenge only hardly achievable at least at international scale), whilst bridging structures enables a framework on a lower level of agreements and administration.

Furthermore, the standard specifies three different Health PKI classes of actors: persons, organisations and other entities. The first class comprises Health Professionals, healthcare non-regulated employees, patients/consumers, sponsored healthcare providers and supporting organisation employees. The second class comprises healthcare organisations and supporting organisations. Finally, the third class comprises devices, regulated medical devices and applications. A set of suitable scenarios within the healthcare Public-Key Infrastructure has been mentioned, such as secure electronic mail, access requests from community based HP to hospital based patient information, access request between hospital information systems' components, billing scenarios, tele-imaging, electronic prescriptions. The standard takes into consideration the basic principles of data protection and data security, providing however a more general point of view in order to comply with the various national jurisdictions.

According to ISO DTS 17090, Health PKI enables authentication, integrity check, confidentiality and digital signature services. It supports authorisation as well as role-based access control. Following components of a PKI have been specified: the Certificate Policy (a named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements), the Certification Practice Statement (CPS) (a statement of the practices which a certification authority employs in issuing certificates), the Certification Authority (CA), the Registration Authority (RA) (should include Naming Authority – NA), the Attributes Authority and the Certificate Distribution and / or Revocation Systems (Directory services, Certificate Revocation Lists). References to other related standards and the Internet Engineering Task Force (IETF) specifications are provided in order to ensure openness and interoperability.

Different options are given for setting up a healthcare PKI across jurisdictions: Single PKI hierarchy, relying party management of trust, cross recognition, cross certification, and bridge CA which might be governed by different legal schemes.

ISO DTS 17090-2 "Health Informatics – Public Key Infrastructure – Part 2: Certificate profile" specifies healthcare specific profiles of digital certificates based on the international standard X.509 and the profile of this specified in IETF/RFC 2459 for different types of certificates. ISO DTS 17090-2 discusses certification authority certificates, cross/ bridge certificates and end entity certificates. It specifies attribute certificates for the different actor classes introduced in Part 1. ISO DTS 17090-2 defines the binding information needed to bind attribute certificates to key-bound identity certificates in a more comprehensive way than IETF does. All certificates are explained with detailed examples. Chapter 6.13.6.1 introduces in the ISO DTS 17090-2 certificates. The different healthcare certificate types defined in ISO DTS 17090 is given in Figure 9.10.

ISO DTS 17090-3 "Health Informatics – Public Key Infrastructure – Part 3: Policy Management of Certification Authority" deals with management issues in the context of implementing and operating a healthcare PKI. It defines a structure and minimum requirements for Certificate Policies and a structure for associated certification practice statements (CPS). The requirements for PKI policy management in a healthcare context are discussed in detail. These requirements concern, e.g., the reliable and secure binding of unique distinguished names (DN) as well as roles (and optional attributes) to the different actors introduced in Part 1. Furthermore, health PKI needs a high level of assurance, a high level of infrastructure availability, a high level of trust, Internet compatibility, and finally methods of evaluation and comparison of Certificate Policies are reflected. Certification practice statements have to meet the requirements established in IETF RFC 2527 "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework". In that

context, a CA with a single CPS should support multiple certificate policies on the one hand and a number of CAs with different CPS may support the same certificate policy on the other hand.

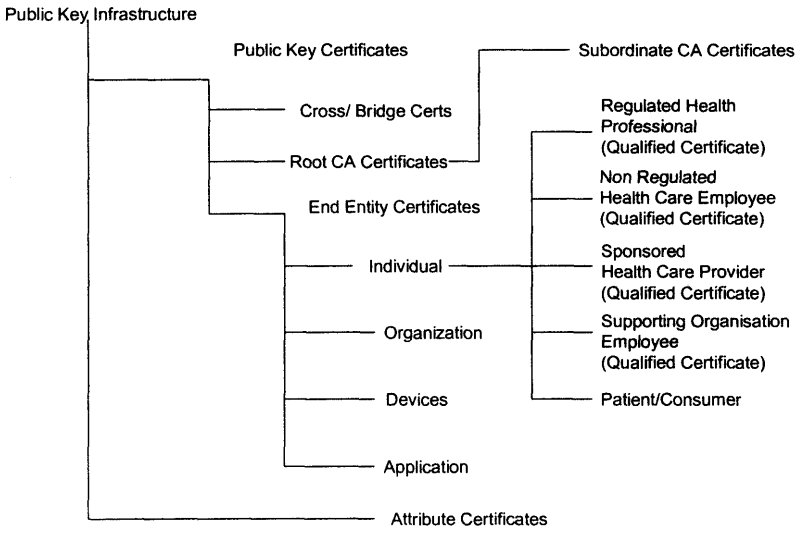


Figure 9.10: Healthcare Cerftificate Types according to ISO TC 17090 "Public Key Infrastructure" [ISO 17090]

CA obligations include the registration of potential certificate holders reflecting also the certificate holder's role, the authentication of the potential certificate holder's identity, procedures for distribution of certificates to certificate holder and to directories as well as the privacy guaranties. Beside procedures for managing keys and their revocation, also procedures for cross certifying with other CAs must be fixed. Templates for CPS are given for facilitating the practical use of the standard. Starting with statement types and statement description, the corresponding certificate policy requirements are mentioned referring to the related sections of the DTS.

9.6.3 Enhanced Trusted Third Party Services

As mentioned already in the HARP's HCSP context (Chapter 12.7), trustworthy interoperability requires more enhanced security services than trustworthy communication does. Because interoperability includes trustworthy co-operation based on information exchanged, application security services such as authorisation and access control for principals involved including their accountability, but also integrity, confidentiality, auditablity, and quality of information and processes as well as additional notary's services have to be established and managed.

Living in an e-World (g-Government, e-Health, e-Commerce, e-Business, etc.), the validity and reliability of information and processes, the quality and authenticated origin of offers and documents, but also liability concerns must be guaranteed.

Obviously, some of those services mentioned have to be provided by special parties able to guarantee the quality of services and products. Regarding the correctness and quality of guidelines concerning medical procedures, leading experts in the field, scientific organisations but also notaries or other accepted authorities may deal with this challenge.

Mechanisms used for such services are the same as introduced before: digital signatures and related certificates, but also watermarks sealing and protecting images. To protect the services, technical seals might be applied destroying equipment which is under unauthorised manipulation.

9.7 The German Security Infrastructure Framework

On September 24th, 1998, the global German root-CA, the so-called "Regulierungsbehörde für Post- und Telekommunikationsdienste (Reg-TP)" has been established. It was the first CA completely following the German "Information and Communication services Act (IuKDG)" and the embedded "Digital Signature Act (SigG)" [Der Deutsche Bundestag, 1997], later on changed to an electronic signature one. Besides the signature certificate, another one for time-stamping services and a third one for directory services has been issued. So the German Reg-TP has been allowed to offer a lot of services required for a trustworthy access and a secure communication based on HPC and TTP. The German SigG defined a hierarchical scheme for a CA structure. This meant that below the root-CA there is one (or more than one) level of CAs. And as usual, the root-CA was established to only certify other CAs. Thus, the Reg-TP has never issued any kind of user certificate.

With a short delay, the European Directive on Electronic Signatures [CE, 1999] came into effect with the obligation to the EU member states to revise their current, or to create, an electronic signature legislation compatible with the directive.

Within the new German legislation on communication services and the fundamental law on Digital Signature the framework of security infrastructure needed in the Information Society of multi-purposes and multi-modal communicating and co-operating systems for citizens has been specified. In that context, detailed requirements and recommendations have been mentioned in the implemention regulations of the Electronic Signature Decree (the German on electronic signatures legislation consists of two components: the Electronic Signature Law – SigG, and the Electronic Signature Decree – SigV) regarding authorities needed, protocols and forms used, and services defined for a common security infrastructure in e-commerce, healthcare, and any other domain. In the meantime, both SigG (in May 2001) and SigV (in November 2001) have already been adapted to the EU Directive.

The first step was the definition, implementation and accreditation of the German root CA, established in the governmental "Regulierungsbehörde für Telekommunikation und Post" observing the scene after privatising the former governmental post and telecommunication provider. This root CA is fulfilling the strong requirements for security including the physical security according the German data protection and data security legislation. Providing the basis for accreditation further CA but not users, the market for CAs is currently under development.

Supported by the legislation and the security infrastructure framework mentioned above, but also driven by the general development of a global market including telematics, communications, e-commerce etc., German and European providers are increasingly offering security infrastructure services and solutions. This includes facilities for centralised and decentralised key generation, key issuing services, directory services and related solutions like certificate revocation handling.

Further developments took place regarding national solutions for HPC. Restricted to medical doctors, in Germany is this HPC often called "Elektronischer Arztausweis", i.e. an electronic doctor's licence. That includes the definition and implementation of an extended HPC data set for physicians as well as for non-medical Health Professionals in accordance to the German Electronic Signature Law and other related legal bindings. The HPC data set includes attributes for handling professional roles and functions. It also concerns the speci-

fication and implementation of procedures for ordering, issuing and managing an HPC. Recently, the extension of the national HPC use first to pharmacists has been decided.

Germany's healthcare and welfare sector has to cover several laws, acts, and regulations that have to be taken into consideration if a regional health network including secure electronic storage, processing, and exchange of sensitive medical patient-related information is established. Among those are data protection laws, telecommunication laws, and laws ruling the electronic exchange of data including business procedures like contract signing.

Based on these laws, Germany has just passed the amendments of several national laws to transpose the relevant EC Directives. These admendments concern the Federal Data Protection Law (Bundesdatenschutzgesetz, BDSG), the Telecommunications Data Protection Decree (Telekommunikationsdatenschutzverordnung, TKDSV), the Electronic Signatures Act (Gesetz zur Regelung der Rahmenbedingungen zur elektronischen Signatur, SigG), and the Digital Signatures Decree (Signaturverordnung).

The ONCONET is completely following these national and European laws and directives. The storage of data is permitted on the basis of the patient's previous informed and written consent. Data security requirements are implemented with strong encryption mechanisms both on application and communication level. The operation of the PKI is aware of the aforementioned Digital Signature Act. Further amendments will be considered.

Patients will benefit from this new legal situation because of a higher privacy level of information. The Health Professionals will benefit because of a higher accountability, integrity, and reliability level of the medical information exchanged.

Actually, there are about 17 certified CA in Germany being allowed to issue user-related certificates by law. This number changes every month, however. The first certified CA – Telesec is officially on-line since January 1999.

9.8 The Security Infrastructure within the Magdeburg ONCONET Pilot

This section shortly describes the practical implementation of the security infrastructure in the framework of the Magdeburg regional Clinical Cancer Registry. Currently in co-operation with the HCE involved in the Cancer Centre Magdeburg/Saxony-Anhalt, the Magdeburg Medical Informatics Department establishes a secure ONCONET as a large-scale pilot within the TrustHealth-2 project [TRUSTHHEALTH_WWW].

9.8.1 The Regional Clinical Cancer Registry Magdeburg/Saxony-Anhalt

As an Electronic Health Care Record (EHCR) in oncology, the Clinical Cancer Register Magdeburg/Saxony-Anhalt, was implemented to improve quality and efficiency of cancer care including quality assurance, to facilitate research and education, and to improve communication and co-operation between the involved Health Professionals [Blobel, 1996b,c; Blobel 1997a]. For a catchment area of about 1.2 million inhabitants, currently a growing number of the 58 partner clinics and the Oncological Follow-up Organisation Centre[32] are online connected to an extended patient-centred and case-oriented tumour documentation covering all essential information from the cancer diagnosis up to the conclusion of the case. The Medical Informatics Department at the University of Magdeburg is hosting and maintaining the registry server. It also contributes to the further development of the ORACLE based registry application called „Giessener Tumordokumentationssystem" (GTDS) and developed by the University of Giessen, Germany, using ORACLE tools as forms, reports etc.

[32] The Oncological Follow-up Organisation Centre facilitates the cancer care of GPs and specialised oncological practices.

Due to the sensitivity of the information stored, processed, distributed, and co-operatively used as well as the additional threats and risks caused by the specific architecture, in the Clinical Cancer Register Magdeburg/Saxony-Anhalt as the first German healthcare information system secure external communications have been introduced in routine in 1993. The security measures provide system authentication and confidential communication, using FAST's MACS™ (Modem Access Control System) for analogue line, Kryptocom's Kryptoguard LAN L3™ boxes for ISDN lines between the secure server and secure external LAN (closed systems), and the ISDN Kryptoguard PC plug-in card for stand-alone (secure) PCs connected to the (secure) server by ISDN lines. The Magdeburg Clinical Cancer Register was the first German healthcare application with advanced security mechanisms ensuring strong authentication of users as well as integrity and confidentiality of data. Improvement and further development of the system is embedded in already mentioned projects, related to both security (e.g. ISHTAR, TrustHealth, EUROMED-ETS, MEDSEC, HARP, RESHEN) and architecture (e.g. HANSA) and funded by the European Union. The system was started as a centralised architecture with PC clients deploying terminal emulation. Currently, a step by step improvement of the application system via client-server architecture using remote procedure calls and remote SQL procedure calls up to independent interoperable systems based upon middleware concepts CORBA, DHE, and HL7 which have been discussed in detail in Chapter 3. The legal framework aspects of cancer registries have been described in [Blobel, 1996b; Blobel and van Eecke, 1999; RESHEN, 2002]. Table 9.2 illustrates the security-related phases of the Magdeburg Clinical Cancer Registry development.

Table 9.2: Highlights of the Clinical Cancer Registry Magdeburg/Saxony-Anhalt

1993	Foundation of the registry
1993	First implementation and routine use of secure connections to remote clinics via public analogue lines in the German healthcare
1995	Implementation and routine use of secure connections to the Oncological Aftercare Organisation's LAN via public ISDN lines, first implementation in the German healthcare
1997	First implementation of the TrustHealth HPC
1997	First implementation of an Internet-based TTP structure including the Universities of Athens, Calabria, and Magdeburg
1998	First implementation of open secure EDI
1999	Implementation of a secure health network in oncology based on the TrustHealth HPC and TTP architecture
2001	First implementation of the HARP Cross Security Platform in a healthcare environment and first demonstration of enhanced architecture and security in a clinical study context.
2002	Extention and enhancement of the ONCONET as well as its integration in a European Best Practice Project involving secure national health networks Greece, Finland, and Germany internationally.

9.8.2 Health Professional Cards Used

Within the TrustHealth-2 ONCONET demonstrator, the only smart card on the market that was evaluated as a secure signature creation device (SSCD) at that time according to German law and the European Electronic Signature Directive, to produce qualified signatures (E4-high), has been introduced: The Giesecke & Devrient Munich (GDM) smart card. The underlying smart card operating system, shortly introduced already in Chapter 9.5, is STARCOS® SPK (Smart Card Chip Operating System / Standard Version with Public Key extension) version 2.3. The current STARCOS® SPK 2.3 smart card offers several functions needed to establish and run a security infrastructure. STARCOS® SPK 2.3 constitutes a

complete operating system for smart cards, based on the former version STARCOS® S 2.1. The main features of STARCOS® SPK 2.3 include the support for several applications in the card, which may be installed independently of each other (multi-functionality), the implementation of several hierarchical file structures (file organisation), multi-level security mechanisms during communication (secure messaging), the implementation of various access controls (authentication), the generation and verification of digital signatures, asymmetric authentication, and key generation RSA-CRT up to 1,024 bits. The number of loadable applications is only limited by the amount of EEP-ROM memory available. The registration, creation and loading of data for an application can be done independently with defined security levels. The application designer is responsible for the definition of the security level and structure of its own application.

As introduced in Chapter 9.4, elementary files (EF) establish the actual data storage. STARCOS® SPK 2.3 distinguishes between the common elementary files (EF) for applications, special internal secret files (ISF) and internal public files (IPF) used by the operating system. Internal public file (IPF) contains public keys for cryptographic methods to be used just within the card. They must be created explicitly and require a file identifier. Keys may be entered, modified, or overwritten only one per Master file (MF) and Directory file (DF). The card itself has a prescribed transparent data and file structure. The keys in public key crypto systems are divided into public and private (secret) parts. The public keys must be entered in the IPF and can be published in a certificate directory, whereas secret keys will be stored in the ISF of the card. A key format list (KFL) provides access to individual key components, providing information about all key parts installed and their respective lengths.

The public directory service used along with the well-known LDAP connection interface offers easy access to the public key certificates of the users and thus to the related public keys as well as to many further information items called directory attributes. These attributes do not represent attribute certificates as such but are used to uniquely identify a person an encrypted file shall be sent to. Among this information items mentioned are postal and email address, phone and fax numbers, etc.

The SECUDE™ run-time service library implemented closely to the medical application provides the link to the higher level functions and thus offers algorithms for the mechanisms for both communication security and application security services. This includes well known and established symmetric and public-key cryptography like several hash algorithms as, e.g. MD5, SHA-1, RipeMD-160, etc., as well as algorithms for asymmetric (RSA, DSA) and symmetric cryptography (DES, 3DES, IDEA). The library further offers security functions, security APIs and a number of utilities with the following functionality:

- security functions for origin authentication, data integrity, non-repudiation of origin and data confidentiality purposes on the basis of digital signatures and symmetric as well as asymmetric encryption;
- X.509v3 key certification functions, handling of certification paths, cross-certification, and X.509 v2 certificate revocation lists (CRL);
- utilities and functions for the operation of certification authorities (CA) and interaction between certifying CAs and certified users;
- utilities to sign, verify, wrap, unwrap and hash files;
- Internet PEM processing according to RFC 1421 – 1424;
- processing of RFC 1422-defined certificate revocation lists;
- Generic Security Services - API Version 2 (GSS-API) RFC 1508 and 1509;
- Public Key Cryptographic Standards (PKCS #1, #3, #7, #9, #10, #11);

- all necessary ASN.1 encoding and decoding.

All security relevant information of a user (like secret keys, verification keys, certificates etc.) are integrity-protected and confidentiality-protected stored in a so called Personal Security Environment (PSE). SECUDE™ provides two different PSE realisations which are both only accessible through the usage of a Personal Identification Number (PIN):

- a DES-encrypted directory (software PSE);
- a smart card environment (smartcard PSE).

The behaviour of some API's and commands in SECUDE™ can be controlled by environment variables. SECUDE™ can be configured to use a directory service to access certificates, certificate pairs and certificate revocation lists (CRL) stored in a directory using a DAP or an LDAP interface. Most of SECUDE™'s functionality is available through a command line utility called "secude". With this utility it is possible to maintain a PSE or wrap and unwrap data. It has a rudimentary user interface and can also be used for batch file processing.

SECUDE™ has originally been adjusted by the GMD Darmstadt (now SIT within Fraunhofer Germany) for the usage of Health Professional Cards (HPC) and is currently under development by SECUDE GmbH Darmstadt.

9.8.3 Architecture and Services of the Pilot TTP

In the context of the different requirements of the current German legislation, related rules and regulations, the different TTP functions are provided by different partners inside and outside of the TrustHealth-2 pilot project of a secure ONCONET. The policy of the TTP includes the procedures of card request, naming, individual and professional registration, individual and professional certification, card issuing, directory services including revocation procedures, and card distribution [Blobel and Pharow, 1997b; Blobel and Pharow, 1999; Pharow and Blobel, 1999]. Because the security infrastructure market is permanently changing and evolving, the implementation of the several needed services will be managed highly dynamically. Figure 9.11 shows the current layout of the TTP structure and infrastructure including the different roles to be played by the Magdeburg Medical Informatics Department and its partners as the Cancer Centre Magdeburg/Saxony-Anhalt, the Physician Chamber Saxony-Anhalt, the Darmstadt Society for Mathematics and Data Processing (Gesellschaft für Mathematik und Datenverarbeitung – GMD, meanwhile merged with the Fraunhofer Society), the Munich Giesecke&Devrient company and others. A preferred solution was the common management of registration and professional registration by the Physician Chamber.

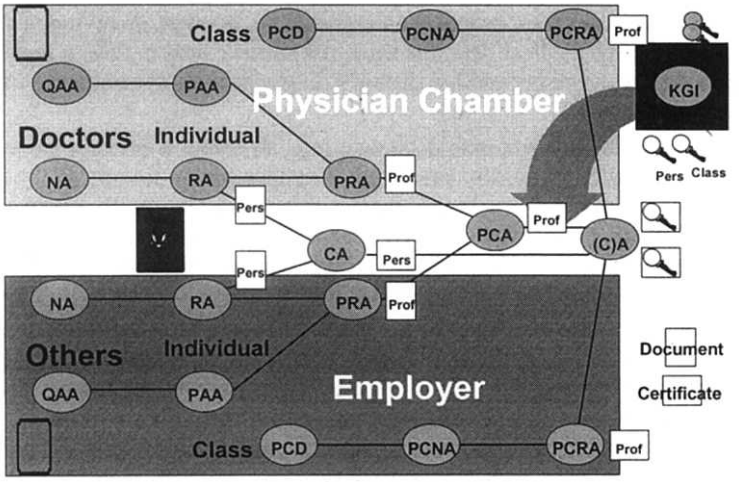


Figure 9.11: The Magdeburg TTP structure

Using the upcoming German security infrastructure, the certification authorities will be providers officially accredited by the German root CA. Annex C gives a detailed description of the TrustHealth-2 scenario of a secure regional network on oncology based on the European security infrastructure deploying HPCs and TTP services.

Abbr.	Name	Function	Partner
CA	Certification Authority	creation of certificates, directory service including revocation service	TC Hamburg
KGI	Key Generation Instance	instance responsible for generation of keys: SSCD (secure signature creation device)	TC Hamburg
NA	Naming Authority	authority to distribution of unique net names	PCSA
PAA	Professional Authentication Authority	authenticates the professional relevance of qualifications, specialities, etc.	PCSA / TRM
PCA	Professional Certification Authority	certification of professional information of the users (attribute certificates)	TC Hamburg
PRA	Professional Registration Authority	registration of profession-related information of users	PCSA
QAA	Qualification Authentication Authority	authenticates available qualifications for Health Professionals	PCSA
RA	Registration Authority	registration of users and user identities	PCSA
TS	Timestamp Provider	Provision of secure and trustworthy time-stamps for documents and workflow	TC Hamburg

Figure 9.12: ONCONET responsible TTP structure functions and partners (as implemented)

9.8.3.1 The Time Stamp Service

Replacing the paper world's business procedures by electronic means, also the paper world's procedures of time stamping such as post stamps or data marks on newspapers and letters must be replaced by electronic ways of proof of the actuality of a piece of information. Here too, a reliable proof of time is required in order to establish security concerning

the validity of electronic transmissions. In particular in the case of contract-relevant or content-relevant messages and files, orders and purchases via Internet, archiving of digital data, or exchange of sensitive medical data: the time of a specific action plays a major role. Just imagine that a doctor questions another doctor's diagnosis and especially the time when it took place.

A time signature "seals" an electronic document (e.g. an email, a doctor's letter, an image, or even a web page) with the legally valid time. Any subsequent forgery of the time signature is impossible. It proves that a digital patient information or health record was created or updated, and that it has not been changed since that time. For example: One can now prove timely the receipt of an image sent for second opinion by e-mail to another clinic. Or one can check whether or not any data retrieved from a medical data base is still valid.

Time signatures are used in several business fields including, e.g., archiving of any kind of electronic documents in healthcare and welfare, electronic order processing (e.g. for pharmacies), settlement of time-critical treatment statements, acknowledgement of any kind of receipt, medical workflow management, electronic registration of current operating time, or – last but not least – electronic signatures in general. This calls for possibilities to communicate and interact with each other via network systems, which particularly applies to contract-relevant and time critical transactions that need to be proven. The ability to furnish a proof of transactions in health and beyond has to be secured also in the long term. An awareness of the necessity to establish certainty about the time of electronic business processes and workflow needs to be created. Time signatures meet these requirements.

9.8.3.2 The timeproof® Time Stamp Solution

The timeproof® company [TIMEPROOF_WWW], a Hamburg-based German vendor, develops and markets time signature systems (hardware and software) for certification authorities, especially those conforming with the German Electronic Signature Law (SigG) [Der Deutsche Bundestag, 2001a] and the related Electronic Signature Act (SigV) [Der Deutsche Bundestag, 2001b] but also for service providers to offer secure time stamping services, and to companies interested in authenticating their own electronic transactions. Besides the provision of time stamping components, timeproof® also assists in integrating secure time stamping into business applications. Important timeproof® partners are the German CA TC TrustCenter Hamburg, the international iD2 group (e.g. Sweden), the German SecuNet security company and others.

The timeproof® time signature system comprises hardware and software components based on the German Electronic Signature Act in its current version. The so-called trust box stamps all digital data with the legally valid time, and then gives them a system certificate. Even if the time signal fails or is manipulated, the time signature function of the device continues with the same accuracy. The server software establishes the connection between the Internet and the trust box, logs all time signature events, and thus guarantees the trustworthiness of the signature for the duration (life time) of the certificate. The time signature system is hardware-protected against unauthorised access.

9.8.3.3 Timeproof® Trust Box

In order to meet different requirements by different client categories, timeproof® has developed three levels of devices with nearly identical technical parameters. All devices are compatible to the current RFC / IETF standards and to the work that is performed by the European ETSI institute.

The highest level device is called TSS 400 and has been developed for CAs which conform to the German Electronic Signature legislation. It is E2-high evaluated, certified, and confirmed. By means of TSS 400, CAs can provide their customers with time signatures that

conform to the German legislation. This way they can attest that a document existed at a fixed point of time and has not been changed since then. A time signature is also recommended when a certificate is first issued and every time it is altered. The second device called TSS 380 has been developed for commercial services. This system enables service providers to offer their customers a secure time stamping service providing security for all their Internet transactions.

Finally, the smallest of the devices, TSS 80, produces time signatures for a company-wide usage. It safeguards the time of all internal business transactions and enables companies with liability risks or requiring traceability to provide all internal documents and processes like workflow, archiving, or emails with a time signature. Not only can the integrity of a business activity be safeguarded this way; it can also be proved later on when the transmission or activity took place.

9.8.3.4 The ONCONET Time Stamp Implementation

Using the SFTP approach (Chapter 10.3.4), communication and co-operation in the ONCONET is provided in a trustworthy manner using the system time for time stamping. Within the RESHEN project aiming at delivering best practice demonstrators and guidelines for regional and international secure health networks, these time stamps are replaced by solutions certified according to the German Electronic Signature legislation.

For the German RESHEN pilot implementation, the timeproof® online service will be used because of the restricted number of time signature requests during the pilot time. After having made our own experience concerning this service, a TSS 80 could be bought and implemented within the ONCONET environment. Figure 9.13 describes technical parameter of timeproof® time stamp boxes.

Properties	Technical Solution
System characteristics	• Receives time signal • Processes internal and external time information • (Takes account of summer time and leap seconds) • Manipulation check • Time signature
Dimensions (with/without external casing)	(Height x width x depth) approx.192 x 548 x 385 mm/standard 19 "3HE, 84TE,approx.380 mm incl. ports and smart cards
Weight (with/without external casing)	12 kg/4 kg
Time signal	Germany DCF77 International GPS (optional for TSS 80 and TSS 380)
Time output format	Year, month, day, hour, minute, second, time zone
Time precision	Usually 100 ms and max.500 ms
Operational temperature range	10 °C – 40 °C
Time signature components	Smart cards
Time signatures/hrs.	8,000 TSS 80 and 12,000 TSS 380 and TSS 400
Signature exchange formats	PKCS#7, PKIX time stamp
Identification, authentication and access control for server	Password (4-eye-principle)
Interface	RS232
Voltage supply	Rated voltage 230 V, 50 Hz (voltage range 100 V to 230 V)
Rated current	300 mA
DCF input sensitivity	Minimum 2 µV; maximum 80 mV at 50 Ohm
GPS input sensitivity	Minimum 1.5 µV; maximum 10 mV at 50 Ohm
CE-seal	

Figure 9.13: timeproof® Time Signature Creation Device Parameters

All clients do only need the timeproof® API (a DLL type) in order to contact the server and to perform operations RequestTimeStamp, ResponseTimeStamp, and OperationTime. The server system itself and the software to create time signatures should be implemented on top of, needs to fulfil the following simple requirements:

- Pentium III 500 MHz, 256 MB RAM, 2 serial interfaces or comparable system, hardware handshaking recommended for serial interfaces;
- Operating system with authorisation function, e.g.UNIX (on request), Windows NT, Windows 2000, Linux or similar;
- Java Virtual Machine Version 1.2.2-001 or higher
- Java Communication Extension Version 2.0 or higher
- Active DCF antenna, active GPS antenna (for GPS option)

9.8.3.5 ONCONET patient consent for epidemiologic registry

Covering the UHM and ONCONET patient consent document form for the transmission of medical and administrative patient-related data to an epidemiologic registry, the German version is the legally relevant one. Nevertheless, a translation has been provided to facilitate the reading of the form by readers unable in reading German.

Name: _____ Vorname: _____

Anschrift: _____

_____

I. PATIENTENINFORMATION

Das Tumorzentrum Magdeburg/Sachsen-Anhalt hat zur Verbesserung der ärztlichen Betreuung von Patienten mit Tumorerkrankungen am Universitätsklinikum Magdeburg ein *„Klinisches Tumorregister"* eingerichtet, das den so wichtigen schnellen Zugriff auf notwendige Daten für den Sie jeweils behandelnden Arzt gewährleisten soll.

Des weiteren ist beabsichtigt, mit der Gesamtheit der im *„Klinischen Tumorregister"* enthaltenen Daten aller erkrankten Personen, wissenschaftliche Untersuchungen von Geschwulsterkrankungen zu betreiben. Die Untersuchungen werden ausschließlich in anonymisierter Form durchgeführt. Dazu werden nur medizinische Daten so zur Verfügung gestellt, dass ein Bezug zu Ihrer Person nicht möglich ist.

II. EINWILLIGUNGSERKLÄRUNG

Für diese Aufgaben bitten wir Sie, Ihre Einwilligung zur Datenübermittlung an das *„Klinische Tumorregister"* zu erteilen. Wenn Sie einwilligen, werden

-Ihre personenbezogenen Daten (Name, Vorname, Geburtsdatum, Geschlecht, Anschrift) und

-alle Sie betreffenden ärztlichen Informationen (wie z.B.: Krankheitsverlauf, Diagnosen, Befunde)

an das *„Klinische Tumorregister"* übermittelt, wo sie gespeichert werden. Entsprechend den gesetzlichen Vorgaben werden die Daten durch geeignete organisatorische und technische Maßnahmen gegen unberechtigten Zugriff gesichert.

Ich willige ein, dass meine Daten zu den beiden unter I. angegebenen Zwecken übermittelt werden. Mir ist bekannt, dass ich meine Einwilligung jederzeit widerrufen kann. Durch eine eventuelle Verweigerung meiner Einwilligung entstehen mir keine Nachteile.

Ort, Datum: _____ _____
Unterschrift

III. MELDUNG NACH DEM KREBSREGISTERGESETZ:

Nach der Änderung des Gesundheitsdienstgesetzes vom 14. Juni 2000 (GVBl. LSA Nr. 21/2000, ausgegeben am 20.6.2000) sind Ärzte und Zahnärzte verpflichtet, bestimmte Daten zu Patienten im Kontext von Krebserkrankungen an das *„Gemeinsame Krebsregister der Länder Berlin, Brandenburg, Mecklenburg-Vorpommern, Sachsen-Anhalt und der Freistaaten Sachsen und Thüringen"* zu übermitteln. Diese Übermittlung wird durch das *„Klinische Tumorregister"* übernommen. Der Gesetzgeber hat Ihnen aber hierfür ein **Widerspruchsrecht** gegen diese Übermittlung eingeräumt, das durch die vorgenannte Einwilligungserklärung nicht berührt wird.

Wenn Sie **FRAGEN** haben, sprechen Sie diese bitte sofort an. Der Sie jeweils behandelnde Arzt gibt Ihnen weitere Erläuterungen.

Name: _____ Surname: _____

Address: _____

_____

I. PATIENT INFORMATION

In order to improve the medical treatment for patients suffering from cancer diseases, the Cancer Centre Magdeburg / Sachsen-Anhalt has established a *Regional Clinical Cancer Registry* at the Magdeburg University Hospital. For the doctors treating You momentarily, this registry will guarantee a fast and reliable access to all medical data required which is a very important aspect.

Furthermore it is intended and scheduled to perform scientific research work regarding tumour-related diseases using the full range of medical and administrative patient-related data stored in the *Regional Clinical Cancer Registry*. All investigations without any exception will be performed anonymously. Medical data which are the only data used are presented in a way that there is neither a chance to re-identify Your person nor to derive any linkage.

II. *CONSENT DECLARATION*

For the performance of these tasks and objectives, we kindly ask You to express your commitment that these data are allowed to be transmitted to the *Regional Clinical Cancer Registry* to be stored there. If you do so, the following data items are considered to be transmitted:

-Your specific person-related data (name, surname, date of birth, gender, address), and
-all Your related medical data (e.g. course of disease, diagnosises, findings, etc.)

Conforming to existing legal obligations and limitations, the data stored are protected against unauthorised access by appropriate organisational and technical measures.

I hereby agree, that my data are allowed to be transmitted for both purposes I and II mentioned above. Furthermore I have been informed that I am allowed to cancel this consent at any time. Through a possible cancellation of my consent, no disadvantages occur for me.

Place, Date: _____ _____
Signature

III. CANCER REGISTRY LEGISLATION OBLIGATION OF INFORMATION

According to an update of the German Health Service Law (Gesundheitsdienstgesetz) of June 14th, 2000. published in German in GVBl. LSA Nr. 21/2000 on June 20th, 2000, physicians and dentists are obliged to transmit certain patient-related data recorded in the context of cancer-related diseases to the epidemiological registry called „*Gemeinsame Krebsregister der Länder Berlin, Brandenburg, Mecklenburg-Vorpommern, Sachsen-Anhalt und der Freistaaten Sachsen und Thüringen*". This transmission will be performed by the *Regional Clinical Cancer Registry*. The German legislator has granted you a **right to object against this transmission** however for this, which is not touched in any way through the aforementioned consent.

Please raise any **QUESTIONS** concerning the content of this consent document immediately. Your treating doctor will guide you with further explanations.

9.9 Summary and Conclusions

Modern distributed interoperable health concepts such as shared care, managed care, or disease management can only be introduced in combination with appropriate measures for security, privacy, quality, and safety. Using advanced methodologies like cryptographic algorithms, such measures require an enhanced infrastructure to provide all communication and application security services as well as quality assurance and safety services needed. In that context, basic services, infrastructural services, and value added services must be considered.

Starting in Europe, security tokens such as smart cards and appropriate Trusted Third Party (TTP) services are currently introduced for Health Professionals, but increasingly also for patients.

Advanced specifications for Health Professional Cards, security tool-kits, and TTP are discussed, referring, e.g., to the German security infrastructure framework. The need for legal, organisational, and technological solutions is emphasised. In that context, international specifications for TTP services such as ISO DTS "Health Informatics – Public Key Infrastructure" have been introduced.

Finally, requirements and solutions have been demonstrated based on the German ONCONET example designed, specified, and implemented by the author's department and German as well as international project partners.

10 Security Enhanced EDI Communication

10.1 Introduction

This chapter considers the basics of communication security, looking for common threats and solutions in the framework of the generic paradigm of security concepts and their relationship to security services, mechanisms, algorithms, and data (Chapter 6). As mentioned in that chapter, the implementation must consider specific protocols and products in the framework of the given environment, the solution selected, and the measures intended.

Reflecting communication in a very general and generic way according to Chapter 6 as

- the basis of information which must be exchanged at least in the phase of information use,
- the interrelation between components
  - at any granularity level including single objects, more complex components, component packages, subsystems, systems,
  - at any level of abstraction as business components, logical components, and technical components,
- interaction between any principal class,

communication fits in any architectural approach.

Furthermore, the paradigm proposed enables a generic and systematic way for analysis, design, and implementation of security services in health information systems facilitating the different users' views as well as their involvement in specification, realisation, management, and use of such secure systems.

The results presented in this chapter have been elaborated within the MEDSEC project [MEDSEC_WWW] and have been published in [Blobel et al. 1998a,b]. Furthermore, they have provided the fundamentals of HL7 standardisation activities on EDI security and influenced the ANSI work on that issue. This generic communication security approach based on the generic and systematic basics developed has fertilised the CEN standards on communication security [CEN ENV 13729] and the work in ISO TC 215, by that way exploiting European projects results in the standard as well as in the application domain. Due to the character of recommendations and specifications the chapter has, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are used in this document as described in RFC2119.

Originally starting with security enhanced HL7, the Magdeburg Medical Informatics Department efforts within the MEDSEC projects lead to an open EDI security framework presented in the next sections. Following, the essential parts of both "Standard Guide for Specifying EDI (HL7) Communication Security" (Chapter 10.2) and "Standard Guide for Implementing EDI (HL7) Communication Security" (Chapter 10.3) will be presented to the reader. Both standard guides became 1999 informative parts of the ANSI accredited HL7 standard.

10.2 Standard Guide for Specifying EDI (HL7) Communication Security

10.2.1 Scope

This Standard Guide gives the framework for secure end-to-end communication of EDI messages focusing on HL7. It is based on the common security model that distinguishes the concepts of communication security as rather globally controlled and application security as rather locally controlled. The concepts of quality and safety are not considered. Each of these concepts defines a set of security services, which are provided by sets of security mechanisms based on security algorithms applied to data. The different levels of granularity allow views of different groups of users — including medical users, system administrators, and implementers — within the same specification framework. Additionally, for implementation, the protocol-services-mechanisms relationships with respect to standards and products also have to be considered.

The Standard Guide starts with the specification of internal security services needed for the provision of secure communication between information systems. External security services, like services provided by Trusted Third Parties (TTPs) to facilitate trustworthiness between the principals involved in communication, such as key management, registration services, naming services, certification service, directory services or secure time services, as well as security services for application security, such as authorisation, access control, data element security, data base security and audit, are not considered. These external security services are outside the scope of the present guidance, which only deals with secure communication of EDI messages applying communication security. Communication security includes the assembling and merging of already secured data elements (done by application security services as the integrity of data and the accountability for data and procedures) to complete the security-enhanced EDI messaging. Because the EDI protocols specify only the syntax and semantics of messages exchanged, but not the network infrastructure used, the importance of service availability is not considered here.

Reacting on threats (active users' interactions) or vulnerabilities (systems' behaviour), the security services defined provide the link between the security requirements and objectives as described by security policies, and the security mechanisms and management needed to satisfy these requirements. Each of the security services can be implemented by one or more types of security mechanisms according to different levels of security needed by different policies and applications. The security policy specifies, among other things, the legal, organisational and social business framework, the analysed threats, accepted risks, and intended organisational and technical solutions. If systems of different organisational and/or policy domains communicate, policy bridging is required. The policy agreed upon defines legal, organisational and technical security issues and the functionality permitted.

Considering the granularity of services and mechanisms, and abstracting from the specific and highly dynamic implementation details — such as using various cryptographic mechanisms of different strengths for implementing security mechanisms; using different security infrastructures available, such as public key or symmetric key infrastructure; or using several communication protocols on different layers, such as, HTTP, SMTP or FTP) — this specification follows a generic and open architectural approach. Thus, it is very flexible in terms of composition of services needed to protect health information systems from the security threats and risks according to the specific healthcare processes and environments.

10.2.2 EDI Communication Security Services

Following the concept of communication security, a set of basic security services is required for the security enhancement of EDI messages using wrapping techniques. The selection and composition of these services is needed to protect health information systems of

the security threats and risks according to the specific healthcare processes and environments. Some of the security services are foreclosing security breaches as principal authentication and confidentiality, while other services like integrity or accountability give only the evidence that an attack has taken place without preventing it technically.

10.2.2.1 Threats and Security Services

In the EDI environment, the threat model consists of at least two principals that are authorised to perform message transmissions to each other using several communication protocols over various infrastructures. Threats are active user (attacker) interactions that cause system vulnerability. According to the security policy, threats, vulnerabilities and accepted risks determine the security requirements that are fulfilled by appropriate security services. The following consideration is based on the common security model, which distinguishes the concept of communication security as rather globally controlled and the concept of application security as rather locally controlled. Each of these concepts defines a set of security services, which are provided by sets of security mechanisms based on security algorithms applied to data. The different levels of granularity allow views of different groups of users, including medical users, system administrators, and implementers, within the same specification framework. Additionally, for implementation, the protocol-services-mechanisms relationships with respect to standards and products have to also be considered.

An unauthorised principal may try to attack the communication system using passive techniques, such as monitoring, listening and sniffing of data system exploration, or traffic analysis, or active techniques, such as creation, insertion, deletion and replay of data. This may enable the intruder to perform masquerading. A short summary of threats and security services is given in Table 10.1.

Table 10.1: Threats and Security Services in the Context of Communication Security

Threats	Security Services
Masquerading (unauthorised use of authorised services)	Principal identification and authentication
Data manipulation	Integrity
Concealment or manipulation of data origin	Accountability in the sense of non-repudiation of origin
Repudiation of receipt	Accountability in the sense of non-repudiation of receipt
Disclosure of data	Confidentiality

10.2.2.2 Security Services and Security Mechanisms

Reacting on threats (active users' interaction) or vulnerabilities (systems' behaviour), the security services defined provide the link between the security requirements and objectives as described by security policies, and the security mechanisms and management to satisfy these requirements. Each of the security services can be implemented by one or more types of security mechanism (the multiplicity is 1:n) according to different levels of security needed by different policies and applications. For security policies, policy domains communicate, policy bridging it might be referred to Chapter 6. In general, security services are independent of special scenarios and implementations as they define a set of security functions.

For health information systems, internal and external security services can be distinguished from each other. Internal security services describe functions provided by communicating and co-operating information systems for the provision of communication security.

External security services are services provided by Trusted Third Parties to facilitate trustworthiness between the principals involved in communication and co-operation. These services (e.g. key management, registration services, naming services, certification services, directory services or secure time services) as well as application security services (such as authorisation, access control, integrity and confidentiality of data, accountability for data and procedures, audit) are not discussed here. However, sometimes data secured by such services need to be communicated to facilitate the security management or to verify accountabilities.

The following table (see Table 10.2) lists the internal security services that EDI SHOULD offer for the secure communication of messages including the security mechanisms used to enforce them. Because the EDI protocols specify only syntax and semantic of messages exchanged, but not the network infrastructure used, the important services availability is not considered here.

Table 10.2: Security Services and their enforcing Security Mechanisms

Security Services	Security Mechanisms	
	Asymmetric Techniques	Symmetric Techniques
Principal Identification and Authentication	Digital Signature, TVPs	Encryption, cryptographic check value (MAC), TVPs
Data Origin Authentication	Digital Signature, cryptographic check value, DN	Encryption, cryptographic check value (MAC), DN
Integrity	Digital Signature, cryptographic check value	Encryption, cryptographic check value (MAC)
Confidentiality	Encryption	
Accountability	Security Audit (using reports, log files, receipts, time stamps and distinguished names)	
Non-repudiation[33] (of origin and receipt)	Digital Signature, cryptographic check value, time stamps, DN	Encryption, cryptographic check value (MAC), time stamps, DN

The client/server-based communication protocols used for transmission of EDI messages MUST at least provide principal authentication and data integrity (including data origin authentication). This set of required services is called *minimal set of required security services*.

The implementation of the security mechanisms by specifications, algorithms and products is dependent on the state of the art, the development of (new) technologies, and their availability to potential attackers. Therefore, it is a highly dynamic procedure. Especially the Internet environment raises various possibilities of new attacks, challenges, and countermeasures resulting in new security techniques whereas the security services and mechanisms itself are rather stable. Thus, a correct implementation of security mechanisms is REQUIRED (i.e. that the mechanisms are covered completely by adequate security techniques as cryptographic algorithms for software and technical means for hardware).

[33] Non-Repudiation is a part of the accountability service.

10.2.2.3 Architectural Placement of Security Services and Security Protocols

The protocol stack reflects the fact that the communication functions are complex and usually divided into independent layers or levels. A stack is a collection of protocol layers that implement network communication. The protocol associated with each layer communicates only with the layers immediately above and below it, and assumes the support of underlying layers. Lower layers are closer to the hardware and higher layers are closer to the user. The number of layers and tasks that each layer performs depends on which stack is used (e.g. OSI, TCP/IP).

The protocol stack is a set of operations that work together to create a seamless path across the layers. As data proceeds from one station to another, the data first begins to travel down the protocol stack until it reaches the physical layer. At the physical layer, the data is placed on a medium (i.e. copper, fibre optics, wireless). The data then traverses the network until it reaches the receiving station where the data travels up the protocol stack. As the data traverses the network, it may be reformatted (protocol adapted) to the protocol being used in a particular network segment [ProtStack].

The applicability of security services is not bound to a specific layer of the OSI protocol stack (ISO7498-2, ISO10181), but from the communication protocol perspective, only four levels are needed to be distinguished as application layer, transport layer, network layer, and data link layer ([Ford, 1994], Chapter 3). Abstracting from the placement on different communication layers and choosing different sets of security services, each existing and upcoming security protocol can be described completely. As for example confidentiality, integrity, and principal authentication are selected and placed on the transport layer, the security protocol TLS (see Chapter 10.2.2.3.2) is specified that provides these security services for the session layer and beyond of it. The following Table 10.3 gives an overview of existing protocols usable for secure communication. The table identically repeats Table 6.2 to facilitate the legibility of the paper.

Following the EDI client/server network architecture using communication servers for end-to-end communication where applications meet and exchange their messages, the security services providing EDI communication security MUST be placed on the transport layer or application layer of each principal. Additional protection MAY be applied using security services provided by protocols located at the lower layers (placed on the network layer or data link layer). Placing the services on the application layer allows security protocol elements that are dependent of the application (e.g. HTTP or FTP). Security protocol elements on the transport layer provide protection on an end-system basis. Establishing end-to-end security REQUIRES that the end systems are trusted, but all underlying communication network(s) MAY be untrustworthy.

For example, HL7 communication security, i.e. wrapping HL7 messages by security envelopes when in transit as shown in Figure 10.1, can be achieved by using HL7 external communication protocols with security implemented already (as SHTTP or SFTP) or by securing the HL7 lower layer protocols located at the session layer (based on the sockets interface) and beyond of it.

For interoperability reasons, only standard documents available as ISO Standards, IETF/IESG Internet Standards (RFCs), IETF Internet Drafts (IDs), NIST publications (NIST FIPS PUB) or similar MUST be used for the security enhancement of existing protocols (*standards conformance*).

Table 10.3: Security Services Provided by Protocols on Different ISO-OSI Model Layers[34]

[34] A list of abbreviations is given in the annex.

Security Services / OSI Layers	Confidentiality	Integrity	Entity Authentication	Data Origin Authentication	Non-Repudiation of Origin	Non-Repudiation of Receipt
Data Link	SILS/SDE, PPTP[35], L2TP	SILS/SDE, L2TP	PPTP, L2TP, L2F	SILS/SDE, L2TP	–	–
Network	IPSEC, NLSP	IPSEC, NLSP	IPSEC, NLSP	IPSEC, NLSP	–	–
Transport	SOCKS, TLSP, SSL, TLS, PCT, SSH	SOCKS, TLSP, SSL, TLS, PCT, SSH	SOCKS, TLSP, SSL, TLS, PCT, SSH	TLSP	–	–
Application	SHTTP, SPKM, MHS, MSP, PEM, SFTP, PGP/MIME, MOSS, S/MIME, PKCS#7	SHTTP, SPKM, MHS, MSP, PEM, SFTP, PGP/MIME, MOSS, S/MIME, PKCS#7	SHTTP[36], SPKM, SFTP	SHTTP, SPKM, MHS, MSP, PEM, SFTP, PGP/MIME, MOSS, S/MIME, PKCS#7	SHTTP, SPKM, MHS, MSP, PEM, SFTP, S/MIME, ESS	SPKM, MHS, MSP, SFTP, S/MIME, ESS

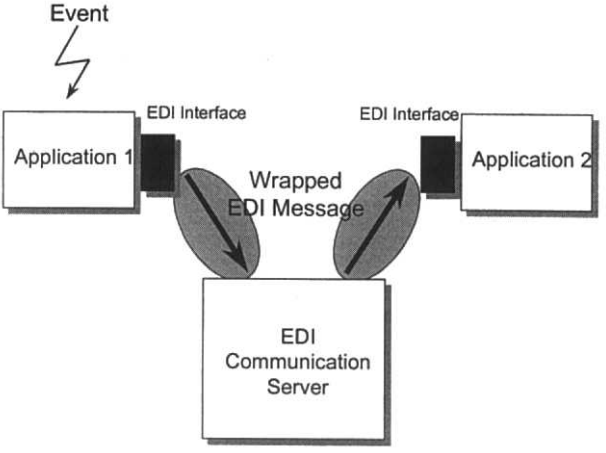


Figure 10.1: EDI Message Security

In the next sections are described the basically protocols usable for secure communication regarding the remarks mentioned above.

10.2.2.3.1 IPv6

TCP/IP, the protocol suite on which the global Internet and corporate intranets are based, is decades old and therefore breaking. IPv4, the current version of the Internet Protocol, is reaching the end of its life. The main reasons are:

[35] PPTP does not address any security issues in the current version, but end-to-end security is addressed by PPP which is tunnelled by PPTP through an IP network.

[36] Only the client is authenticated to the server by showing that he is able to apply message enhancement according to the security requirements of the server.

- Limitations in the number of devices it can address;
- Growing demand for new functionality;
- Lack of essential security features.

IP lives in end computer systems and in the routers that connect them. When an application on one end system wants to send data, it encapsulates the message in an IP protocol data unit (PDU) which traverses a path of networks connected by routers to reach its intended target.

The key services provided by IP in this exchange are as follows:

- Addressing: the PDU must inform each router it encounters of its destination;
- Packetizing: physical networks specify a Maximum Transmission Unit (MTU), or packet size, which PDU's must observe;
- Service class: specifies treatment of PDU relative to other traffic as regards priority, reliability and delay;
- Security: PDUs can be encrypted and contain signature and authentication data.

The design of the new protocol IPv6 (also called IP next generation or IPng) pays special attention to addressing and security services. It also improves overall network performance and provides enhanced service class options.

The address field size in IPv6 is increasing from 32 bits to 128 bits and therefore 296 times bigger than today's IP address space. IPv6 also describes rules for three modes of addressing: unicast (one host to one other host), anycast (one host to the nearest of multiple hosts) and multicast (one host to multiple hosts). Unicast addresses target individual hosts. Several variants of unicast are allowed, including an IPv4-compatibility mode intended to provide a smooth migration path. Anycast addressing is a refinement of unicast that streamlines routing. The address provides the possibility of sending a message to the nearest of several possible gateway hosts with the idea that any one of them can manage the forwarding of the packet to others. Multicast allows messages to be sent to a predefined group of unicast addresses with a single multicast address.

IPv6 provides improved performance in three ways:

- Reduced number of header fields. The so-called packet header in IPv6 is, at 40 bytes, actually longer than the IPv4 header (20 bytes minimum), but it contains fewer fields. This expedites processing by the router.
- Fixed-length packet header. IPv4 allows a number of options in the packet header that can change its size. IPv6 has a 40 byte header, which again streamlines the work done by routers.
- The IPv6 header now includes extensions that allow a packet to specify a mechanism for authenticating its origin, for ensuring data integrity, and for ensuring privacy.
- No fragmentation allowed. IPv6 accommodates the MTU requirements of intervening networks at the source end, using an algorithm to discover the transmission path and lowest MTU. This saves the overhead of fragmentation and reassembly.

Finally, an impressive array of security features has been built into IPv6. In spite of the given abundance of proven application-level mechanisms like S/MIME, Privacy Enhanced Mail (PEM), S-HTTP and SSL this is necessary because of the following: IP-level security works for all applications, whether aware or ignorant of security concerns. IPv6 supports two security functions: authentication and privacy. The authentication mechanism ensures that a received packet was in fact transmitted by the source identified in the packet header, and not by a forger or interloper. As a corollary, authentication ensures that the message has

not been tampered within transit. Privacy, the assurance that a message can be seen only by authorised parties, is implemented by strong encryption.

The set of security services offered includes access control, connectionless integrity, data origin authentication, protection against replays (a form of partial sequence integrity), and confidentiality. These services are provided at the IP layer, offering protection for IP and/or upper layer protocols. Therefore they can be used by any higher layer protocol (e.g. TCP, UDP, ICMP, BGP).

These objectives are met through the use of two traffic security protocols – the Authentication Header (AH) and the Encapsulating Security Payload (ESP) – and by using cryptographic key management procedures and protocols. The set of IPsec protocols employed in any context, and the ways in which they are employed, will be determined by the security and system requirements of users, applications, and/or sites/organisations.

When these mechanisms are correctly implemented and deployed, they ought not to adversely affect users, hosts, and other Internet components that do not employ these security mechanisms for protection of their traffic. These mechanisms are also designed to be algorithm independent for allowing easy integration with new, more powerful, algorithms once they are available. This modularity also permits selection of different sets of algorithms without affecting the other parts of the implementation. For example, different user communities may select different sets of algorithms if required.

A standard set of default algorithms is specified to facilitate interoperability in the global Internet. The use of these algorithms, in conjunction with IPsec traffic protection and key management protocols, is intended to permit system and application developers to deploy high quality, Internet layer, cryptographic security technology. [IPv6]

10.2.2.3.2 SSL and TLS

The Secure Sockets Layer (SSL) was developed by Netscape Communications Corporation to provide security and privacy over the Internet. The protocol supports server and client authentication and is application independent, allowing protocols like HTTP, FTP and Telnet to be layered on top of it transparently. The SSL protocol is able to negotiate encryption keys as well as authenticate the server before data is exchanged by the higher-level application. It maintains the security and integrity of the transmission channel by using encryption, authentication and message authentication codes (Figure 10.2). The protocol basically consists of the following components: Record Protocol, Handshake Protocol and Alert Protocol. The Handshake Protocol consists of two phases, server authentication and client authentication, with the second phase being optional. In the first phase, the server, in response to a client's request, sends its certificate and its cipher preferences. The client then generates a master key, which it encrypts with the server's public key, and transmits the encrypted master key to the server. The server recovers the master key and authenticates itself to the client by returning a message encrypted with the master key. Subsequent data is encrypted with keys derived from this master key. In the optional second phase, the server sends a challenge to the client. The client authenticates itself to the server by returning the client's digital signature on the challenge, as well as its public key certificate. A variety of cryptographic algorithms are supported by SSL. During the "handshaking" process, the RSA public key cryptosystem is used. After the exchange of keys, a number of ciphers are used. These include RC2, RC4, IDEA, DES, and triple-DES. The MD5 message-digest algorithm is also used. The public key certificates follow the X.509 syntax [SSL].

SSL is available as a programming interface on top of (or augmenting) the sockets programming interface which in turn is an interface to TCP. It is possible to see SSL as a layer between the application and TCP, not as a replacement for TCP. SSL offers the following benefits:

- Eavesdroppers cannot read the data;
- Either side can verify the identity of the other side. This is accomplished by one side presenting a "Certificate" to the other;
- Data integrity is assured, any change to a byte will invalidate the checksum on each SSL chunk.

SSL is useful for standard EDI connections that use TCP today. A crucial piece of technology is the digital certificate. The digital certificate proves that you know a per-user secret key. Because this secret key is never transmitted, it is operationally easier to actually keep the secret key secret. Thus, a recipient can place high reliance on the assumption that the holder of the certificate (and the secret key) is who the certificate says it is. It is difficult to forge certificates, or steal meaningful private keys, therefore we allocate trust to the certificate authority. SSL appears to be a good choice to solve the problem of authentication and privacy between two sites using TCP. However, SSL is unsuitable for "store and forward" environments. Once the data is read off the wire, all knowledge/proof of its origin is lost. When message routers are involved, SSL authentication only has the ability to authenticate the last link. Without changes on the Hub, it is not possible to verify that the sender is who the EDI message says it is. The message router would have to check the MSH segment against the SSL certificate.

The TLS protocol was developed based on the SSL protocol. The differences between TLS and SSL are not dramatic, but they are significant enough that they do not interoperate.

The TLS protocol provides communications privacy over the Internet. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery.

The primary goals of the TLS protocol are to provide privacy and data integrity between two communicating applications and:

- Cryptographic security: TLS should be used to establish a secure connection between two parties.
- Interoperability: Independent programmers should be able to develop applications utilising TLS that will then be able to successfully exchange cryptographic parameters without knowledge of one another's code.
- Extensibility: TLS seeks to provide a framework into which new public key and bulk encryption methods can be incorporated as necessary. This will also accomplish two sub-goals: to prevent the need to create a new protocol (and risking the introduction of possible new weaknesses) and to avoid the need to implement an entire new security library.
- Relative efficiency: Cryptographic operations tend to be highly CPU intensive, particularly public key operations. For this reason, the TLS protocol has incorporated an optional session caching scheme to reduce the number of connections that need to be established from scratch. Additionally, care has been taken to reduce network activity.

The protocol is composed of two layers: the TLS Record Protocol and the TLS Handshake Protocol. At the lowest level, layered on top of some reliable transport protocol (e.g. TCP), is the TLS Record Protocol. It provides connection security that has two basic properties:

- The connection is private. Symmetric cryptography is used for data encryption (e.g. DES, RC4). The keys for this symmetric encryption are generated uniquely for each connection and are based on a secret negotiated by another protocol. The Record Protocol can also be used without encryption.
- The connection is reliable. Message transport includes a message integrity check using a keyed MAC. Secure hash functions (e.g. SHA, MD5) are used for MAC computa-

tions. The Record Protocol can operate without a MAC, but is generally used only in this mode while another protocol is using the Record Protocol as a transport for negotiating security parameters.

The TLS Record Protocol is used for encapsulation of various higher level protocols. One such encapsulated protocol, the TLS Handshake Protocol, allows the server and client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before the application protocol transmits or receives its first byte of data. The TLS Handshake Protocol provides connection security that has three basic properties:

- The peer's identity can be authenticated using asymmetric, or public key, cryptography (e.g. RSA, DSS, etc.). This authentication can be made optional, but is generally required for at least one of the peers.
- The negotiation of a shared secret is secure: the negotiated secret is unavailable to eavesdroppers, and for any authenticated connection the secret cannot be obtained, even by an attacker who can place himself in the middle of the connection.
- The negotiation is reliable: no attacker can modify the negotiation communication without being detected by the parties to the communication.

One advantage of TLS is that it is application protocol independent. Higher level protocols can layer on top of the TLS Protocol transparently. The TLS standard, however, does not specify how protocols add security with TLS. The decisions on how to initiate TLS handshaking and how to interpret the authentication certificates exchanged are left up to the judgement of the designers and implementers of protocols which run on top of TLS [TLS].

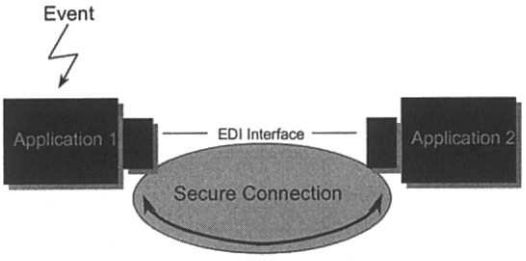


Figure 10.2: EDI Secure Channel

SSL or TLS enables secure communication between security unaware applications but also secure interactive communication and co-operation, e.g. in real-time environments.

10.2.2.3.3 EDI Interfaces and Lower Layer Protocols

EDI interfaces exist at the application layer of the OSI model and are usually located at the application, but sometimes also at the communication server. Thus, they usually require the support of some lower level protocol (LLP) that provides an interface between EDI and the network. It is important to select an LLP that meets the needs of the interface and fits into the overall telecommunications strategy and architecture. The LLPs are built up assembling various modules (e.g. initiating module, accepting module, encoding rules module, communication module). The communication module delivers a message from the source to its destination using the network socket interface. If this delivery mechanism is used for message transmission instead of external communication protocols, it MUST be secured by the security services defined to gain communication security.

In Appendix C, the "HL7 Implementation Support Guide" for HL7 Version 2.3 (final version 6/98) defines lower layer protocols usable for non-secured communication of HL7 messages. In this appendix, the minimal lower layer protocol (MLLP), the hybrid lower

layer protocol (HLLP), and an ANSI X3.28 based data link protocol are specified. Moreover, the HL7 sequence number protocol is explained and pseudo code for a TCP-based implementation of LLPs is given. More information about the LLPs of HL7 can be found in the "HL7 Implementation Support Guide" (especially Chapter 1.5, 2.5, 3.4, and 3.6 as well as appendix C)

10.2.2.4 Communication Protocol Security Requirements

For secure end-to-end communication between two principals in an EDI environment the following security services are needed:

- Principal authentication (applications and systems);
- Data origin authentication;
- Confidentiality;
- Integrity;
- Non-repudiation of origin;
- Non-repudiation of receipt;

In general, communication protocols distinguish control data from message data. Control data is used to emit protocol specific commands from the sender to the receiving principal and to reply codes back to the originator for status report. More structured and differentiated protocols, such as FTP, separate these kinds of data and use two connections, called control connection and data connection, simultaneously.

10.2.2.4.1 Control Data

This specification REQUIRES the control data to be secured by integrity. Additionally, the control data SHOULD be protected by the security services non-repudiation of origin and non-repudiation of receipt. Since the control data is a well-known and often small set of commands, confidentiality SHOULD NOT be applied. Furthermore, the control data MUST be armed against loss of data bits in environments not capable of full binary transport. This may be achieved by various conversion techniques such as Base64-Encoding or Quote-Printable-Encoding. Finally, before applying the integrity service, the data MUST be canonicalised after the encoding process to prevent system dependency (like different end of line codes for PC and Unix environments; *system independence*) leading to invalidation of the integrity code (digital signature or MAC).

10.2.2.4.2 Message Data

The data connection delivering the message data MUST be secured by the integrity service. Confidentiality, non-repudiation of origin, and non-repudiation of receipt SHOULD be applied as well to gain more security. Switching between different operation modes such as plain text, encrypted-only, signed-only, or signed-and-encrypted, SHOULD be possible according to the security policy given (*operation modes independence*). Moreover, the message data MUST be character converted and canonicalised to prevent loss or manipulation of certain EDI characters (as, e.g., the HL7 segment terminator) leading to invalidation of the integrity code (digital signature or MAC).

For the correct handling of received data concerning the operation modes, content encoding, and other parameters, insertion of the message data into a cryptographic syntax capable of wrapping and feature negotiation is REQUIRED. For this purpose, an encapsulation scheme using MIME entities consisting of headers and bodies MAY be used (e.g. PGP/MIME, S/MIME, or MOSS). The recipient MUST be able to recognise the data received as an EDI interchange.

When transporting signed data by Internet (HTTP, SMTP) or end-to-end in non-MIME environments, gateways are generally not aware of security encapsulation schemes and therefore mistreating the data or even applying conversions to the structure and its contents according to the local format. Thus, either the original message could not be reconstructed and the signature could not be verified, or the signature verification fails. Additional measures SHOULD be applied to avoid this behaviour (e.g. using a special wrapping mechanism as defined by S/MIME).

Fulfilling these requirements, also other data than HL7 messages are enabled to be sent securely over the data connection (*data-type independence*), i.e., other EDI messages (e.g. X12, xDT) or non-EDI data (arbitrary binary) can be wrapped and transmitted, too. Thus, the secure communication protocol can be used in *any desired environment* for data delivery. Data-type independence means that the receiving principal MUST be able to recognise the type of data received. For that reason, if inserting an HL7 message into an encapsulation scheme, header information identifying the message content MUST be supplied (like content-type for MIME). This scheme SHOULD be capable of specifying additional parameters to state encoding rules (syntax) or other information (e.g. version number). Another possible solution is to map the HL7 message – including the additional protocol parameters – into a ANSI X12 message, using the standardised mapping rules, and to insert the result into an encapsulation scheme as defined in RFC1767 (MIME encapsulation of X12 and EDIFACT objects using the content-types `application/EDI-X12` and `application/EDIFACT`). When operating in an *HL7-environment*, data type independence SHOULD NOT be attended, since the HL7 interface definitely knows that only HL7 message data is sent between applications.

For *large file processing*, compression of EDI messages MAY be done before encryption, after applying the integrity service if needed. Applying compressing before encryption strengthens cryptographic security since repetitious strings are reduced due to their redundancy. If compression is used, additional data MUST be provided to convey compression information.

10.2.2.4.3 Authentication

Before any control or message data is exchanged, except the control command to request authentication, principal identification and authentication of applications or systems MUST be performed as described in section 0. Again, for interoperability reasons, the authentication tokens exchanged MUST be character-converted and canonicalised.

In the context of some EDI specific protocols or user-applications interactions based on standardised EDI protocols for enabling open communication independent of platform and environment of that application, also a human user may occur as a principal instance. Than, other security services, mechanisms and techniques may be used as, e.g., the European security infrastructure based on electronic identity cards (smart cards, Health Professional Cards = HPC) and corresponding TTP services.

10.2.2.4.4 Cryptographic Algorithms

If the cryptographic algorithm to be applied is not an algorithm approved by any national authority or other community authorities, then it SHOULD be an algorithm registered and identified using the procedures described in ISO/IEC9979. The communication protocol used SHOULD be independent of the underlying cryptographic mechanisms (*cryptographic mechanism independence*) and cryptographic message syntax (*cryptographic message syntax independence*). It SHOULD allow the negotiation of different algorithms, operation modes, and cryptographic message syntax as well as the selection of different technical means (as smart card, biometric device, directory server, CRL server).

10.2.2.4.5 Communication and Networking

Basically, communication of EDI messages SHOULD be carried out by direct link connections. Solutions are protocols based on store-and-forward delivery (asynchronous as SMTP, MHS) or real-time delivery (synchronous as FTP, FTAM). Since the delivery of EDI or cryptographically enhanced data objects is independent of the communication protocol, there are many different protocols and options used for such solutions.

The HL7 Standard is defined in terms of the client/server model (remote operation) and is therefore applicable to file transfers (batch processing). One or more messages MAY be encoded according to the Encoding Rules, grouped in a file and transferred using external communication protocols as FTP, or any other file transfer protocol. Responses MAY be grouped in a file and similarly transmitted. General mechanisms for the batch transmittal of HL7 messages are provided in the "HL7 Implementation Support Guide" (Chapter 1.5) and in the HL7 Standard (Chapter 2.23.3). Following the HL7 paradigm of bi-directional, synchronous communication where applications meet in a rendezvous and exchange their messages, security protocols capable of synchronous message transfer featuring the given security services above MAY be preferred. Transfer of the EDI interchange can take place in real time, without any deferred delivery. The data transmission is point-to-point, with no requirement for temporary storage anywhere. The size of interchanges can be very large and the exchange could be initiated by both, the sender or the recipient (e.g. for data collection). Modern communication servers support the variety of protocols.

For reliable transmissions, (connection-oriented) TCP/IP based networks are REQUIRED.

10.2.2.4.6 Protocol Model Implementation

Among the security considerations described, the specification of the protocols used — like FTP, TCP, and IP — contains a number of mechanisms inherent to their protocol model that can be used to compromise network security. Thus, there are many so-called Internet attacks based on infrastructure weakness; for instance DNS spoofing, ICMP bombing, source routing (IP spoofing), TCP sequence guessing/hijacking, TCP splicing, FTP bouncing, racing authentication and denial of service.

Attacks arising from the weakness of the process protocol and the underlying protocols SHOULD be addressed by appropriate countermeasures in the implementation model. For example, the Computer Emergency Response Team (CERT Coordination Center, [CERT]) studies such Internet security vulnerabilities, provides incident response services, publishes a variety of incident reports and security alerts, and develops information to improve security.

Racing authentication, which is based on faster authentication of the attacker than the victim, SHOULD be prevented by strong authentication based on challenge-response protocols. Moreover, a restriction to one simultaneous login of the same principal and to the total number of control connections possible at once SHOULD be carried out.

To protect against FTP bouncing, which is namely the misuse of the PORT command where the attacker is acting as a server, the server SHOULD NOT establish connections to arbitrary machines (for instance to a second FTP server called proxy FTP) and to ports on these machines. The server SHOULD ensure that the received IP address, which specified in the PORT command, must match the client's source IP address for the control connection. to prevent this attack from occurring at all, FTP driven protocols SHOULD use the PASV command instead of the PORT command to establish data connections.

Furthermore, the server SHOULD disallow data connections on TCP-ports that are well-known ports (port 0 to 1023) or registered ports (1024 to 49151). Only dynamic, private ports (port 49152 to 65535) SHOULD be allowed.

Hence, a port scan against another site hiding the true source and bypassing access controls like firewalls (for instance bouncing to a well-known port) cannot be performed.

Random local port (private) numbers SHOULD be used for the data connection to address port number guessing. Guessing the next port number is much easier when using simple increasing algorithms (for example: next port = old port + constant number) enabling attacks like the denial of a data connection or hijacking a data connection to steal files or insert forged files.

TCP splicing, which is the hijacking of the connection on the TCP layer, MAY be prevented by the application of level end-to-end confidentiality since the attacker cannot generate messages that will decrypt to meaningful data. When confidentiality is applied, network sniffing does not pay, but from the TCP layer downwards creating a traffic flow analysis evaluating packet headers and trailers is still possible. Traffic flow confidentiality (e.g. address-hiding or traffic padding) MAY be provided by applying confidentiality on the data link layer. In the context of HL7 EDI, this threat may be overestimated and could be neglected.

In addition to the authentication procedures given in Chapter 0, restrictions based on network addresses MAY be provided. The server accepts only connection requests from predefined IP addresses within authorised organisations and confirms that this address matches on both the control connection and the data connection. Authentication MUST NOT rely on IP address authentication only. Relying solely on an IP address authentication makes an attack like source routing of IP packets (IP spoofing) is possible.

To address DNS spoofing, hostname to IP address resolution or vice versa (DNS) SHOULD NOT be performed. It is RECOMMENDED that the destination machine be caught by the IP address directly.

For the detection of compromises like denial of service attacks and other attacks, the server SHOULD keep reports and log all activities, including connection attempts, disconnection, command executions and others. Reports and logs SHOULD be integrity and confidentiality protected.

10.2.2.5 Authentication Service Requirements

10.2.2.5.1 Purpose

The authentication service *provides* assurance of the identity of a principal. When a principal claims to have a particular identity, the authentication service will provide a means of confirming that this claim is correct. Generally, a principal to be authenticated proves its identity by showing its knowledge of a secret. The mechanisms are defined as exchanges of data between principals, and, where required, with a Trusted Third Party.

Authentication is the most important of the security services; all other services depend upon this assurance. The identity authenticated is used for accountability, data origin authentication, and access control depending on the assured knowledge of identities.

10.2.2.5.2 Service Requirements

This service MUST use cryptographic techniques that establish *strong authentication*; weak authentication (e.g. passwords) SHALL NOT be used. The authentication framework given by ISO/IEC10181-2 is on top of a hierarchy of authentication standards that provide concepts, nomenclature, and a classification for authentication models. Directly below, standards such as ISO/IEC9798 provide a particular set of these methods in more detail. Finally, at the bottom, standards such as ISO/IEC9594-8 (ITU-T Recommendation X.509) use these concepts and methods in the context of a specific application or requirement (ISO/IEC9594-8 was initially developed for use with the directory service). NIST

FIPS PUB196 is based on ISO/IEC9798-3 and might be helpful for implementation details like token formatting. ISO/IEC9798 specifies different protocols that address both unilateral and mutual authentication by mechanisms using

- symmetric encipherment algorithms (ISO/IEC9798-2),
- digital signature techniques (ISO/IEC9798-3, ISO/IEC9594-8, NIST FIPS PUB196),
- a cryptographic check function (ISO/IEC9798-4), or
- asymmetric zero knowledge techniques (ISO/IEC9798-5).

For a higher level of assurance, mutual authentication carried out by challenge-response protocols, using symmetric or asymmetric security techniques, is RECOMMENDED. In large networks, mechanisms that use symmetric techniques, such as Kerberos, depend upon trusted on-line authentication servers to distribute public key certificates and CRLs. Mechanisms that use asymmetric techniques require off-line servers — which need not to be trusted — for the distribution of public key certificates and CRLs. Due to the inherent and well-known disadvantages of symmetric techniques (secret key cryptography), asymmetric techniques, such as digital signature or zero knowledge techniques, SHOULD be used for open networks like the Internet, where many principals communicate with each other. The application of this technique requires the management of security certificates (e.g. using a directory server) inside a public key infrastructure (PKI) that has to be established. The authentication depends on the successful verification of the digital signature, which is bound on the key pair. Thus, there is a requirement to validate the public key used to verify a claimed identity. The commonly used mechanism for validating a public key is the use of certificates issued by a trusted certification authority (CA). This technique is described in ISO/IEC9594-8 (ITU-T Recommendation X.509).

The authentication service MUST be built up using the services of principal authentication and data origin authentication. This implies that the integrity service MUST be applied. Assurance of origin and receipt (non-repudiation of origin, non-repudiation of receipt) SHOULD be used for a higher level of security. Confidentiality of the authentication tokens is NOT RECOMMENDED in order to avoid unnecessary interactions between security mechanisms, which may result in security flaws.

The *principal authentication service* assures that a principal, which has a specific communication relationship with the verifier, is the one claimed. Principal authentication techniques generally involve an exchange of cryptographic protected authentication data, which is used to validate a claimed identity.

The *data origin authentication service* assures that a principal is the source of data as claimed. This is provided by use of digital signature (asymmetrical techniques) or encryption (symmetric techniques). Applying digital signatures, the principal signing the data cannot deny that he or she applied the signature, since the principal is the only one with knowledge of his or her private key. This method also achieves non-repudiation of origin. Applying encryption, the principal to be authenticated corroborates its identity by demonstrating knowledge of a secret authentication key.

An important factor in authentication exchange techniques is the need to protect against replay of authentication using *time variant parameters* (unique numbers) as time stamps, sequence numbers, and random numbers. *Assurance of continuity of authentication* MUST be provided for the exchanges of data in order to prevent cutting in or taking over after authentication has completed. This can be achieved by carrying out the integrity service over the whole connection period (binding of principal authentication and integrity service), and by performing further authentication exchanges from time to time.

Authentication protocol design needs to take into account an array of possible attack scenarios and provide appropriate countermeasures. To cover all the services required for

authentication (principal authentication, data origin authentication, non-repudiation of origin and non-repudiation receipt) and all possible threats as described in ISO/IEC9798 and ISO/IEC10181-2, each *authentication token* exchanged MUST be completely integrity protected and SHOULD consist of

- a token identifier,
- a sequence number,
- the IP address of sender and receiver,
- the network hardware adapter address (MAC) of sender and receiver,
- the DNs of sender and receiver,
- a state indicator (authentication request or invitation),
- the role of the principal sending the token (initiator, responder),
- time stamps stating token generation and token expiration (in UTC time),
- a random number challenge,
- the MIC of the last authentication token received.

The *implementation* of the authentication service SHOULD give a consistent view on token parameters (the parameters must be distinguishable independent of their position inside the token) and on the token order in order to prevent security flaws. For both, additional information SHOULD be included in the tokens gaining more efficiency due to unambiguousness, and eliminating security threats. For a consistent view on parameters, the tag-length-value encoding SHOULD be used, and for a consistent view on the token order, sequence numbers and token identifiers SHOULD be applied.

As mentioned before, in the context of some EDI specific protocols or user-applications interactions based on standardized EDI protocols for enabling open communication independent of platform and environment of the application, a human user may also occur as a principal instance. Before trustworthiness between principals can be established by authentication, the human user MUST authenticate himself to a cryptographic module of the local end system (*human user authentication*), which then acts as an initiator (client) and performs the integrity service on behalf of the human user toward the responder (server) carrying out *system authentication*. Human user authentication relies on principles of something known (e.g. passwords), something possessed (e.g. chipcards following ISO/IEC7816), characteristics of the individual human user (biometrics), and accepts that an identified TTP has established the human user's identity, or context (e.g. source address). This Standard Guide RECOMMENDS the usage of chipcards (smartcard with cryptographic processor) in combination with biometrics (e.g. fingerprint) and/or PIN codes. For the authentication services described above, the user must keep and protect his private asymmetric key (or a secret symmetric key). This SHOULD be done using chipcards, protected with a PIN code and/or biometrics for reasons of security (the keys are protected in a physical device carried by the user, which cannot be copied and from which a readout cannot be displayed) and mobility (the user can authenticate himself in any environment that has proper devices). Mobility is an essential argument for using chipcards in the healthcare sector for Health Professionals.

10.2.2.6 Confidentiality Service Requirements

10.2.2.6.1 Purpose

The confidentiality service *protects against* information being disclosed or revealed to principals not authorised to read and interpret message data obtaining the information. This service *does not* prevent against the reading of the protected data.

Concerning the granularity of the confidentiality with respect to communication security, this service *applies to all message data transmitted* on the connection. This technique is called wrapping or enveloping of data. Selective field confidentiality, which applies only to designated data fields within a data unit, *is not a matter* of communication security but of application security.

10.2.2.6.2 Service Requirements

A general framework for provision of confidentiality services is given by ISO/IEC10181-5. This Standard Guide defines basic concepts of confidentiality, identifies classes of generic mechanisms and describes confidentiality policies. It neither specifies nor depends on the use of particular mechanisms and algorithms since ISO does not standardise cryptographic algorithms, but rather their procedures for registration (see Chapter 10.2.2.4.4).

Confidentiality SHOULD only be served to the message data delivered by the data connection (see Chapter 10.2.2.4). It MUST NOT be applied on the authentication or control data. Confidentiality SHOULD be provided by the use of strong cryptographic mechanisms employing hybrid techniques. For bulk encryption (*content encryption*), a strong symmetric session key having at least 112 significant key bits (as IDEA or DES3) SHOULD be used and for each transfer of message data another key SHOULD be applied. The session key is protected by asymmetric encryption techniques using at least 1024 key bits (as RSA, El-Gamal or Elliptic Curves = EC, *key encryption/transport*). Content encryption by asymmetric techniques MUST NOT be applied. Modes of operation of symmetric keys SHOULD follow ISO/IEC8372 for 64-bit block ciphers or ISO/IEC10116 for n-bit block ciphers.

The combination of the services of confidentiality and integrity is RECOMMENDED for the transport of message data. The interaction between these security mechanisms and their ordering may result in security weakness (ISO/IEC10181-1 Chapter 10). In general, the integrity service MUST be applied first. If three services are desired (triple wrapping), two integrity services MUST be applied: one before and one after the confidentiality service. Following this approach, in message handling systems, two different message integrity codes can be placed on the data; one computed on the encrypted data (applied by transfer agents) to provide chained non-repudiation, and one on the plaintext (applied by the first sender) to provide data origin authentication.

10.2.2.7 Integrity Service Requirements

10.2.2.7.1 Purpose

Data integrity service *ensures* data consistency while in communication by giving the possibility to *detect* its modification. Changing the value of data includes insertion, deletion, modification, or reordering parts of the data.

This service *does not prevent* the manipulation of data but *allows the detection* of its alteration. Duplication that may be a result of a replay attack *can be neither* inhibited *nor* recognised without additional, unique tokens as random numbers or time stamps.

Concerning the granularity of the integrity with respect to communication security, this service *applies to all the data transmitted* on the connection. Selective field integrity, which applies only to particular data fields within a data unit *is not a matter* of communication security but of application security. However, integrity and accountability dealing with items could be required for exchange.

10.2.2.7.2 Service Requirements

A general framework for the provision of integrity services is given by ISO/IEC10181-6. This Standard Guide defines basic concepts of integrity, identifies classes of generic mechanisms and describes integrity policies. It neither specifies nor depends on the use of

particular mechanisms and algorithms since ISO does not standardise cryptographic algorithms, but rather their procedures for registration (see Chapter 10.2.2.4.4).

It is REQUIRED that integrity be applied to the authentication tokens exchanged, the message data transferred (possibly before its confidentiality protected), and the control data transmitted.

Integrity MUST be achieved by the application of cryptographic techniques. In general, a cryptographic check value (i.e. a message digest computed by a hash function) MUST be calculated over the data in order to be integrity protected. Then, this check value has to be shielded by transformation through an encipherment mechanism (sealing, symmetric cryptography) or by combination with the private key to form a digital signature (asymmetric cryptography). The usage of digital signatures for the provision of integrity by asymmetric techniques is RECOMMENDED.

For calculation of the message digest, keyed (message authentication code, MAC) or unkeyed (called modification detection, MDC) hash function can be used. In general, hash function using MDCs SHOULD follow ISO/IEC10118 and those using MACs SHOULD follow ISO/IEC9797. Unkeyed hash function are based upon block ciphers (as MDCx-DES), the MD4-family (as MD5, RIPEMD-x), or modular arithmetic (as MASH-x), whereas keyed hash functions rely on block ciphers (as MAC-DES-CBC, MAC-IDEA-CBC), hash MACs (as HMAC-MD5, HMAC-RIPE-x, HMAC-SHA-1), or steam ciphers (as CRC-based MAC). The difference between these two types lies in the application of keys. Keyed mechanisms require a secret key as input for the MAC algorithm to calculate a MAC, whereas unkeyed mechanisms apply the key (secret or public) on the MDC that has been previously calculated by the MDC algorithm.

For integrity provision based on asymmetric techniques (i.e. digital signatures) the usage of MDCs based on the MD4-family is RECOMMENDED since these functions are specifically designed for the explicit purpose of hashing, with optimized performance (dedicated hash functions).

When symmetric techniques are applied, the concatenation of the text and the appended message integrity code (either MAC or MDC) has to be sealed (encrypted). The usage of MACs offers the advantage that, should the encryption algorithm be defeated, the MAC still provides integrity. A drawback is the requirement of managing both an encryption key and a MAC key, which may lead to security weakness by unwanted algorithm dependencies. Nevertheless, it should be noted that the MDC is a known function of the plaintext, while a MAC is itself an authenticator secret .

The cryptographic check function, the signing algorithms (as RSA, DSA, ElGamal, or EC) as well as the authentication algorithms (as IDEA or DES) SHOULD offer an appropriate strength.

RSA and related signature schemes SHOULD use formatting of ISO/IEC9796 or PKCS#1 for digital signatures. The digital signature features supplied SHOULD conform to ISO/IEC9594-8 (ITU-T X.509).

Integrity based upon digital signatures with an appendix (the MDC or MAC is appended to the processed data) SHOULD follow ISO/IEC14888. For digital signature schemes giving message recovery, ISO/IEC9796 SHOULD be obeyed. Applying integrity using digital signatures with an appendix, data origin authentication and non-repudiation of origin and receipt can be provided.

Duplication that may result in replay attacks SHOULD be countered by including additional time variant parameters (TVPs), such as random numbers or time stamps in the signature process.

The combination of the services confidentiality and integrity is RECOMMENDED for the transport of message data. Interaction between these security services and their ordering may result in security weaknesses (ISO/IEC10181-1 Chapter 10). In general, the integrity service MUST be applied first. If three services are desired (triple wrapping), two integrity services MUST be applied: one before and one after the confidentiality service. Following this approach in message handling systems, two different message integrity codes can be placed on the data; one computed on the encrypted data (applied by transfer agents) to provide chained non-repudiation, and one on the plaintext (applied by the first sender) to provide data origin authentication.

10.2.2.8 Data Origin Authentication Service Requirements

10.2.2.8.1 Purpose

Data origin authentication (message authentication) is used to authenticate the real source of data, in that it *provides* assurance of the source of data by gluing the originator's identity along with the data using the integrity service. It *does not provide* protection against duplication, reordering, or loss of data. In contrast to non-repudiation, this service is initiated by the data originator that wants to give proof of source against the recipient.

Concerning the granularity of the data origin authentication with respect to communication security; this service *applies to all the data transmitted* on the connection. Selective field origin authentication, which applies only to designated data fields within a data unit *is not a matter* of communication security but of application security.

10.2.2.8.2 Services Requirements

It is REQUIRED that data origin authentication be applied on the authentication tokens exchanged, the message data transferred (before its confidentiality protection), and the control data transmitted.

The data origin can be authenticated by applying digital signature algorithms, MACs, or sealed authenticators. In contrast to MDCs, MACs are themselves secret authenticators. The appended authenticator, MAC, is used along with encryption (see Chapter 10.2.2.7).

Data origin authentication SHOULD be provided by using digital signatures with an appendix. The source of data is assured by signing the concatenation of data and the originator's DN. Then, the message digest, the DN, and the data are transferred to the recipient.

Data origin authentication based on shared secret keys (as MACs) does not allow a distinction to be made between the parties sharing the key, and thus – in contrast to digital signatures – does not provide non-repudiation of origin. If a resolution is required, either an online TTP as a notary authority, or asymmetric techniques may be used.

For further requirements see Chapter 10.2.2.7.

10.2.2.9 Non-Repudiation Service Requirements

10.2.2.9.1 Purpose

Especially inter-institutional communication and co-operation in the *shared care* paradigm sense requires accountability services to provide legal evidence of responsibility of principals involved. Non-repudiation is the property that ensures that the actions of an entity may be traced uniquely to the entity [ISO 7498-2]. Requirements to ensure that relevant information recorded about actions performed by users or processes acting on their behalf so that the consequences of those actions can later be linked to the user in question, and the user held accountable for his actions [EC, 1991]. In the context of communication security, the non-repudiation as part of accountability is most important.

In general, the intended usage of non-repudiation is to *ensure* availability of irrefutable evidence for resolution of any dispute about occurrence or non occurrence of some event or action so that a principal cannot falsely deny being responsible. Evidence establishes accountability regarding a particular event or action. This service *does not* prevent principals to attempt repudiation.

10.2.2.9.2 Service Requirements

A general framework for provision of non-repudiation is given by ISO/IEC10181-4. This Standard Guide defines basic concepts of non-repudiation, identifies classes of generic mechanisms, and describes non-repudiation policies. It neither specifies nor depends on the use of particular mechanisms and algorithms since ISO does not standardise cryptographic algorithms, but rather their procedures for registration (see Chapter 10.2.2.4.4).

Non-repudiation service can be separated into non-repudiation of origin (NRO), non-repudiation of submission (NRS), non-repudiation of transport (NRT), and non-repudiation of delivery (NRD). NRO is a combination of non-repudiation of creation and non-repudiation of sending; NRD must be seen as catenation of non-repudiation of receipt (NRR) and non-repudiation of knowledge. Concerning communication security, only NRO, NRS, NRT, and NRR have to be addressed; all other kinds of non-repudiation *are not* matters of communication security but of application security. If delivery authorities (DA) are involved (i.e. operating in store-and-forward systems (e.g. using SMTP)), the support of NRO, NRS, NRT, and NRR is RECOMMENDED. Otherwise, if no DAs are present (e.g. synchronous transfer using FTP), NRO and NRR SHOULD be provided. If NRR has been successfully proven for the latter scenario, NRS and NRT have also been assured. NRR is initiated by the data originator that wants to have proof of reception against the recipient, and NRO is initiated by the intended recipient that wants to give proof of source against the originator. NRS and NRT are used by the originator to protect against the DAs. It is RECOMMENDED that the kinds of non-repudiation mentioned for the authentication tokens exchanged — the message data transferred (before its confidentiality protection), and the control data transmitted — be used.

Non-repudiation mechanisms that provide evidence MUST be based upon cryptographic techniques using symmetric or asymmetric techniques as described by ISO/IEC13888-2 or ISO/IEC13888-3, respectively. It is REQUIRED that the generalities of ISO/IEC13888-1 be followed. The application of asymmetric techniques, using digital signatures, is RECOMMENDED and REQUIRES the involvement of an offline TTP to guarantee the genuineness of keys (public key certificates management including CRLs and directory servers). Symmetric techniques, using secure envelopes, MAY be applied instead and REQUIRE an on-line TTP for generation and validation of the secure envelopes including resolution of origin preventing fraudulent repudiation. Mechanisms using shared secret keys do not allow a distinction to be made between the parties sharing the key, and thus – in contrast to digital signatures – do not provide NRO. The mechanisms have to provide protocols for the exchange of non-repudiation tokens specific to each kind of non-repudiation. These tokens MAY be stored as information by disputing parties for arbitration.

The non-repudiation service involves the generation, verification and recording of evidence, and the subsequent retrieval and reverification of this evidence in order to resolve disputes. For evidence generation, the TTP MAY act on behalf of a principal involved as a token generation authority (TGA), digital signature generating authority (DSGA), time stamping authority (TSA), notary authority (NA), or monitoring authority (MA). Evidence transfer MAY be carried out by a TTP acting as a delivery authority (DA) or evidence record-keeping authority (ERA). At last, the TTP MAY be in the role of an evidence verification authority (EVA). As NA the TTP SHOULD arbitrate disputes by providing evidence

about the properties of the principals involved and of the data stored or communicated using a generic notarisation token (NT) as defined in ISO/IEC13888-1. In providing an evidence recording service, the TTP SHOULD keep records of operations so that they will be available for the resolution of any disputes that may arise at some time in the future. When a trusted time is required and when the clock provided by the token generating party cannot be trusted, it is necessary to rely on a TTP. As a TSA, the TTP SHOULD provide a time stamping service by generating a generic time stamping token (TST) as specified in ISO/IEC13888-1. The non-repudiation tokens NROT, NRDT, NRST, and NRTT are all derived from a generic non-repudiation token (GNRT) given by ISO/IEC13888-1. For provision of NRR, the NRRT SHOULD be also derived from the GNRT. An overview of the non-repudiation tokens and their usage is given in Figure 10.3 (after ISO/IEC13888-1 and ISO/IEC13888-3).

In general, the token data (TD) of GRNT, TST, and NT SHOULD consist of DNs, a service indication flag, time stamps, and the imprint of the message data m, which can either be the hash code of m or m itself. Return of content MAY be wasteful of network bandwidth and time. Thus, is it RECOMMENDED that only the hash code, and not the whole message, be returned. To be more specific, the GRNT contains the DNs of the message originator, of the message recipient, and of any other authority (as TGA, DA, TSA, and MA) involved. Two time stamps are included stating the date and time when the evidence token was generated, and when the message data was processed (e.g. sent, received, submitted, delivered).

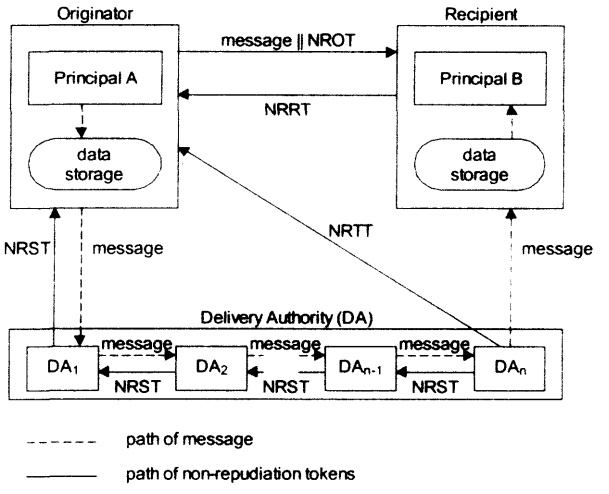


Figure 10.3: Non-Repudiation Tokens and their Usage

When using symmetric cryptographic techniques, an on-line TTP is REQUIRED that takes over the roles of TGA, TSA, NA, and EVA. Additionally, it MAY act as MA, ERA or DA (in-line). The non-repudiation tokens (NRxT) used consist of a cryptographic check value (e.g. MAC) computed on TD by symmetric integrity techniques and the TD itself. Any principal holding that secret key can verify the integrity and origin of TD. For the purpose of generating and verifying evidence, the envelope is constructed and verified by a TTP using a secret key known only to the TTP. The secure envelope MAY also be used for the origin and integrity protected communication between a TTP and any other principal. In that case the envelope is generated and verified with a key known by both the principal and the TTP. For additional assurance, the TSA and TGA SHOULD be different authorities.

For asymmetric mechanisms, an off-line TTP is REQUIRED as TSA and NA. Additionally, it MAY act as MA, ERA or DA (in-line). The Principal that wants to obtain evidence (service requester) MUST generate and verify the evidence on its own. The non-repudiation tokens used consist of a digital signature computed on TD using a private signing key and the TD itself. Any principal having access to the corresponding public key is able to verify the integrity and origin of TD. A chain of public key certificates or identities MAY have to be verified to obtain the necessary assurance. If confidentiality is needed, the result itself (non-repudiation token) MAY be enveloped using the confidentiality service.

To prevent an endless chain of non-repudiation tokens (i.e. giving NRR for the NRRT received) only two transactions SHOULD be carried out between client and server as follows. The client sends an HL7 request message that includes a request for a signed receipt and the server responds by transmitting the HL7 reply message including the receipt, cryptographic enhanced as described above. The client abandons sending a receipt for the server's response in turn.

10.2.2.9.3 Non-Repudiation of Origin

The NRO service provides the recipient of data with proof that protects against any attempt by the sender to falsely deny sending the data. The evidence (non-repudiation of origin token, NROT) is generated by the originator of the message and sent to the intended recipient. The originator sends both the message and the NROT to the recipient. To provide proof, the identities and the integrity of data must be confirmed, and the time stamps must be within the given time window.

10.2.2.9.4 Non-Repudiation of Receipt

The NRR service provides the sender of data with proof that protects against any attempt by the recipient to falsely deny having received the data. The evidence (non-repudiation of receipt token, NRRT) is generated by the recipient of the message and sent to the originator.

The recipient sends both the reply message (if any) and the NRRT to the originator. To provide proof, the identities and the integrity of data must be confirmed, and the time stamps must be within the given time window.

10.2.2.9.5 Non-Repudiation of Submission

The NRS service provides the sender of data, which may be another DA, with proof that protects against any attempt by the DA to falsely deny having accepted the data for transmission. The DA *does not care* what the content of the message is. The originator, or a preceding DA has sent a message to the next DA that receives this message and sends the NRS token to the originator or the preceding DA, establishing a chain of intermediate NRST tokens providing chained NRS.

To provide proof, the identities and the integrity of data must be confirmed, and the time stamps must be within the given time window.

10.2.2.9.6 Non-Repudiation of Transport

The NRT service provides the sender of data with proof that protects against any attempt by the DA to falsely deny having delivered the data to the intended recipient. The DA *does not care* what the content of the message is and *cannot guarantee* that the message is duly received by the recipient. The evidence, in the form of a non-repudiation of transport token or NRTT, is generated by the DA that is delivering the message to the intended recipient (the last DA in the chain of DAs) and sent back to the originator. To provide proof, the identities and the integrity of data must be confirmed, and the time stamps must be within the given time window.

10.2.3 Merging secured Data Elements to EDI Messages

Communication security includes the assembling and merging of already secured data elements to complete EDI messages. The protection of data elements (i.e. application of security services, and their type of storage (e.g. using a database)) is part of application security (data element security, data base security) and is not considered here.

The merging process MUST build a complete EDI message gathering certain secured data elements by retrieving them from the storage device (e.g. database). The way in which the message is constructed (structure) and what element delimiters are used determines the kind of the EDI message generated. Possible kinds of EDI message are HL7, EDIFACT, xDT and others.

If the data element is integrity protected, the integrity code (digital signature or MAC) MUST be character converted (e.g. using Base64-encoding) and canonicalised to prevent loss of characters resulting in invalidation of integrity. Furthermore, the DN is REQUIRED for origin authentication. It MUST be possible to separate these data fields (i.e. original data element, integrity code as well as DN) from each other for further processing (e.g. integrity verification, data base storage). This can be achieved by introducing special delimiters. The approach of using only one new delimiter requires some intelligence of the EDI interface, namely counting the delimiter, to recognise that the first entry separated by the delimiter contains the DN and the second entry separated by the same delimiter contains the integrity code. For that reason, different delimiters may be more applicable, limiting necessary interface modifications for parsing and evaluation.

When the data element has confidentiality protection, it MUST be character converted and canonicalised as well to provide interoperability.

For insertion of protected data fields into EDI messages and to distinguish the different levels of protection — plain, signed-only, encrypted-only and signed-and-encrypted — proper means of identification MUST be provided. Again, appropriate delimiters SHOULD be applied.

The protected data items have to be transmitted (forwarded, processed) by EDI applications together with the original data element as part of the message, regardless of any communication security mechanisms that could be used as envelope.

Construction of EDIFACT messages is specified by ISO9735. Security as specified by ISO9735-5 (authenticity, integrity, and NRO), ISO9735-6 (message authentication and acknowledgement AUTACK, i.e. providing NRO and NRR), ISO9735-7 (confidentiality), ISO9735-9 (key management), and ISO9735-10 (security for interactive EDIFACT) is aimed at EDIFACT structures (segment levels). Data element security for HL7 is currently under construction. Until now, there is no security standard for xDT structured data available.

10.3 Standard Guide for Implementing EDI (HL7) Communication Security

10.3.1 Scope

This Standard Guide gives the framework and implementation details for implementing secure end-to-end EDI communication focusing on HL7. It is based on the "Standard Guide for Specifying EDI (HL7) Communication Security" and addresses system implementers.

Starting with an introduction to security services and general requirements for applications and especially for communication security, in Chapter 6, the fundamental security services needed as strong mutual authentication are described as well as securing information exchange by securing control data and message data, mentioning possible security attacks,

security requirements and proposed implementations of solutions. In that context, different exchange protocols are considered. Regarding accountability, different non-repudiations services are discussed in detail.

The principles are illustrated in detail for a Secure File Transfer Protocol (SFTP), implemented in the authors' environment.

10.3.2 Basics

Establishing secure communication of EDI messages requires the selection and implementation of appropriate security services like authentication (user, application and system), integrity, confidentiality, and accountability (in the sense of non-repudiation of origin and receipt), as described and defined in the "Standard Guide for Specifying EDI (HL7) Communication Security" [HL7CommSec]. Communication security is related to the cryptographic enhancement of the whole message, such as message signing and message encryption, regardless of its internal structure.

Following the recommendations of the "Standard Guide for EDI (HL7) Communication Security" the solution for secure communication of EDI messages presented in this standard guide is based upon public key cryptography within a proper security infrastructure (public key infrastructure, PKI).

In the first section, the fundamentals of the solution are described independently of the communication protocol used. For each security service selected, the specific structure and contents of tokens exchanged between client and server is described. This includes all security services proposed in the "Standard Guide for Specifying EDI (HL7) Communication Security" as strong mutual authentication, integrity and confidentiality assurances, as well as non-repudiation of origin and receipt.

Subsequently, the communication protocol for token exchange is presented in detail, serving as the most convenient mechanism next to the paradigm of EDI communication in client/server architectures. This protocol is called the secure file transfer protocol (SFTP) and is a security enhanced version of the unsecured RFC0959 compliant FTP.

10.3.3 Security Services and General Realisation

After giving some basic information about the public key infrastructure and the notation used in this standard guide, this section selects the security services needed for the control and data connection regarding the "Standard Guide for Specifying EDI (HL7) Communication Security". Then, for each service, the general realisation is given independently of the communication protocol used. Possible attacks and countermeasures are considered, and the resulting structure and contents of the tokens exchanged are presented.

10.3.3.1 Fundamentals and Notations

This approach is based upon public key cryptography using asymmetric security techniques and symmetric security techniques as well. The latter is only taken for bulk data encryption within hybrid encryption employing a symmetric session key. Trust is established by a public key infrastructure (PKI) using trusted public keys certified by a certification authority (CA). For this purpose, X.509 certificates that are stored and managed in X.500 directories are applied.

Different key pairs MUST be used for authentication, digital signature generation/verification and encryption/decryption to avoid security compromise by possibly adaptive chosen-text attacks where the intruder chooses challenges to extract information about the key. This MAY be possible since the private key is used to compute the response and, thus, may reveal partial information. Hence, there are cryptographic needs for key separa-

tion requiring the use of one key for each purpose. For key separation, the notations given in Table 10.4 are used in the following.

The main symbol indicates the type of asymmetric key (*PrK* for private key and *PK* for public key), whereas the upper index denotes the key usage (*auth* for authentication, *digSig* for digital signature generation/verification, and *crypt* for encryption/decryption) and the lower index identifies the owner of the key (*client* for client, *server* for server and *ca* for the CA).

Table 10.4: Key Separation by Key Usage

Notation	Usage
PrK^{auth}	Private key for authentication (generation of digital signature)
PrK^{digSig}	Private key for integrity service (generation of digital signature)
PrK^{crypt}	Private key for decryption
PK^{auth}	Public key for authentication (verification of digital signature)
PK^{digSig}	Public key for integrity service (verification of digital signature)
PK^{crypt}	Public key for encryption

Strong security measures MUST be achieved by using strong cryptographic mechanisms and public key certificates following X.509. The certificates MUST be verified before usage every time. For verification, the public key certificate of the CA for digital signature verification (PK_{CA}^{digSig}) is needed. This certificate MUST be checked itself before usage. Certificate verification MAY involve directory or local cache access performed prior to the authentication exchange.

For *token formatting*, the tag-length-value (TLV) format MUST be applied. Each token field is preceded by a tag-byte specifying the type of data and a length-word (little-endian order: first is low byte) that determines the amount of data that follows, as shown in Table 10.5. The concatenation of tokens is expressed by "||".

Table 10.5: Tag-Length-Value Format of Tokens

Token Offset	Purpose
0x0000	TAG-byte (identifies the type of data)
0x0001	LEN-byte (low-byte data amount)
0x0002	LEN-byte (high-byte data amount)
0x0003 and following	DATA-bytes (data)

As XML becomes increasingly important, the TLV encoding format MAY be easily replaced by defining new XML attributes. The TLV-TAG-byte is transformed to an XML-tag, the values follow immediately, and the length is given implicitly by the XML-ending-tag. All authentication and control connection messages are then XML-messages. In addition, XML-messages can be sent over the data connection, transforming e.g. S/MIME or PKCS#7 into XML. The client and the server must have an XML generator and parser as well.

10.3.3.2 Strong Mutual Authentication

HL7 realises event-driven exchange of messages between healthcare applications. Depending on specific circumstances or protocols used, the communicating principals may also include the human user of an information system. As a basic requirement, the communicating principals MUST be authenticated mutually. Because the solution implemented is intended to be open and flexible and also allow inter-protocol communication, different use cases must be reflected. In that context, the solution has to be fitted into the security environment, e.g., the European security infrastructure.

Following the "Standard Guide for EDI (HL7) Communication Security," asymmetric techniques are applied for strong authentication. Before mutual trust between client and server can be established (the client is authenticated to the server and vice versa), the human user must authenticate oneself to a cryptographic module of the local end system (*user authentication*), which then acts as an initiator (client) and performs the digital signature generation and verification on behalf of the human user towards the responder (server), carrying out *system authentication*.

As recommended in the "Standard Guide for EDI (HL7) Communication Security," human user authentication SHOULD be carried out by ISO/IEC7816 compliant chipcards (e.g. Health Professional Cards = HPC) in combination with a PIN code. During a communication session, the chipcard needs to be kept inserted in the chipcard terminal for timed chipcard request. When removing the chipcard, the application MUST inhibit further operations and only continue to work if the chipcard is inserted again and the subsequent user authentication is performed successfully.

The strong authentication of an initiator to the responder depends on the successful verification of the initiator's digital signature binding with its key pair and also on a successful verification of the initiator's digital signature (signing means showing the possession of the secret key) on a random number challenge generated by the responder. Accordingly, for mutual authentication the successful authentication of the responder to an initiator is checked. The binding of a principal's unique identifier (i.e. the distinguished name (DN)) with its key pair is essential for proving the authenticity of its identity and must be checked prior to any authentication exchange. This is achieved by user authentication following verification of X.509 public key certificates retrieved from a X.500 directory.

Protocols for strong mutual authentication using asymmetrical security techniques can be found in [ISO/IEC 9798-3], [FIPS196] and [X.509] Chapter 10. There are some differences between these three sources concerning the authentication data structure, particularly regarding what token fields (such as digital signature, random numbers, time stamps and DNs) are recommended or optional in each step and what fields are covered by the digital signature. The order of authentication (first, the client is authenticated to the server, and afterwards the server is authenticated to the client in turn) remains the same, of course. Mutual authentication is performed by a three-way challenge-response-protocol for security and efficiency reasons, limiting the amount of tokens exchanged.

The authentication procedures defined in [X.509] are intended to be used between directory user agents (DUAs), but mainly follow the other specifications. Random numbers are used in combination with time stamps.

[ISO/IEC 9798-3] serves as a basis for the authentication protocols defined in [FIPS196] and specifies different protocols addressing both unilateral and mutual entity authentication, which make usage of public key cryptography algorithms.

In [FIPS196] only one protocol for each unilateral and mutual authentication has been selected from [ISO/IEC 9798-3] and certain authentication token fields and protocol steps are described in greater detail than in the ISO specification. Furthermore, [FIPS196] is less strict, allowing the arbitrary ordering of token fields. The appendices A through D of this

specification contain several optional methods and sets of rules for formatting and encoding authentication information (ASN.1 Notation and CER/DER encoding, Simple Public Key GSS-API Mechanism (SPKM), formatting based on ANSI X9.26-1990 and Base64 encoding) helping to promote the interoperability of various implementations of the authentication protocols defined. These appendices are provided for informational purposes only and are not part of the standard. Formatting and encoding is left to the discretion of the implementers. Moreover, to avoid the use of synchronised clocks to verify the timeliness of authentication tokens, authentication exchanges using time stamps were not chosen for [FIPS196]. Beyond that, sequence numbers have not been chosen, due to their requirement of maintaining sequence number log tables. Instead, random number challenges are used for both time stamps and sequence numbers. Finally, biometric authentication techniques are not included, but discussed in [FIPS196].

For the protocol presented in the following pages, all standards and recommendations mentioned above are combined and enhanced, gaining as much robustness and security as possible.

10.3.3.2.1 Possible Attacks and Countermeasures

For carrying out strong mutual authentication using asymmetric techniques, authentication tokens have to be exchanged by a challenge-response protocol. In general, these tokens MUST be completely integrity-protected to detect alteration of any kind. Furthermore, data origin authentication as well as non-repudiation of origin and receipt MUST be offered by the authentication protocol.

To cover all the security services mentioned and to detect or prevent attacks on the protocol, the tokens exchanged MUST contain certain fields like random numbers, time stamps, DNs and others. In the following section, possible attacks are listed giving appropriate countermeasures resulting in mandatory token fields. Forms of attack include impersonation/masquerading, token manipulation, replay, relay/forced delay, interleaving, reflection, key-related and implementation-related.

Impersonation or *masquerading* is the representation or implication of a false identity (i.e. assuming the identity of one of the legitimate parties in the network). This threat is eliminated by the authentication service itself assuring that the identity claimed is authentic. This is achieved by binding the identity (DN) to the token sent using the integrity service to provide data origin authentication. Authenticity of a public key is given by a CA-created certificate that binds identity and public key together. Proving this authenticity means verifying the certificate and possibly a chain of certificates to establish a hierarchy of trust. For data origin authentication and non-repudiation of origin, the distinguished name (DN) and physical location of the sender MUST be included. The physical location is specified by IP address and network adapter hardware address (MAC).

Token manipulation is addressed by integrity protection of the complete token for all exchanges performed. *Continuity of authentication* MUST be assured by binding the authentication service and the integrity service for further token exchanges (see control data) so that no intruder can cut in or take over after the authentication has been completed. Moreover, system authentication MUST be performed at the beginning and throughout each session (timed authentication). For integrity protection, digital signatures MUST be applied. For each signed message, the signature MUST be included for verification purposes.

Replay attacks MUST be addressed by including pre-assigned random numbers generated by the counterpart that are checked for equality when the counterpart receives the reply. Moreover, a chaining of random numbers MUST be carried out to verify if the number received by the counterpart in the last message is the same as that which has come with the current message (see Figure 10.4). As the same random number challenge MUST never be

issued, or accepted, twice by the same machine (client, server), a random number log-table MUST be maintained. Furthermore, a sequence number MUST be applied for each token. This number must be recorded as well to prevent duplications. To reduce logging of the unique numbers (random number, sequence number) to a certain time, window time stamps SHOULD be used as continuous transaction numbers. The time stamps of token generation and expiration time are included in the token applying UTC time (Universal Time Coordinated). Thus, a secure time service that offers synchronous clocks is needed. Without a secure time server, the time difference of the stamps can only be verified using local time. Different report logs have to be maintained by the server for each client.

A token identifier MUST be included to recognise and distinguish the different kinds of tokens exchanged for authentication (request, data1, data2, and data3).

To eliminate *relay attacks* where the intruder acts as a wire (i.e. forced delay and intruder-in-the-middle), additional measures MUST be applied. First, the continuity of authentication is assured as described above. Furthermore, the DN, IP address and the MAC (determining the physical location) of the recipient are included (dedicated challenge). Then, the time stamps included prohibit delays that are longer than the time window given. Short time outs are applied for the communication protocol model (i.e. the temporal distance of commands and replies as well as of each command in relation to the next in a chain of commands is checked using short time windows). Lastly, the authentication tokens indicate the role of the issuer (initiator, responder). A state indicator (authentication invitation, authentication request) marks whether authentication is being invited by the verifier or requested by the initiator. This is needed if both parties can start the authentication procedure, which is generally the case for mutual authentication protocols. This additional information may not be included if the entity that initiates interaction is either always the claimant or always the verifier.

Attacks on the implementation as *interleaving*[37] or *reflection*[38] MUST be addressed also. Uni-directional keys, where each principal has a unique signing key must be used. The identifiers of the target party are included, and tag-length-value encoding (TLV, see Table 10.5) is applied for field identification. TLV encoding permits randomisation of the order of fields inside tokens and prevents message symmetries. Furthermore, TLV protects against an inconsistent view of token fields giving a unique standardisation of all possible contents. The fields are distinguishable from each other independent of their token position. Thus, the implementation is more efficient due to the unambiguousness gained.

Token IDs and sequence numbers protect against an inconsistent view of tokens giving each message a unique tag of position within a stream of messages.

Confidentiality MUST NOT be applied, otherwise *key attacks* are possible. Since the control data is a well-known and often a small set of commands resulting in short tokens, key attacks like forward search[39] or dictionary[40] may be successful. Moreover, applying unnecessary security services may result in security flaws due to possible interaction between security mechanisms. When used in isolation, security mechanisms provide an acceptable

[37] The attack concerns the selective combination of information from one or more previously or simultaneously ongoing protocol sessions, including possible origination of one or more protocol sessions by an intruder itself.

[38] This concerns an interleaving attack involving sending information from an ongoing protocol session back to the originator of the information.

[39] Applying forward search means that the intruder takes all 2^n possible entries of a token field and encrypts them using the public key of the original recipient and compares each of the 2^n ciphertexts with the value in the transaction actually encrypted (n denotes the amount of bits per token).

[40] When performing dictionary attacks, the intruder encrypts all entries of a dictionary (e.g. the list of communication protocol commands) with the public key of the original recipient and compares the result to the transmitted value.

level of security, but may become more vulnerable when used in combination with other mechanisms (see [ISO/IEC 10181-1] page 15).

NRR SHOULD be provided by including the hash value of the previously received token in the new message to be sent.

After the authentication (user and system) has been successfully performed, authorisation and audit based upon the user's identity MAY be carried out. The identity involved is obtained from the authentication tokens (as the field containing the DN).

10.3.3.2.2 Proposed Implementation

In addition to the fields needed in the authentication tokens as given in the "Standard Guide for EDI (HL7) Communication Security" and explained above, all tokens MUST be Base64-encoded and canonicalised afterwards before transmission for *system interoperability preventing loss of data bits* in environments not capable of full binary transport. Not applying these conversions may result in invalidation of the digital signature.

The resulting protocol scheme is shown in Figure 10.4 and the tokens exchanged (Authentication Request, AuthData1, AuthData2 and AuthData3) are presented within the sequence of the protocol. Each token field is TLV-formatted which is not shown explicitly.

It may be difficult for the client to obtain the MAC address or the DN of the destination server in order to build the authentication request. A possible solution is to store them in the client environment statically, but handling this becomes difficult when changing the Ethernet card or the name of the server. In such cases, these values MAY be omitted from the authentication request token (and the server does not check them). In this case, the client obtains these parameters in AuthData1 for further usage. However, the client SHOULD know the IP address of the server in order to establish a direct network connection. This address MUST be included in the request and checked by the server.

Among other things, the TLV enables unique identification of the values and free ordering. However, if a signature is calculated over some TLV-encoded items, the ordering MUST be the same for the verification process. Otherwise, the digital signature becomes invalid. To ensure equality of the encoded items on both the client and the server, the signature MUST be included either before or after all other data items signalling that, for verification purposes, the signature has to be calculated over all data following or all data before respectively.

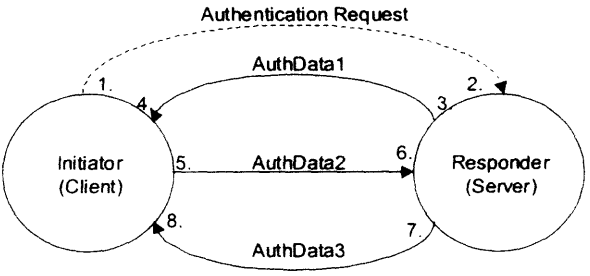


Figure 10.4: Strong Mutual Three-Way Authentication

1. At first, the client initiates system authentication by sending an authentication request token to the server:
   a. Generate:

AuthReq' =
$SeqNo_1 \| TokenID_1 \| IP_{client} \| MAC_{client} \| DN_{client} \| IP_{server} \| MAC_{server} \| DN_{server} \|$
$TS_{gen1} \| TS_{exp1} \|$ Role-Initiator $\|$ State-Request $\| R_{client1} \|$ Auth-Mechanism.

b. Calculate the digital signature over all fields:
$DS_1 = PrK^{auth}_{client}(AuthReq')$.

c. Concatenate the digital signature and the generated token:
AuthReq'' = AuthReq' $\| DS_1$.

d. Perform Base64-encoding and canonicalisation afterwards:
$AuthReq = (AuthReq'')_{Base64, Canon}$.

e. Send the token to the server.

2. On receipt of AuthReq, the server performs the following operations:
   a. Apply Base64-decoding.
   b. Check if all necessary fields are included (using the TAG-byte).
   c. Verify the certificate of PK^{digSig}_{CA}, use PK^{digSig}_{CA} to verify PK^{auth}_{client} and take PK^{auth}_{client} to check the digital signature of the token received.
   d. Check if the TLV-format is correct, i.e. if the values of length are correct matching the length of data supplied.
   e. Check token type ($TokenID_1$), sequence number ($SeqNo_1$), time stamps (TS_{gen1}, TS_{exp1}) as well as role and state indicator (Role-Initiator, State-Request) for validity.
   f. Verify the identifiers of sender and recipient (DNs, IP addresses and MACs).
   g. Evaluate the authentication request command (AUTH-Command).

3. Then, the server builds AuthData1:
   a. Generate:
   AuthData1' =
   $SeqNo_2 \| TokenID_2 \| IP_{client} \| MAC_{client} \| DN_{client} \| IP_{server} \| MAC_{server} \| DN_{server} \|$
   $TS_{gen2} \| TS_{exp2} \|$ Role-Responder $\|$ State-Request $\| R_{client1}' \| R_{server1} \|$ Hash-$Value_{AuthReq'}$.
   b. Calculate the digital signature over all fields:
   $DS_2 = PrK^{auth}_{server}(AuthData1')$
   c. Concatenate the digital signature and the generated token:
   AuthData1'' = AuthData1' $\| DS_2$.
   d. Perform Base64-encoding and canonicalisation afterwards:
   $AuthData1 = (AuthData1'')_{Base64, Canon}$.
   e. Send the token to the client.

4. On receipt of AuthData1, the client performs the following operations:
   a. Apply Base64-decoding.
   b. Check if all necessary fields are included (using the TAG-byte).
   c. Verify the certificate of PK^{digSig}_{CA}, use PK^{digSig}_{CA} to verify PK^{auth}_{server} and take PK^{auth}_{server} to check the digital signature of the token received.
   d. Check if the TLV-format is correct, i.e. if the values of length are correct matching the length of data supplied.
   e. Check token type ($TokenID_2$), sequence number ($SeqNo_2$), time stamps (TS_{gen2}, TS_{exp2}) as well as role and state indicator (Role-Responder, State-Request) for validity.
   f. Verify the identifiers of sender and recipient (DNs, IP addresses and MACs).
   g. Check if $R_{client1} = R_{client1}'$.

h. For NRR, check $\text{HashValue}_{\text{AuthReq}'}$.

5. Now, the client builds AuthData2:
   a. Generate:
      AuthData2' =
      $\text{SeqNo}_3 \parallel \text{TokenID}_3 \parallel \text{IP}_{\text{client}} \parallel \text{MAC}_{\text{client}} \parallel \text{DN}_{\text{client}} \parallel \text{IP}_{\text{server}} \parallel \text{MAC}_{\text{server}} \parallel \text{DN}_{\text{server}} \parallel \text{TS}_{\text{gen3}} \parallel \text{TS}_{\text{exp3}} \parallel \text{Role-Initiator} \parallel \text{State-Request} \parallel \text{R}_{\text{client1}}'' \parallel \text{R}_{\text{server1}}' \parallel \text{R}_{\text{client2}} \parallel \text{HashValue}_{\text{AuthData1}'}$.
   b. Calculate the digital signature over all fields:
      $\text{DS}_3 = \text{PrK}^{\text{auth}}_{\text{client}}(\text{AuthData2}')$
   c. Concatenate the digital signature and the generated token:
      $\text{AuthData2}'' = \text{AuthData2}' \parallel \text{DS}_3$.
   d. Perform Base64-encoding and canonicalisation afterwards:
      $\text{AuthData2} = (\text{AuthData2}'')_{\text{Base64, Canon}}$.
   e. Send the token to the server.

6. On receipt of AuthData2, the server performs the following operations:
   a. Apply Base64-decoding.
   b. Check if all necessary fields are included (using the TAG-byte).
   c. Verify the certificate of $\text{PK}^{\text{digSig}}_{\text{CA}}$, use $\text{PK}^{\text{digSig}}_{\text{CA}}$ to verify $\text{PK}^{\text{auth}}_{\text{client}}$ and take $\text{PK}^{\text{auth}}_{\text{client}}$ to check the digital signature of the token received.
   d. Check if the TLV-format is correct, i.e. if the values of length correctly match the length of data supplied.
   e. Check token type (TokenID_3), sequence number (SeqNo_3), time stamps (TS_{gen3}, TS_{exp3}) as well as role and state indicator (Role-Initiator, State-Request) for validity.
   f. Verify the identifiers of sender and recipient (DNs, IP addresses and MACs).
   g. Check if $\text{R}_{\text{client1}} = \text{R}_{\text{client1}}''$ and $\text{R}_{\text{server1}} = \text{R}_{\text{server1}}'$.
   h. For NRR, check $\text{HashValue}_{\text{AuthData1}'}$.

   After successfully processing step h., the client is authenticated to the server.

7. Then, the server builds AuthData3:
   a. Generate:
      AuthData3' =
      $\text{SeqNo}_4 \parallel \text{TokenID}_4 \parallel \text{IP}_{\text{client}} \parallel \text{MAC}_{\text{client}} \parallel \text{DN}_{\text{client}} \parallel \text{IP}_{\text{server}} \parallel \text{MAC}_{\text{server}} \parallel \text{DN}_{\text{server}} \parallel \text{TS}_{\text{gen4}} \parallel \text{TS}_{\text{exp4}} \parallel \text{Role-Responder} \parallel \text{State-Request} \parallel \text{R}_{\text{client2}}' \parallel \text{R}_{\text{server1}}'' \parallel \text{R}_{\text{server2}} \parallel \text{HashValue}_{\text{AuthData2}'}$.
   b. Calculate the digital signature over all fields:
      $\text{DS}_4 = \text{PrK}^{\text{auth}}_{\text{server}}(\text{AuthData3}')$
   c. Concatenate the digital signature and the generated token:
      $\text{AuthData3}'' = \text{AuthData3}' \parallel \text{DS}_4$.
   d. Perform Base64-encoding and canonicalisation afterwards:
      $\text{AuthData3} = (\text{AuthData3}'')_{\text{Base64, Canon}}$.
   e. Send the token to the client.

8. On receipt of AuthData3, the client performs the following operations:
   a. Apply Base64-decoding.
   b. Check if all necessary fields are included (using the TAG-byte).
   c. Verify the certificate of $\text{PK}^{\text{digSig}}_{\text{CA}}$, use $\text{PK}^{\text{digSig}}_{\text{CA}}$ to verify $\text{PK}^{\text{auth}}_{\text{server}}$ and take $\text{PK}^{\text{auth}}_{\text{server}}$ to check the digital signature of the token received.

d. Check if the TLV-format is correct, i.e. if the values of length correctly match the length of data supplied.
e. Check token type ($TokenID_4$), sequence number ($SeqNo_4$), time stamps (TS_{gen4}, TS_{exp4}) as well as role and state indicator (Role-Responder, State-Request) for validity.
f. Verify the identifiers of sender and recipient (DNs, IP addresses and MACs).
g. Check if $R_{client2} = R_{client2}'$ and $R_{server1} = R_{server1}''$.
h. For NRR, check $HashValue_{AuthData2'}$.

After successfully processing step h., the server is authenticated to the client.

An overview of the authentication tokens exchanged and the verification carried out for the random numbers is shown in Figure 10.5.

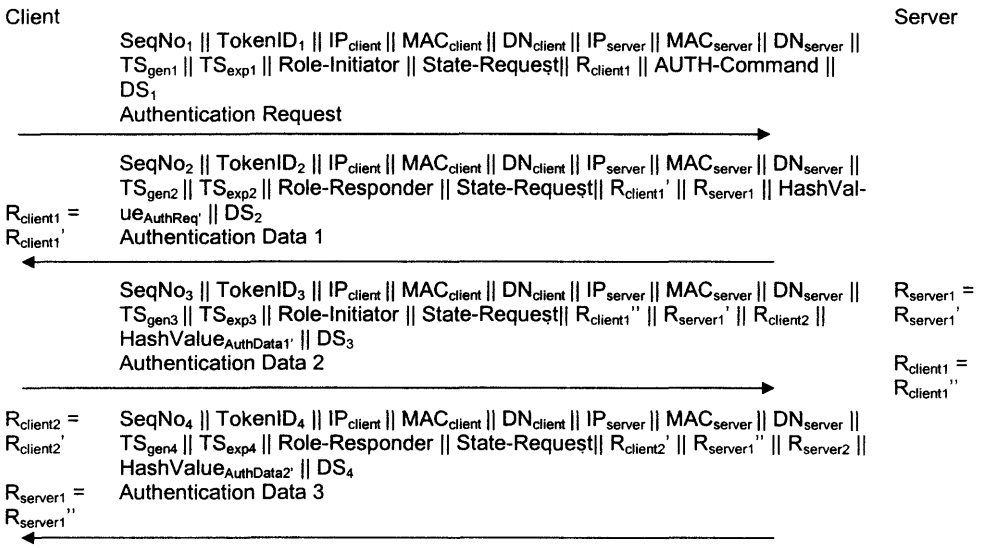


Figure 10.5: Overview of the Authentication Tokens Exchanged

10.3.3.3 Securing the Control Data

When user and system authentication have been performed successfully, the control connection MUST be integrity protected as required by the "Standard Guide for EDI (HL7) Communication Security". Furthermore, data origin authentication, non-repudiation of origin (NRO), and non-repudiation of receipt (NRR) SHOULD be provided.

10.3.3.3.1 Possible Attacks and Countermeasures

Integrity protection, using digital signatures, MUST be applied during the whole session over all fields contained in a message of control data (token). This allows *detected token manipulation* and assures the *continuity of authentication* binding the authentication service and the integrity service so that no intruder can cut in or take over after the authentication has been completed. For each signed message, the signature must be included for verification purposes.

Replay attacks MUST be addressed by including pre-signed random numbers generated by the counterpart that are checked for equality when the counterpart receives the reply. Moreover, a chaining of random numbers MUST be carried out to verify if the number received

by the counterpart in the last message is the same as that which comes with the current message (see Figure 10.5). Furthermore, a sequence number MUST be applied for each token. To reduce logging of the unique numbers (random number, sequence number) to a certain time window, time stamps SHOULD be used as continuous transaction numbers. The time stamps of token generation and expiration time are included in the token applying UTC time.

A token identifier MUST be included to distinguish control data that is sent from the client to the server, which contain commands, from control data that is transmitted from the server to the client, which contain reply codes.

For data origin authentication and non-repudiation of origin, the sender MUST include his distinguished name (DN), IP address and network adapter hardware address (MAC).

To eliminate *relay attacks* where the intruder acts as a wire (i.e. forced delay and intruder-in-the-middle) additional measures MUST be applied. First, the continuity of authentication is assured as described above. Furthermore, the DN, IP address and the MAC (determining the physical location) of the recipient are included. Then, the time stamps included prohibit delays that are longer than the time window given. At last, short time outs are applied for the communication protocol model (i.e. the temporal distance of commands to replies as well as of a command to the next in a chain of commands is checked using short time windows.)

Attacks on the implementation as interleaving or reflection MUST be addressed also. Unidirectional keys are used (each principal has a unique signing key), the identifiers of the target party are included, and tag-length-value encoding (TLV) is applied for field identification. Token IDs and sequence numbers protect against an inconsistent view of tokens giving each message a unique tag of position within a stream of messages.

Confidentiality MUST NOT be applied, otherwise *key attacks* are possible. Since the control data is a well-known and often a small set of commands resulting in short tokens, key attacks like forward search or dictionary may be successful. Moreover, applying unnecessary security services may result in security flaws due to possible interaction between security mechanisms. When used in isolation, security mechanisms provide an acceptable level of security, but may become more vulnerable when used in combination with other mechanisms (see [ISO/IEC 10181-1], page 15).

NRR SHOULD be provided by including the hash value of the previously received token in the new message to be sent.

10.3.3.3.2 Proposed Implementation

In addition to the fields needed in the authentication tokens as given in the Standard Guide and explained above, all tokens MUST be Base64-encoded and then canonicalised before transmission for *system interoperability preventing loss of data bits* in environments not capable of full binary transport. Not applying these conversions may result in invalidation of the digital signature.

The resulting token contents for the control data connection are given in the following. First, the generation and verification of the tokens are described in detail. Then, a general overview of the token exchanged within the communication protocol is shown regarding the continuity of authentication by resuming the authentication protocol given below (see Figure 10.6). Each token field is TLV-formatted, which is not shown explicitly.

Besides others, the TLV enables unique identification of the values and free ordering. However, if a signature is calculated over some TLV-encoded items, the ordering MUST be the same for the verification process. Otherwise, the digital signature becomes invalid. To ensure equality of the encoded items on both the client and the server, the signature MUST be included either before or after all other data items, signalling that for verification pur-

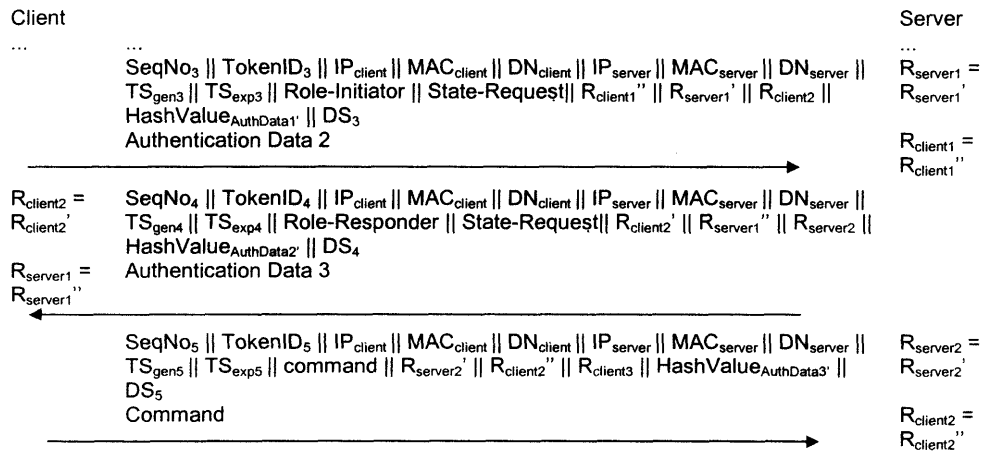
poses, the signature has to be calculated over all data following or all data before, respectively.

In general, the token generation process (of command or reply codes) for every control data token looks like the following:

a. Generate:
   $Token'_n$ =
   $SeqNo \parallel TokenID_m \parallel IP_{client} \parallel MAC_{client} \parallel DN_{client} \parallel IP_{server} \parallel MAC_{server} \parallel DN_{server} \parallel TS_{gen} \parallel TS_{exp} \parallel [command \vee replyCode] \parallel randomNumbers \parallel HashValue_{Token'(n-1)})$.
b. Calculate the digital signature over all fields:
   $DS_n = [\, PrK_{client}^{digSig}(Token'_n) \vee PrK_{server}^{digSig}(Token'_n)]$.
c. Concatenate the digital signature and the generated token:
   $Token''_n = Token'_n \parallel DS_n$.
d. Perform Base64-encoding and canonicalisation afterwards:
   $Token_n = (Token''_n)_{Base64,\ Canon}$.
e. Send the token to the [server $\vee$ client].

Basically, the following steps are performed on receipt of the token for verification:

a. Apply Base64-decoding.
b. Check if all necessary fields are included (using the TAG-byte).
c. Verify the certificate of PK_{CA}^{digSig}, use PK_{CA}^{digSig} to verify $[\, PK_{client}^{digSig} \vee PK_{server}^{digSig}\,]$ and use $[\, PK_{client}^{digSig} \vee PK_{server}^{digSig}\,]$ to check the digital signature DS_n of the token.
d. Check if the TLV-format is correct (i.e. if the value of length correctly matches the length of data supplied.)
e. Check the token ID (TokenID), sequence number (SeqNo) and time stamps (TS_{gen}, TS_{exp}).
f. Verify the identifiers of sender and recipient (DNs, IP addresses and MACs).
g. Check the random numbers for equality (see
h. Figure 10.6).
i. For NRR, check $HashValue_{Token'(n-1)}$.
j. Evaluate the [command $\vee$ reply code].



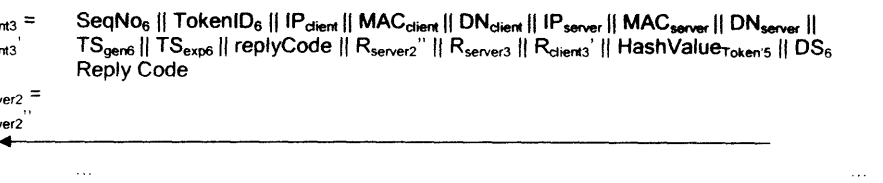


Figure 10.6: Control Data Tokens Exchanged Regarding Continuity of Authentication

10.3.3.4 Securing the Message Data

As described in the "Standard Guide for EDI (HL7) Communication Security," the communication protocol implementation MUST provide integrity protection. Furthermore, confidentiality and non-repudiation services (of origin and receipt) SHOULD be offered. The protocol has to be usable in any desired environment (such as HL7) for secure delivery of data files containing different types of data such as HL7, X12, xDT, XML messages or binary data; data type independence. The message data MUST be character converted and canonicalised to prevent loss of data bits. Furthermore, for correct handling and feature negotiation, a cryptographic syntax MUST be used for encapsulation. Thus, to satisfy all these requirements, MIME-object security MUST be applied.

Security Multiparts for MIME [RFC1847], S/MIME version 2 [SMIME2], S/MIME version 3 [SMIME3], MOSS [RFC1848] or PGP/MIME [RFC2015] are appropriate for this purpose. Informational examples for applying Security Multiparts for MIME and S/MIME version 2 are given in Annex B and Annex C, respectively. Being *independent of the cryptographic protocol syntax*, any other desired cryptographic syntax can be added when offering the featured needed.

Each cryptographic protocol SHOULD be used in three different operation modes (besides plain text) according to the local security policy: signed-only, encrypted-only or signed-and-encrypted. These modes MUST be realised by applying MIME-object nesting. For bulk encryption (content encryption) a strong symmetric session key (having at least 112 significant key bits) MUST be used and MUST be secured by strong asymmetric techniques (preferably by RSA with 1024 bits and above) for transport (hybrid encryption). The session key algorithm MUST be selectable and a new key SHOULD be calculated for each message data transport. Switching between the cryptographic protocols and their operation modes SHOULD be performed easily by the human user.

For *transmission of large files*, data compression or delivery of raw cryptographic objects MAY be applied. For Security Multiparts for MIME and S/MIME these raw objects are PKCS#7-based as PKCS#7-objects [RFC2315] or CMS-objects [SMIME3]. PGP/MIME is based upon PGP-objects, whereas MOSS is not bound to a specific syntax.

Compression of EDI messages MUST be done before encryption, *after* applying the digital signature if needed ([MIME-SECURE], Chapter 5.4.1). In general, EDI messages compress well, since there is much repetitive data in most of the messages. Applying compression before encryption strengthens cryptographic security since repetitious strings are reduced due to their redundancy. The MIME standards [MIME] do not define any content encoding concerning compression, but allow the definition of additional content fields (see Chapter 9 of RFC2045). As presented in [MIME-SECURE], an additional content field "`Content-Encoding:`" (following RFC2068 Chapter 3.5 and 14.12 for HTTP1.1) may be inserted to convey compression information. If `gzip` (see RFC1952) is used, this looks like" `Content-Encoding: gzip`".

Transport of raw cryptographic objects (as PKCS#7, CMS or PGP) can be applied to avoid the cryptographic syntax overhead of MIME security as Base64-encoding. MIME headers

and trailers. Raw objects of this kind MUST NOT be used for transport of EDI messages, because neither canonicalisation nor Base64-encoding is performed. Without MIME headers, no content handling and feature negotiation can be performed. Furthermore, NRR can be only provided for CMS-objects in combination with the Enhanced Security Services (ESS, [SMIME3]). Otherwise, there is no NRR support available for these raw objects. For NRR-related issues see Chapter 10.3.3.4.3.

10.3.3.4.1 Encapsulating EDI-messages in MIME

Before delivery of EDI messages using MIME security, the message MUST be Base64-encoded to prevent loss or manipulation of certain EDI characters (as the HL7 segment terminator) leading to invalidation of the digital signature. Furthermore, the message MUST be inserted into a MIME body for delivery that must also be canonicalised. On receipt, the MIME body MUST be canonicalised for signature validation and the message has to be Base64-decoded afterwards. Informational examples for applying Security Multiparts for MIME and S/MIME version 2 are given in Annex B and Annex C, respectively.

As mentioned above, the implementation can be used in *any desired environment* for delivery of any type of data. Data type independence means that the receiving application must be able to recognise the type of data received. For that reason, if inserting an HL7 message into a MIME body, a content-type identifying HL7 messages MUST be used. Thus, the content-type `application/x-EDI-HL7` SHOULD be applied. Additional parameters (for example syntax and version) MAY be stated in the content-type to specify encoding rules, for instance. When operating in an *HL7environment*, data type independence MUST NOT be attended to, since the HL7 interface definitely knows that only HL7 message data is sent between applications. For that reason when inserting HL7 messages, the specialised content-type `application/x-EDI-HL7` need not be used, but the content-type chosen MUST be able to carry the additional parameters as well. Another possible solution is to map the HL7 message (including the additional protocol parameters mentioned above) into a X12 message using the standardised mapping rules, and to insert the result into the content-type `application/EDI-X12` defined in [RFC1767]. Other content-types could not be used as they do not feature the additional protocol parameters mentioned above. MIME encapsulation of X12 and EDIFACT objects is specified in [RFC1767] using the content-types `application/EDI-X12` and `application/EDIFACT`.

For delivery of EDI messages, general requirement for interoperable EDI and security-related issues are found in [EDI-REQ].

10.3.3.4.2 Encapsulating Signed MIME Messages for Transport

When transporting signed data using `multipart/signed` by Internet (http, mail) or end-to-end in non-MIME environments, gateways are generally not aware of MIME security and treat this content-type as `multipart/mixed` or also apply conversions to the MIME structure and its contents according to the local format. Thus, either the original message cannot be reconstructed and the signature cannot be verified, or the signature verification fails.

To counter this problem, [SMIME2] and [SMIME3] propose two solutions. Either the content-type `application/pkcs7-mime` or the content-type `application/mime` SHOULD be used to pass signed data through the gateway, intact, for an S/MIME facility. The major difference between these two alternatives is that the first uses a PKCS#7 object and the latter encapsulates the whole `multipart/signed` entity.

The encapsulation using `application/mime` has been also specified by [APP-MIME], but this Internet-Draft is expired and has been deleted without publication as a RFC.

A description for secure exchange of EDI documents using http transport is given in [MIME-HTTP].

10.3.3.4.3 Non-Repudiation

According to the "Standard Guide for EDI (HL7) Communication Security," non-repudiation of origin (NRO) and receipt (NRR) SHOULD be provided for the transmission of message data (EDI messages and other data files).

Generally, *NRO* MUST be provided by inserting information about the sender (in the role of the signer) as its distinguished name or public key (certificate).

For PKCS#7 and CMS, the `signedData` object MUST be used to assure NRO. This can be achieved by including certificates or authenticated attributes. For PKCS#7-objects, certificates are included using the field `ExtendedCertificatesAndCertificates` for a set of PKCS#6 extended certificates and X.509 certificates (chains of certificates) or using the field `ExtendedCertificateOrCertificate` for either a PKCS#6 extended certificate or an X.509 certificate. For CMS-objects, certificates are included using the field `certificates` containing a collection of PKCS#6-certificates (obsolete for CMS) or X.509 (attribute) certificates.

Authenticated attributes are inserted in the attribute `authenticatedAttributes` of the field `signerInfo` for PKCS#7-objects, whereas the attribute `signedAttrs` is used for CMS-objects. For MOSS, the field `Originator-ID:` can hold the DN (including email if desired) or the public key of the sender (originator).

For providing *NRR*, signed receipts MUST be used. In general, NRR can be realized by the MIME-syntax itself or the cryptographic objects embedded. The way of providing NRR by MIME-syntax is given by [RFC1892], [RFC2298] as well as [MIME-SECURE] and described in Chapter 10.3.3.4.3.1. Following this scenario, NRR can be provided independently of the objects embedded. When using S/MIME version 3, NRR MUST be provided by the CMS-objects embedded in combination with the Enhanced Security Services (ESS) as defined by [SMIME3]. This scenario is described in Chapter 10.3.3.4.3.2. There is no other way to offer NRR yet. No NRR support is available on the PKCS#7-level.

Since the return of message content MAY be wasteful of network bandwidth and time, an appropriate strategy SHOULD be chosen. Thus, only the hash value of the last message received SHOULD be included and not the full message itself.

10.3.3.4.3.1 NRR for MIME-Object Security Protocols

When using MIME-object security protocol as Security Multiparts for MIME, S/MIME version 2, MOSS or PGP/MIME the following specifications and formats for receipts and signed receipts MUST be applied for provision of NRR. For S/MIME version 3, NRR MUST be implemented as given in Chapter 10.3.3.4.3.2.

The format of *requesting and the format of receipts* are defined in [RFC2298]. The format of *signed receipts and their requests* are specified in [MIME-SECURE] Chapter 5. In order to request a signed receipt, the sender places the following headers before the first content-type of the message. The header `Disposition-notification-to:` contains the return address (usually mail address), `Disposition-notification-options:` as well as its parameter `disposition-notification-parameters=` specifies how and what (as protocol and message digest algorithm) message disposition notifications should be generated.

Receipts are built using the content-type `multipart/report` as defined in [RFC1892] that encloses bodies for textual status description (*first body*; for instance content-type `text/plain`), for message disposition notification (*second body*; MDN, namely the con-

tent-type message/disposition-notification) as specified in [RFC2298] and a" reference" to the original message (*third body*). For human diagnosis, the textual status description (first body part of multipart/report) can be used to include a more detailed explanation of the error conditions reported by the disposition headers. Following [RFC2298] and [RFC1892] for receipts, the original message (if encrypted, in its encrypted form) or part of it (for instance received headers) should be included as a third body part (optional body part) or omitted if message headers are not available. Full message inclusion is only recommended if the request level is absent, otherwise partial inclusion is recommended. In any case, the reference is achieved by the field Original-Message-ID: in addition to other fields like Reporting-UA:, Original-Recipient:, Final-Recipient: and Disposition: of the second body part without any security protection (for example: possible forgery of MDNs).

Signed receipts are built following [MIME-SECURE] using the content-type multipart/report as described above, but the Base64-encoded MIC (message integrity check or message digest) of the original plain text message is inserted into the new field Received-content-MIC: in the second body to establish the reference. For any signed messages (this means that signed/encrypted must be decrypted first), the MIC to be returned is calculated on the canonicalised (multipart) MIME header and content. For encrypted-only messages, the MIC to be returned is calculated on the decrypted and, afterwards. canonicalised (multipart) MIME header and content. For plain text messages the MIC must be calculated over the message contents before their transfer encoding and without any MIME or other headers. Returning the original or parts of the received message in the third body of multipart/report is not required (optional body part), but placing the received headers into that body is recommended. At last, the complete content-type multipart/report is signed *after* its canonicalisation using application/pkcs7-mime with smime-type=signed-data or multipart/signed for S/MIME version 2, or multipart/signed for secure MIME.

For *validation*, the MIC contained in multipart/report received from the server must be compared with the MIC calculated by the client.

For *bundling purposes*, the server's response comprised of the reply message and the signed receipt (the whole content-type multipart/report as described above but unsigned and unencrypted), are bound together by the content-type multipart/related [RFC2112]: The server computes a reply message and inserts this message into the MIME entity (for HL7 the content-type application/x-EDI-HL7). This entity is inserted as the first part of the multipart/related MIME entity. The multipart/report entity is inserted unsigned and unencrypted as the second body part. A prototype of the multipart/report entity is shown in Figure 10.7.

Then, the multipart/related entity (parameter type application/x-EDI-RESPONSE and consisting of two bodies) is canonicalised and then signed. If confidentiality is needed, the result itself can be enveloped.

To summarise, there are only two transactions between client and server (if the client abandons sending a MDN receipt for the server's response in turn): The client sends an request message including a request for a signed receipt and the server responds by transmitting the reply message and the receipt signed and encrypted as explained above.

```
Content-Type: multipart/related;<CR><LF>
type="application/x-edi-response";<CR><LF>
boundary="<boundary1>"<CR><LF>
<CR><LF>
```

```
--<boundary1><CR><LF>
Content-Type: application/x-EDI-HL7<CR><LF>
Content-Transfer-Encoding: base64<CR><LF>
<CR><LF>
<base64-encoded EDI reply message>
<CR><LF>
--<boundary1><CR><LF>
Content-Type: multipart/report;<CR><LF>
report-type="disposition-notification";<CR><LF>
boundary="<boundary2>"<CR><LF>
<CR><LF>
--<boundary2><CR><LF>
Content-Type: text/plain<CR><LF>
Content-Transfer-Encoding: 7bit<CR><LF>
<CR><LF>
<some text describing the status>
<CR><LF>
--<boundary2><CR><LF>
Content-Type: message/disposition-notification<CR><LF>
Content-Transfer-Encoding: 7bit<CR><LF>
<CR><LF>
Reporting-UA: <ua-name>; <ua-identifying-string><CR><LF>
Final-Recipient: <address-type>; <generic-address ><CR><LF>
Original-Message-ID: <message-id><CR><LF>
Disposition: <action-mode>/<sending-mode>;<CR><LF>
<disposition-type>/<disposition-modifier ><CR><LF>
Received-Content-MIC: <mic>,<micalg><CR><LF>
<CR><LF>
--<boundary2>--<CR><LF>
<CR><LF>
--<boundary1>--<CR><LF>
```

Figure 10.7: Prototype of the `multipart/related` Content-type

10.3.3.4.3.2NRR for S/MIME Version 3

When using the *S/MIME version 3* as defined by [SMIME3], the Enhanced Security Services for S/MIME (ESS, [SMIME3]) MUST be used for providing NRR by signed receipts. The ESS use the CMS (Cryptographic Message Syntax) as defined by [SMIME3]. The CMS is derived from PKCS#7 version 1.5. Signed receipts may be requested only if a message is signed and can optionally be encrypted by the sender of the receipt.

As described in Chapter 2 of the ESS specification, the *request* is indicated by adding the attribute `receiptRequest` to the `authenticatedAttributes` field of the `SignerInfo` object for which the receipt is requested. The attribute `receiptRequest` consists of the fields `signedContentIdentifier`, `receiptsFrom` and `receiptTo`. The field `signedContentIdentifier` is used to associate the signed receipt with the message requesting the signed receipt by a unique message identifier. Entities which has been requested to return a signed receipt are noted in the field `receiptsFrom`. For each entity to whom the recipient should send the signed receipt, the message originator must provide the `GeneralNames` (usually the originator's name only) in the field `recipientTo`.

A *signed receipt* is a `signedData` object encapsulating the receipt object identifier and the attribute `receipt` (in `encapContentInfo`) that consists of the fields `version` (set to 1 for now),`contentType`, `signedContentIdentifier` and `originatorSignatureValue`. The object identifier from the `contentType` attribute of the origi-

nal message is copied into the contentType field of the receipt attribute, and the value of the signedContentIdentifier is copied also. The signature digest (including the receiptRequest attribute) of the original signedData object is copied into the field originatorSignatureValue.

The field authenticatedAttributes of signerInfo (a field of signedData) contains the attributes messageDigest, msgSigDigest, contentType and other attributes (for example the signingTime) indicating the time the receipt was signed.

The receipt is signed and the digest is included in messageDigest, the digest value, calculated to verify the signature of the original signedData object, is included in msgSigDigest and the receipt object identifier is inserted into contentType. At last, all authenticated attributes are signed and the signature is included in signature of signerInfo.

The signedData object is then put into an application/pkcs7-mime body with the parameter type signed-receipt. If this object should be encrypted within an envelopedData object, then an outer signedData object must be created encapsulating the envelopedData object, containing a contentHints attribute with the receipt object identifier as contentType. This is needed for the receiving agent to indicate that a signed receipt is contained within an envelopedData object.

To *validate a signed receipt*, the requestor must retain either the original signedData object, or the signature digest value of the original signedData object (contained in signature of signerInfo) and the digest value of the attribute receipt.

First, contentType, signedContentIdentifier and originatorSignatureValue are extracted from the receipt attribute to identify the original signedData object that requested the receipt.

Now, the digest of the original signedData object is compared with the value of msgSigDigest. If the originator has not retained the digest, it must be recalculated. If these values are identical, it is proven that the digest calculated by the recipient is based upon the received original signedData object including the same authenticatedAttributes containing the receiptRequest.

Then, the digest calculated by the originator for the receipt attribute is compared with the value of messageDigest. If the originator has not retained the digest, it must be recalculated. If these values are identical, it is proven that the recipient received the original signedData object signed by the originator to build the receipt attribute.

At last, the originator verifies the signature of the received signedData object (signature field of signerInfo) using the calculated digest of authenticatedAttributes. If the signature verification is successful, the integrity of the received signedData object containing the receipt attribute is proven and the identity of the recipient included in signerInfo is authenticated.

10.3.4 The Secure File Transfer Protocol (SFTP)

In this section, the communication protocol over TCP/IP-based networks is described that offers user and system authentication as well as a secure control and data connection according to the "Standard Guide for EDI (HL7) Communication Security." This is achieved by exchanging the tokens given in this guide (see Chapter 10.3.3). This protocol is a security-enhanced version of the fundamental file transfer protocol given in [RFC0959] and is based solely on standards (e.g. ISO, NIST FIPS-PUB, ANSI and IETF/IESG RFCs). The protocol is called the secure file transfer protocol (SFTP).

File transfer of HL7 messages (batch processing) is carried out by transmitting one or more messages grouped in a file and encoded according to the encoding rules of HL7. Responses are grouped and transported similarly. According to communication security requirements, SFTP wraps HL7 messages applying various selectable, cryptographic message syntax such as PKCS#7, security multiparts for MIME, S/MIME (version 2 or 3), MOSS or PGP/MIME. Security based on MIME takes advantage of the object-based features of MIME and allows secure messages. In general, SFTP is independent of the cryptographic syntax used; thus, any other syntax can be implemented without much effort. Moreover, SFTP is able to process any desired type of file data as EDI messages, including EDIFACT, HL7, X12, xDT and others, or arbitrary binary data. Different operation modes (i.e. plain text, signed-only, encrypted-only or signed-and-encrypted) can be selected for message transmission according to the security policy given. Character encoding using the Base64-encoding scheme is selected and canonicalisation is applied for system interoperability, preventing loss of data bits that may lead to invalidation of the digital signature. For establishing a public key infrastructure (PKI) using trusted public keys, all public keys are embedded into a certificate whose structure follows X.509, and the distinguished names (DN) used therein conform with X.501. The certificates are stored and managed in X.500 or LDAP directories.

10.3.4.1 The Protocol Model

The Secure File Transfer Protocol (SFTP) is based upon the TCP/IP protocol suite using the FTP client/server model, as defined in [RFC0959], regarding the additional requirements of [RFC1123] (Chapter 4) that FTP implementations should follow. The TCP/IP protocol suite, compared to the OSI model, is presented in Figure 10.8.

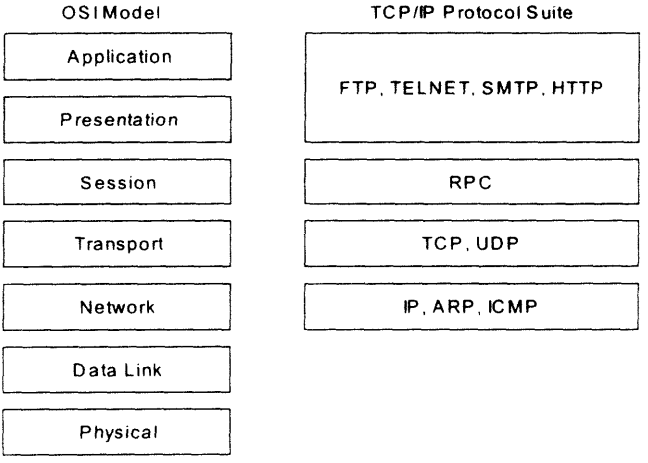


Figure 10.8: The TCP/IP Protocol Suite compared to the OSI model

An overview of the SFTP process model is shown in Figure 10.9. It is derived from the fundamental FTP model given in [RFC0959]. The protocol interpreter (PI) and the data transfer process (DTP) involved realise FTP processing by analysing and evaluating commands and replies (the part of the PI) as well as performing data transfer if needed (the part of the DTP). Thus, the PI is managing the control connection and the DTP is responsible for the data connection.

Basically, the SFTP process works like this. The server is listening on the well-known FTP service port (TCP port 21) waiting for a client connecting to that port. If the client performs a connection (from a dynamic port X), a so-called *control connection* is initiated that remains active for the whole session. On this connection, the client sends commands to the server and the server responds by sending reply codes using this connection in full-duplex operation mode. Normally, the control connection is closed by the client by sending an appropriate command (QUIT), but the server could also close the control connection in case of serious errors.

The data transfer is performed by establishing a second temporary connection in simplex operation mode. There are two modes for the establishment of such *data connection*:

1. In *active mode*, the client listens on a dynamic TCP port Y and sends a PORT command containing his IP address and port Y to the server, which then attempts to connect to that IP address and TCP port.
2. When using *passive mode*, the client sends a PASV command to the server, which listens on port 20 (or alternatively on a dynamic port) and informs the client where to connect by sending an appropriate reply code containing its IP address and TCP port.

As stated in [RFC1579], the passive mode should be preferred for firewall-friendly FTP. Switching between the active and passive data connection mode must be possible at any time.

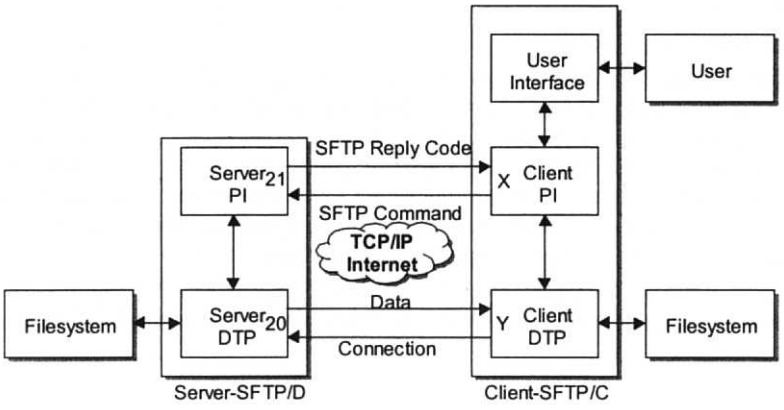


Figure 10.9: SFTP Process Model

All transfers (control and data connection) performed by the original RFC0959-FTP protocol are unsecured and have no security services such as strong authentication, confidentiality, integrity or accountability. Only simple authentication is carried out by transmitting the password in plain text using the USER and PASS command.

Looking at the process model described above, the enhancement of security for the FTP protocol MUST be located at the PI *securing the control connection* and at the DTP *securing the data connection*. Furthermore, before the client could perform any command (except the command to request authentication) and data transfer on the server, a *strong mutual authentication* MUST be performed between them. This is exactly the approach realised by SFTP. For the enhancement of security, many standard documents available are considered such as ISO Standards, IETF/IESG Internet Standards (RFCs), IETF Internet Drafts (IDs) and NIST publications (NIST FIPS PUB).

In addition, both client and server MUST apply timers to check if a connection is timed-out, that is, if the response or chained commands are out of time. This MUST be performed for the control and data connection as well as if the server is running on idle.

10.3.4.2 Strong Mutual Authentication

For *user authentication*, the human user SHOULD provide his or her HPC user name and PIN in combination with biometrics. After the SC has been opened successfully, all objects (e.g. keys) MUST be checked for completeness and validity (for instance certificates). During the SFTP session, the chipcard needs to be kept inserted in the chipcard terminal (timed chipcard request). When removing the chipcard, the application inhibits further operations and only continues to work if the chipcard is inserted again and the user authentication that follows is successful.

Before the SFTP client can perform any command (except the command to request authentication) and data transfer on the server, *strong mutual authentication* MUST be performed between them as described in Chapter 10.3.3.2.2.

As given in Chapter 10.3.3.2.2, a unique identifier is included for each token exchanged to indicate its type and position in the exchange as shown in Figure 10.5. The following values SHOULD be used (in the style of [FIPS196] appendix A) in "byte" representation:

$\text{TokenID}_{\text{AuthReq}} = 0x10$

$\text{TokenID}_{\text{AuthData1}} = 0x11$

$\text{TokenID}_{\text{AuthData2}} = 0x12$

$\text{TokenID}_{\text{AuthData3}} = 0x13$

The sequence number SHOULD have the data type "word" (2 bytes, little endian order). Two time stamps SHOULD be included: one time stamp for token generation time and one for token expiration time. For time stamp generation, UTC time MUST be used and converted to seconds for the purpose of comparison (using "dword" (4 bytes, little endian order) representation). The time window MUST be of an appropriate length according to the physical properties of the underlying network (e.g. not smaller than 2 minutes. Role and state SHOULD have "byte" representation, all other items "string".

For token formatting, the tag-length-value (TLV) format MUST be applied. The values used for the tag-byte of the TLV format (see Table 10.5) are presented in Table 10.6 and MUST be used.

Table 10.6: Valid Values for the TAG-byte

TAG-byte	Purpose
0x00	Token identifier
0x01	Sequence number
0x02	Time stamp for token generation time
0x03	Time stamp for token expiration time
0x04	DN of initiator (client)
0x05	DN of responder (server)
0x06	IP address of initiator (client)
0x07	IP address of responder (server)
0x08	MAC of initiator (client)
0x09	MAC of responder (server)
0x0a	Role indicator (initiator/responder)
0x0b	State indicator (request/invitation)
0x0c	Random number 1
0x0d	Random number 2
0x0e	Random number 3

0x0f	Authentication mechanism, command or reply code
0x10	Hash value for NRR
0x11	Digital signature

According to the "Standard Guide for Specifying EDI (HL7) Communication Security" and Chapter 10.3.3 of this implementation guide, all token bytes (all fields including TLV encoding) MUST be Base64-encoded, canonicalised before delivery for interoperability reasons and decoded on the server before evaluation. Base64-encoding protects against loss of data bits in environments not capable of full binary transport. Canonicalisation is performed after the encoding process to prevent system dependency. Applying neither encoding nor canonicalisation may lead to invalidation of the digital signature.

The commands and reply codes for FTP authentication MUST be implemented in the style of [RFC2228]. AUTH is used for authentication request and security mechanism transmission. ADAT is applied for transmission of authentication data. The AUTH command is used by the client to request authentication by giving an authentication mechanism as argument. Valid mechanisms must be registered with the IANA (Internet Assigned Numbers Authority) and can be found at [RFC2222] or [IANA]. For local use, the values begin with "X-", so for this protocol "X-SFTP" is applied.

The authentication mechanism "X-SFTP" is embedded in the token field AUTH-Mechanism of the token AuthReq', which is built according to Chapter 10.3.3.2.2 step 1. All tokens containing authentication data such as AuthReq, AuthData1, AuthData2 and AuthData3 are sent to the server as an argument of the ADAT command. Figure 10.10 shows the flow of authentication tokens for SFTP.

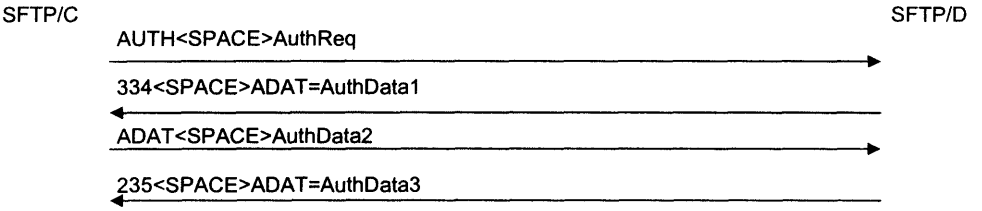


Figure 10.10:Flow of Authentication Tokens Exchanged for SFTP

After the authentication has been successfully performed, authorisation based upon the user's identity MAY be carried out by the server. The identity involved is obtained from the DN contained in the authentication tokens. Thus, no additional USER command must be used as explained in [RFC2228].

The checking of time stamps, as mentioned above, only applies either when synchronised clocks are available in a local environment, or if clocks are logically synchronised by bilateral agreements. In any case, Coordinated Universal Time (UTC) and secure time servers must be used.

10.3.4.3 Securing the Control Connection

When authenticated successfully, the control connection MUST be secured as described in Chapter 10.3.3.3. The client commands and server reply codes MUST be in the style of [RFC2228]. Command tokens MUST be generated according to Chapter 10.3.3.3.2 and are sent as an argument of the security commands of [RFC2228] (for example:

MIC<SPACE>Token$_5$ for signed transmission). The reply of the server MUST be generated analogously and the codes follow [RFC2228] (for example: 631<SPACE>Token$_6$).

10.3.4.4 Securing the Data Connection

The data connection MUST be secured as described in Chapter 10.3.3.4 and provide integrity, confidentiality and non-repudiation of origin and receipt. Switching between the cryptographic protocols (e.g. S/MIME version 2, S/MIME version 3, MOSS) and their operation modes (e.g. signed-only, signed-and-encrypted) as well as selection of the session key MUST be possible. The 'PROT' command, as defined in [RFC2228], is restricted and not well specified and does not allow more than one different protocol. Therefore, this protocol uses the 'PROT' command with a word encoded argument (2 Bytes in little endian order).

The first byte (low byte) MUST state the cryptographic protocol and its operation modes as detailed in Table 10.7. All unused entries of this byte between hexadecimal 0x00 and 0x3F are user-definable, other values MUST not be allocated or re-allocated.

For now, MIME Security Auto-detection (value 0x3F) MUST be used only for MIME-object security protocols, such as Security Multiparts for MIME, S/MIME version 2 and 3, MOSS and PGP/MIME. When setting auto-detection, the receiving application knows that something is transmitted using MIME-object security, but it neither knows the specific MIME-object security protocol nor the operation mode: signed-only, encrypted-only or signed-and-encrypted. The auto-detection mechanism identifies the MIME-object security protocol and the operation modes. Furthermore, this mechanism must be able to process files containing multiple messages that may also vary in their MIME-object security protocol and operation mode. Automatic detection is based upon the MIME type indicated by the content-type and the evaluation of accompanying parameters.

The operation modes SHOULD only be given if non-MIME protocols are used as PKCS#7-only and CMS-only. In this case, the value stating the protocol and the value of the desired operation modes are combined using OR-operations bit by bit. For example, PKCS#7-only in signed-and-encrypted operation mode will result in the value ((0x10 OR 0x40) OR 0x80) = 0xD0.

Table 10.7: Encoding for the Cryptographic Protocol and its Operation Mode

Value	Usage
	Cryptographic Protocol:
0x00	Plain Text (ASCII)
0x10	PKCS#7-only
0x11	CMS-only
0x20	Security Multiparts For MIME
0x21	S/MIME Version 2
0x22	S/MIME Version 3
0x30	MOSS
0x31	PGP/MIME
0x3F	MIME Security Autodetection
	Operation Mode:
0x40	Sign
0x80	Encrypt

The second byte (high byte) of the 'PROT'-command argument MUST define the session key algorithm as shown in Table 10.8. Here, all unused entries are user-definable.

Table 10.8: Encoding for the Session Key Algorithm

Value	Usage
0x00	IDEA
0x10	DES-EDE2-CBC
0x11	DES-EDE3-CBC

10.3.4.5 Security Considerations Regarding the Protocol Stack

According to the "Standard Guide for EDI (HL7) Communication Security," the specification of the protocols used, such as FTP, TCP, and IP contains a number of mechanisms that can be used to compromise network security. There are many known Internet attacks based on infrastructure weakness, such as DNS spoofing, source routing (IP spoofing), FTP bouncing, racing authentication and denial of service. Attacks arising from the weakness of the FTP protocol and underlying protocols SHOULD be addressed by this protocol regarding [FTPSEC] or [CERT].

Racing authentication, which is based on faster authentication of the attacker than the victim, SHOULD be prevented by the strong mutual three-way authentication, based on challenge/response and digital signature, and the restriction to one simultaneous login of the same user. Moreover, the total number of control connection possible SHOULD also be limited.

To protect against FTP bouncing (namely the misuse of the PORT command), the server SHOULD not establish connections to arbitrary machines (for instance a second FTP server called proxy FTP) and ports on these machines. Following [CERT] and [FTPSEC], the server SHOULD ensure that the IP address specified in the PORT command matches the client's source IP address for the control connection. Furthermore, the server MUST disallow data connections if the TCP-port specified in the PORT command is a well-known port (port 0 to 1023) or registered port (1024 to 49151). Only dynamic, private ports (port 49152 to 65535) are allowed. Hence, a port scan against another site hiding the true source and bypassing access controls like firewalls cannot be performed (for instance bouncing to a well-known port). The PORT command is used in the active mode only It is not used in the passive mode that is initiated by the PASV command. Since the PASV command is not affected by the bounce attack since the server gives the IP address and port to connect to and an attacker cannot act as a server, it is preferred to the PORT command providing firewall-friendly FTP (see [RFC1579]),as well. Using passive initiation of the data connection means that the TCP connection establishment is performed from the client network toward the server network.

Furthermore, random local port numbers SHOULD be used for the data connection, as stated in [FTPSEC], to address port number guessing. Guessing the next port number is much easier when simple, increasing algorithms are used (for example: next port = old port + constant number). Using simple, increasing algorithms enables attacks like the denial of a data connection or the hijacking a data connection to steal files or insert forged files.

In addition to the authentication procedures, access restrictions based on network addresses MAY be provided. The server accepts only connection requests from pre-defined IP addresses within authorised organisations and confirms this address matches on both the control connection and the data connection. When relying on IP address authentication only, an attack like source routing of IP packets (IP spoofing) is possible.

To address DNS spoofing, hostname to IP address resolution or vice versa (DNS) SHOULD NOT be performed by client or server. The destination machine SHOULD be reached by the IP address directly.

For the detection of compromises such as denial of service attacks and other attacks, the server SHOULD keep reports logging all activities including connection attempts, discon-

nection, command executions and others. Since local machines are considered trusted, integrity and/or confidentiality protection is not required.

10.4 Implementations

At the Magdeburg Medical Informatics Department, the generic open solution for security enhanced EDI communication has been implemented and is now used in routine. At the HL7 Working Group Meeting in Baltimore, USA, in April 1997 this solution was publicly demonstrated using open networks as the Internet and the TH.HPC. Annex C describes this implementation in detail for the HL7 example, especially mentioning security wrapping mechanisms for FTP providing an open secure FTP (SFTP) protocol.

10.5 Summary and Conclusions

Based on the generic systematic methodology for modelling secure health information systems, a solution for security enhanced EDI communication has been specified, developed, and implemented. This solution is a really open one considering both secure messaging – i.e., the provision of end-to-end security also called message security, object security or message wrapping – and secure channel providing system to system security. While the first solution requires security aware applications, the second solution facilitates a security framework or interface system usable for many systems, also legacy ones.

The security enhanced EDI communication has been implemented for a secure FTP developed wrapping HL7 messages (see Annex C). It is routinely used now. The security framework is applicable to any message systems including HL7, EDIFACT, XML, XDT, etc. Both standard guides [Blobel et al., 1998a,b] have been approved as informative part of the ANSI accredited HL7 communication standard. They complete the practical HL7 security solution of secure email which is covered by our approach and has been specified for implementation in [Schadow et al., 1998].

11 Secure Chipcard-Based Health Information Systems – the DIABCARD Example

11.1 Introduction

To meet the challenge of *shared care*, the specialisation and decentralisation must be accompanied by comprehensive communication and co-operation. The corresponding communication may be supported through any kind of networks from a departmental Local Area Network (LAN) up to the Internet. Another way is the connection of patient's information with the patient's being itself: Acting as data subject and data source, but also as carrier of any data collected, the patient can realise the informational self-determination guaranteed by the privacy acts and/or constitutions. For electronic health information systems held by patients, an appropriate carrier is needed to store the person's medical data. Additionally, also an environment must be provided for the authorised use of the information in the sense of collecting, storing, processing, and communicating the data. Starting in Europe, smart cards, i.e. microprocessor cards are used for that purpose around the world.

Generally, it should be mentioned that the smart card could be deployed in two ways. On the one hand, the card could bear all information needed, in the case of PDC, e.g., all relevant medical data, as shown in this chapter. On the other hand, the card can be used as a pointer providing references and linkage to the information stored in networked systems. However, also a combination of those two principles could be imaginable and probable for the future.

Based on the European security infrastructure for health, the chapter describes the first security solution for a PDC environment really implemented for the currently used as well as for the next generation DIABCARD architecture. It doesn't discuss the genuine issues of the DIABCARD project itself as the DIABCARD data set, the application around, etc.

Sharing sensitive personal medical information requires the provision of appropriate data protection and data security in both, the network-based health information system and the chipcard-based one. This chapter concerns two projects funded by the European Commission: the DIABCARD project [DIABCARD_WWW] and the TrustHealth project, mentioned earlier already [TRUSTHEALTH_WWW]. The chapter deals with the secure use of a specific Patient Data Card (PDC) held by the patient and called DIABCARD, which facilitates the communication and co-operation between them, GPs and secondary care departments providing medical services for diabetes patients.

11.2 Advantages and Disadvantages of Network-Based and Chipcard-Based Health Information Systems

Already at the beginning of the nineties, the German Medical Informatics Association GMDS has defined motivation and objectives as well as health-political aspects for the use of machine-readable cards in health [GMDS_AG]. As a short-term challenge, the implementation of pilots has been mentioned to demonstrate the possible use of such cards and to rationalise time-consuming administrative work in the context of patient's request for health services. The health-political aspects concern

- facilities for decentralised medical documentation,
- the coincidence with constitutional and data protection rights as well as with ethical principles keeping the Electronic Health Care Record (EHCR) by the patient,

- the improvement of data quality, integrity and consistency by a unique document and finally,
- the improvement of quality and efficiency of health delivery in general.

Regarding network-based and chipcard-based health information system, series of advantages and disadvantages can be stated.

In pure chipcard-based health information systems,

- the medical and administrative workflow can be optimised,
- within the *shared care* framework, the information flow between different healthcare providers can be improved,
- information security can be enhanced by reliable, valid and in time and location available data,
- due to the availability of patient information, the stress caused by repeated observations may be reduced,
- the emergency care can be facilitated by the directly available emergency data set,
- prevention and intervention studies may be supported,
- autonomy and responsibility of patients as well as the partnership between patients and Health Professionals may be increased.

However, there are also some problems like

- the tough procedure of standardisation in health terminology and procedure,
- the lost of information due to the restriction of data by the storage capacity limitations,
- the unsolved legal problems of information ownership in health,
- the need of the required infrastructure and appropriate interface to applications,
- the impossibility of teamwork between different medical parties including pre- and post-caring activities due to the need of the physical presence of the patient with her PDC at all the sites involved, and
- the lost of informational *shared care* interactions.

In pure network-based health information systems with extended network architecture, services and the existence of unique identifiers the following advantages can be found:

- A comprehensive interoperability and real interactions between all parties involved into the *shared care*;
- The realisation of a comprehensive, complete, high-quality EHCR;
- The emergency care can be facilitated by the directly available emergency data set;
- Data quality, integrity and consistency, if the appropriate services are provided and an extended network is available;
- The opportunity of pre- and post-activities independently or co-ordinated done at the sites involved;
- The medical and administrative workflow can be optimised;
- Due to the availability of patient information, the stress caused by repeated observations may be reduced, and last but not least;
- Prevention and intervention studies may be supported.

As problems and disadvantages occur for example

- the emergency care is not supported directly,

- the patient doesn't held her data and must therefore trust the doctor-patient relationship, and
- data security solutions are system-related only

The way Europe is going is to combine the architectural approach by both using smart cards as token for identity authentication and person-related security services like accountability, reliable authorisation, access control and audit etc., as well as by introducing PDCs.

11.3 The DIABCARD

The DIABCARD card is a microprocessor card with a storage capacity of 16 Kbytes (approx. 12 Kbytes for medical data) for patient data, therefore also called a Patient Data Card (PDC). It contains the European administrative data set, the European emergency data set, the DIABCARE basic information sheet (a follow of three records of essential data) as well as groups of actual data describing essential information of the last visit in the ophthalmologic, the internal, and the foot care department of a Health Care Establishment (HCE). The authentication of the cardholder as the card owner is provided by entering the Personal Identity Number (PIN) [Engelbrecht et al., 1997]. At the time of writing, the mutual authentication between the DIABCARD PDC and the Health Professional Card (HPC) introduced already in Chapter 9.4 as well as the authentication of the patient towards his/her card using a patient identity card is not yet supported due to the lack of appropriate standards, but this enhancement is in preparation now. Therefore, some of the security services needed had to be delegated first to the application environment as shown in the next sections.

11.4 DIABCARD Threats

The DIABCARD is used to provide communication and information needed to the Health Professionals (HP) under control of the patient realising his/her right of informational self determination. For that reason, a card reading device and a PC with a specific DIABCARD application is needed. The PDC and the DIABCARD workstation have to be protected by authentication services as a Personal Identity Number (PIN) and strong authentication of the user to the Workstation using an authentication token like an HPC instead of the usual password. Specific equipment and an appropriate application scenario help to respond to this challenge. The installation of the DIABCARD-TrustHealth Extension project was also caused to fulfil the requirements. Furthermore, the information is often integrated in departmental systems.

Because the DIABCARD contains parts of a medical record related to the patient as the card holder, the European and the corresponding national legislation require appropriate security solutions to protect the patient and his personal medical data from attacks. These attacks could be active or passive, causing accidental or intentional threats. In the DIABCARD scenario, threats occurring are, e.g.,

- lost or theft of the PDC,
- access to the card by unauthorised users,
- unauthorised manipulation of information,
- unauthorised access to the DIABCARD system,
- unauthorised access to the data stored locally or remotely (at the departmental server site), and
- unauthorised manipulation of the DIABCARD workstation implementation.

Introducing the HPC, these threats and the resulting, in the health environment essential risks can be avoided. Based on cryptographic algorithms, the HPC provides the basic security service strong authentication including user's attributes, but also integrity check, confidentiality and accountability services.

11.5 Overall Description of the Pilot and Security Requirements

In the following section an overall description of the pilot is given by explaining two typical scenarios for a patient-doctor-system interaction including the smartcards used. Afterwards, the security requirements concerning the application (first scenario) and the communication (second scenario) are listed in detail. Regarding these requirements, the security solution is presented in the next chapters. Figure 11.1 presents the overall TrustHealth-DIABCARD Extension Scenario. As explained in the further chapters, only the application security issues at the local DIABCARD workstation including also services for integrity, confidentiality, accountability, notary's services, and audit as well as the steps 1-4, 6 and 8 of communication security services have been implemented.

The professional-related security services are provided by the HPC bearing the 3 keys for authentication, integrity, and confidentiality as well as the certificates needed. Storing the certificates on the card enables the use of isolated workstations as specified as the original DIABCARD scenario. In a network environment, certificates may be hold in TTP directories saving storage capacity especially in the context of the advanced PDC. Furthermore, the certificate management including in-time revocation procedures could be speeded up by that way or even enabled. In the international DIABCARD environment, encoding of data on the DIABCARD PDC using either the PDC or the HPC encoding/decoding key pair will not be used currently to avoid the exclusion of partners not yet enabled for enhanced security services. As a general service however, it is inevitable for enabling, e.g., future home care activities and communications. Establishing a proper security infrastructure however, the PDC confidentiality service will be applied in regional solutions such as German federal states health networks (e.g. in Bavaria). Therefore, PDC confidentiality is in preparation for our demonstrator.

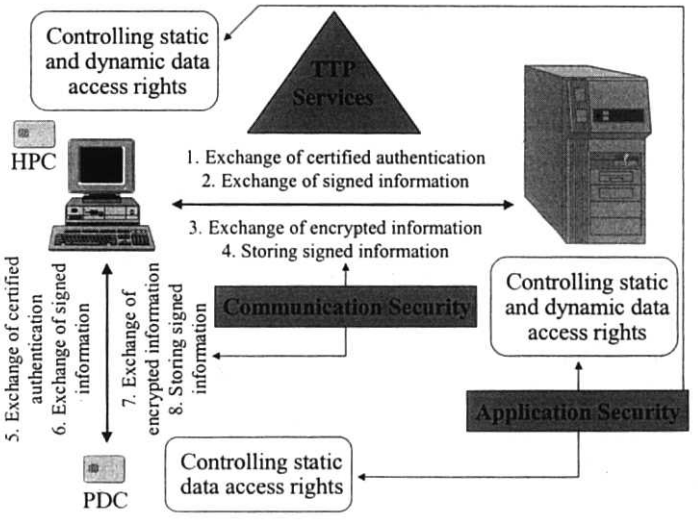


Figure 11.1: TrustHealth-DIABCARD Extension Scenario

11.6 Typical Scenarios for Interactions between Patient, Doctor and the System

As a typical scenario, a patient requests for health services going to the health provider. He/she holds a DIABCARD PDC (DIAB.PDC) and the Health Professional starts the DIABCARD client application, which is secured to avoid the misuse of data and functions in the context of the DIABCARD environment. The Health Professional (HP) authenticates his identity and role using the HPC in the sense of local authentication [CEN_1999b] and typing his Personal Identifier Number (PIN) on the keyboard of the card reader device by that way verifying that the holder is also the owner of the HPC. Based on cryptographic algorithms, the HPC provides the basic security service *strong authentication*, but also integrity check, confidentiality and accountability services. The DIABCARD application is ready now for use, i.e., for reading the DIAB.PDC, recording new items, interacting with the DIABCARD database (Paradox), and storing new or updated data on the DIAB.PDC. The patient presents the DIAB.PDC to the DIABCARD system. The application requests the patient to put the DIAB.PDC into the secure PDC reader device typing the PIN on the keyboard verifying that the holder is also the owner of the PDC. Now, the application can manage the patient data reading from or writing on the PDC and the application may be used related to the concrete patient, facilitating the communication between healthcare providers involved in the patient's care of the. In that context, also the other security services provided by the HPC are used. At the server site, the medical data of the patient is stored and updated on a departmental information system managing all data about the patient communicating with the PDC via the doctor's working place.

Another typical scenario is the secure exchange of patient-related medical data between the DIABCARD workstation and a directly or indirectly diabetes-related departmental application dedicated to manage information about the actual patient. The procedure starts with the mutual strong authentication between the HP and the departmental application using the HP's HPC and the Software Personal Security Environment (PSE) of the application server in the sense of remote authentication [CEN_1999b]. The security services are managed by the Secure FTP (SFTP) communication program presented in Chapter 10.3.4, which provides the appropriate user interface. The secure EDI communication via SFTP was firstly implemented in the framework of the MEDSEC project funded by the European Commission and dealing with the security enhancement of healthcare standards [Blobel et al., 1998a,b; Blobel et al., 1999].

The following issues are focused for realisation of the security services:

- The existing login has to be replaced by a dialogue that uses an HPC in combination with a PIN for user identification and authentication.
- Two separate card readers, a multifunctional card terminal (MCT) for the HPC, which in the future also may be integrated into the keyboard, and a PC/SC card terminal for the DIABCARD PDC are used. Mentioning the social and psychological implications, also dual slot card terminals via PC/SC may be applied.
- The HPC provides roles that have to be implemented in the DIABCARD Core System, i.e. different access rights for the various DIABCARD Data Set groups.
- The database of the DIABCARD Core System (DCC) must be protected against unauthorised access. Encryption of the database files might be a solution.
- For accountability services, digital signatures must be added when writing data onto the DIABCARD PDC and to the database.

Direct interactions between the DIAB.PDC and the TH.HPC are outside the scope of the pilot. An additional phase is planned realising these interactions. However, drafts for secure

access to patient cards via an HPC already exist. One possible solution is described in the German HPC specification [Arbeitskreis, 1997] proposing the use of card verifiable certificates (CVCs). Nevertheless, the problem for this pilot is situated on the interface level. Comparing the system architecture used in the DIABCARD project to the security architecture described by TrustHealth-1 there are fundamental differences in the smartcard interface. Moreover, there is a large gap between TrustHealth-1 and TrustHealth-2 as well. The specifications from TrustHealth-1 were overrun by quasi-standards like PC/SC mainly developed by the computer industry. In the time of writing, card terminals and software is available for both the MCT and PC/SC concept, but the specifications as well as the software and hardware products from TrustHealth 1 are MCT-based. Until now, a card terminal cannot work simultaneously as MCT and as PC/SC IFD[41], but it is possible to run card terminals with different interfaces at one computer concurrently. This also affects the other phases of the security.

Therefore, two separate card readers with PIN pad are used: A multifunctional card terminal for the TH.HPC, which may be integrated into the keyboard in the future, and a PC/SC card terminal for the DIAB.PDC (the serial communication ports for the PDC and the HPC can be changed easily allowing to work side by side). Regarding social and psychological implications, PC/SC driven dual slot card terminals may be applied too.

11.7 The Health Professional Card

The HPC is an ISO7816 conformant microprocessor card with an additional co-processor specialised for cryptographic algorithms (RSA crypto-processor) which has at least 4Kbytes for key objects (EEPROM, non volatile), 256bytes working memory (RAM, volatile) and 6Kbytes operating system (ROM). The authentication provided concerns both the identity (expressed by identity certificates) and the roles (expressed by attribute certificates) of the Health Professionals (HP). The identity certificate issued by the Physicians' Chamber guarantees the first. The latter is expressed by several attribute certificates issued by the Physicians' Chamber (specific domains of care or specific qualifications) or by the Statutory Health Care Administration "Kassenärztliche Vereinigung" (specific permissions = "Ermächtigung"). The card contains keys with dedicated usage as for authentication, digital signature and encryption (e.g. the session key) as well as the X.509v3-based certificates mentioned. In the card's Master File, the global profession (physician, nurse, etc.) is specified. Based on the identity and the roles of the user on the one hand and the decision rules agreed in the security policy on the other, the HPC enables application security services that are related to the person as authorisation, access control, integrity, confidentiality, accountability, and audit.

The chipcard terminal used within the pilot is called ICT 800 STD. It is manufactured by Giesecke & Devrient Munich following the MCT-specification. In addition to a normal chipcard terminal, which is used for ID cards, also two plug-in cards can be inserted at the bottom side of the terminal. The terminal is equipped with a keypad according to ISO/IEC 9564 and a liquid-crystal display (LCD).

11.8 Placement of Application Security Services in the DIABCARD Environment

The first step of introducing trustworthiness is the mutual strong authentication between the communicating principals, i.e. user and application. Because applications normally consist of multiple components establishing distributed systems, the mutual authentication has to

[41] Interface definition

be provided to all the components involved. Otherwise, the invocation of components not participating in the authentication procedure must be denied.

In the DIABCARD PDC version implemented in 1999, the strong authentication of that PDC was not enabled. Because also DCS source code was not available, its components (e.g., the invocation of the DIABCARD Data Access API) could not be protected by access control services. Following, the authentication/access control services have been provided on (step 1a.) ... (step 1c.) only. Therefore, the access control services could only be established via encryption hindering the unauthorised use of those components, being aware that access control and confidentiality services are coherent in certain manner[42]. The encryption was performed using a symmetric key, which is either stored on the HPC or in all authorised users' Personal Security Environment (PSE). Starting the DIABCARD application, only the authorised user could decrypt these protected components for use.

In the next version of the DIABCARD system, the components DIABCARD Core System (DCC, [Gogou et al., 1998]), the underlying Paradox database (PDD), the DIABCARD Server (DCS, [Demmer et al., 1998]) including the interfaces (DIABCARD Server API, DIABCARD Data Access API [Sulzmann, 1998]), and last but not least the PDC itself are protected by peer-authentication-based access control for authorised users only (step 1a.) ... (step 1e.) as shown Figure 11.2.

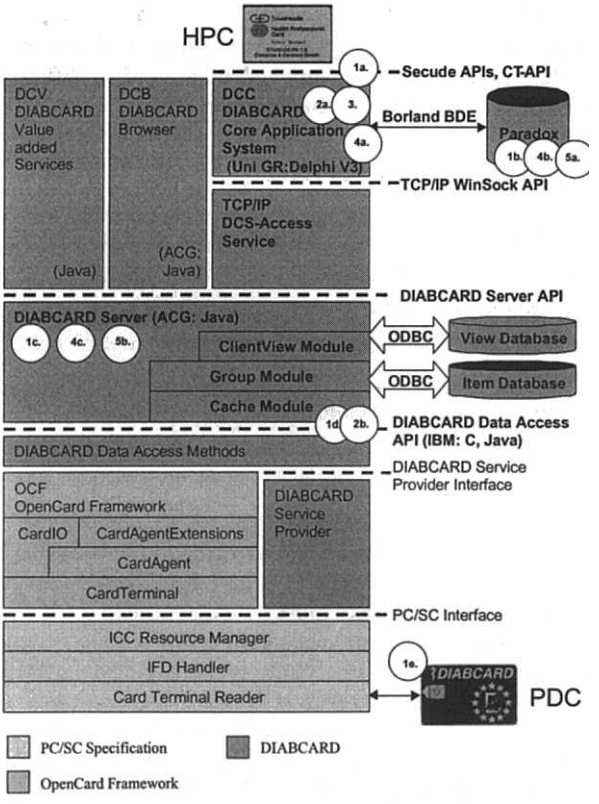


Figure 11.2: Architectural Schema and Placement of Application Security Services in the DIABCARD Workstation

[42] Nevertheless, the latter is more problematic, less stable and trustworthy. Therefore, encryption should be applied, if possible, additionally to prevent unauthorised disclosure when access control is not available (e.g. in de-mounted hard disks).

Using an HPC specified within the TrustHealth project (TH.HPC), a single logon is realised.

11.9 Application Security Services

For application security, the following services have to be provided for the software components of the client:

- Access control (user identification and authentication),
- Authorisation (role management based upon the identification/authentication process),
- Accountability (in the sense of non-repudiation of origin provided by user-related digital signature for data items or groups),
- Integrity (including data origin authentication),
- Confidentiality (encryption of data items or groups).
- Audit, and
- Notary's services (e.g. timestamps).

Excluding the last one, these services are based on the communication security services strong authentication which is enabled among the other services as integrity check, confidentiality and accountability by the HPC.

However, each component of the client system does not need all security services. Moreover, the services integrity and confidentiality can be applied to different kind of data as medical data or program data according to the intention.

11.10 Communication Security Services

Secondly, the exchange of sensitive personal medical data between the doctor's workplace and the departmental information system including authorisation and legal responsibilities among the other application security services needed for the data stored and processed requires appropriate *communication security* measures including, e.g.,

- identification, strong mutual authentication and access control between the principals communicating,
- integrity (including data origin authentication),
- accountability (in the sense of non-repudiation of origin and receipt),
- confidentiality.

The communication between two or more principals (users, components, applications, systems, etc.) might be provided at the one hand user-related or at the other one not (directly) user-related. The first communication scenario is initialised by the user directly who is accountable for this activity. Then, the user must be authenticated to the local system which communicates after its mutual strong authentication to the remote system on behalf of the user. Because only end-to-end security of security-aware principals provides maximal security, even the communication security services between the client and server systems involved may be secured by mechanisms facilitated through the user's security token HPC and its private keys for the corresponding services. In the case of communications integrated in the DIABCARD application scenario, the authentication at the application start should be used thus extending the secured components mentioned in Chapter 11.13 by the secure communication protocol (e.g. secure FTP). This helps to ensure the acceptability of the solution proposed.

The second communication scenario of not (directly) user-related communication concerns event-driven massaging (e.g. batch processes, event-driven EDI like HL7, EDIFACT,

XML) between the communicating principals (e.g. client and server). In those cases, all necessary key objects should be stored securely in a local software personal security environment (SW-PSE). Detailed descriptions of the communication security services are given in the MEDSEC project [Blobel et al., 1998a,b].

11.11 Not User-Related Security Services

Because the user should not be accountable for not (directly) user-related security services provided by non-user principals (applications, systems), such services must not be implemented within user-related PSEs (PSEs opened through the user's authentication). Therefore, third party PSEs should be introduced, related, e.g., to a system or security administrator. Such solution requires active interactions of the related accountable principal at least to start the security enhanced application. To avoid the presence of these principals at the local site which would reduce the acceptability of the solution, the third party PSE installed will be opened by the remote administrator system. This continuously active system works like a Key Distribution System or Ticket Server well-known from the Kerberos protocol. The third party PSE holds the keys needed for not user-related secure communication (e.g. secure FTP) as well as the symmetric keys needed for not user related application security services like confidentiality and integrity of program files. Also the *audit* service provided for inspection of user activities must be secured independent of the user. The keys needed to provide audit are also securely stored in the third party PSE.

During the authentication procedure, the user transparently starts a secure communication (secure FTP) with the administrator system. After the strong mutual authentication between both principals, the third party securely transfers the authentication token needed for opening its PSE. The procedure described for not user-related security services requires network-based systems. Because this architectural prerequisite cannot be stated for all possible DIABCARD test sites, the not user-related security solutions described are not implemented in the DIABCARD security demonstrator. Therefore, the keys needed are securely stored within the user PSE (the HPC and its software extension). Because such solution is incredible for the audit service, auditing is not realised in the demonstrator. This is consistent with the Technical Annex of the DIABCARD-TrustHealth Extension. In the future however, especially regarding secured PDC systems in general, this solution will have increased importance.

11.12 Directory Services

For both concepts of application and communication security, a Public key Infrastructure (PKI) must be established including directory services for public key certificates and revoked certificates (certificate revocation list, CRL) considering all certificates contained in each SC-PSE or SW-PSE including the certificates of the CA.

11.13 Access Control

First, the access to the Paradox Database and each DIABCARD system component as the DCC and the DCS including its interfaces are secured preventing unauthorised application usage. For that purpose, user identification and authentication applying the TrustHealth Health Professional Card (TH.HPC) with PIN protection have been implemented.

The COM port for PDC access of the DCS can be changed easily by altering the configuration file(s), and the serial communication port for HPC is selectable as well. To prevent bypasses, namely accessing the PDD or starting the DCS directly without authorisation, appropriate means of security are provided.

11.13.1Access Control to DCC

Concerning the DCC, an authentication dialogue requiring the TH.HPC and the correct PIN is applied. If a user has been authenticated this way, there is no further authentication dialogue for accessing the PDD or the DCS. Due to the lack of stored procedures or similar means in the PDD, it is not even possible to integrate TH.HPC authentication on database level.

11.13.2Access Control to PDD

For protection of the PDD, locking as well as cryptography-based mechanisms like database file encryption are possible.

The locking mechanism is applied for each table using the Borland Delphi IDE changing the table properties (exclusive access). A drawback with this concept is the limited range of effect. Only the tables that are currently open under the DCC are locked for third-party applications trying to get access to the database tables whereas the other tables are not locked. It is definitely not practicable to exclusively open all tables on start-up of the DCC.

In addition to this mechanism of locking the database tables that are currently open, confidentiality is applied to the database files preventing the interpretation of table data for unauthorised persons. The implementation issues for this cryptographic-based mechanism are described in detail in paragraph 9.4 dealing with confidentiality.

Moreover, the PDD files are integrity protected by detecting file changes or replacements. This is achieved by symmetric techniques encrypting a cryptographic check value (MD5-DES3-EDE2-CBC) calculated over the files. The encryption key is stored in a smartcard-PSE (SC-PSE) for security and management reasons. For file integrity, a different key is used as for file confidentiality minimising effects of key attacks.

For a higher level of security, a strong symmetric session key may be used that is changed after each operation (i.e. check the integrity of the DCS, trash the old session key, compute new session key, calculate MAC, store new key). However, the key is only changed on the current HPC and is thus not available for other users. Integrity checking is done before accessing the table data and re-calculation is performed after the closing the table data.

Alternatives enabling all security services mentioned are only achievable through replacing the database by a security-enabled one. This concerns also the accountability service keeping all database functionalities running. Appropriate databases are available on the market (e.g. Oracle 8) and should be taken into consideration for future implementations if required due to the threat and risk assessment.

11.13.3Access Control to DCS

Because the source code of the DCS product was not available, only limited security is implemented on this level. Since there should be no further authentication dialogue for accessing the DCS when authenticated to the DCC already, the DCC establishes a security context to the DCS by starting the server directly without a separate user interaction (means execution of the DCS).

In this scenario, the DCS is protected by integrity detecting program changes or replacements. This is achieved by means explained in paragraph 11.16.1.

Integrity checking is done after the authentication dialogue before starting of the server. To prevent a bypass starting the DCS without using the DCC confidentiality is be applied to the DCS files (see 9.4 dealing with confidentiality).

11.14 Accountability

Next, the responsibility of the Health Professional for data items has been realised enabling to determine the originator of the data as well as detecting any data manipulation. For that purpose, user-related digital signatures on data item have been implemented. Since there is no source code available for the DCS, the signature generation and verification process takes place inside the DCC (see Capter 11.13.1). Group-signing may be preferred to item-signing due to performance and memory storage reasons, but has not been considered due the lack of source code for the DCS and compiling problem of the DIABCARD Data Access API.

The DCC accesses the PDC (read/write) using the TCP/IP DCS-Access Service [Demmer et al., 1998] to pass card operating commands to the DCS and to get the results.

In general, the digital signature is generated and attached to the data values before writing them in the tables of the PDD or to the PDC. Verification of the signatures is performed after reading the items from the tables of the PDD or the PDC. Regarding the statements above, item-wise signing is preferred, but may lead to performance losses.

11.15 Authorisation

Restrictions are necessary for authenticated users concerning the acquisition and handling of medical data. Therefore, a detailed access control management has been implemented processing the *functional rights* (program functions) as well as the *data access rights* within a function like selection, creation, deletion, reading, writing, alteration of data and right management. This prevents unauthorised disclosure and manipulation of data, respectively.

Based upon user authentication each physician is only permitted to process certain functionality on the medical data she or he is allowed to access. Minimal authorisation was already available in the DCC so far only featuring two roles: user and administrator. The HPC provides roles, which have been implemented in the DCC, i.e. different, access rights for the various groups.

After authentication of the physician, authorisation inside the DCC is based upon the information stored in her or his authentication certificate and attributes. The existing "Security Level" in the DCC is used for this purpose. Moreover, the professional identifier connected with the items is adjusted to realise personal right management.

11.16 Confidentiality

Last but not least, confidentiality of the medical data has to be provided for application security preventing bypasses by accessing the data with other applications or tools that have no appropriate means of authentication as described in Chapter 11.9. All in all, there are three different levels where confidentiality may be applied: the PDD (database files), the DCS (files), and the DIABCARD Data Access API.

Since there is the demand for interoperability between the different DIABCARD test sites in Germany, the medical data on the PDC is not encrypted in any way. If encryption would be necessary in the future, confidentiality has to be placed on the level of the DIABCARD Data Access API. In this phase, confidentiality has been implemented on the level of the PDD (Chapter 11.13.2) and on the level of the DCS (Chapter 11.13.3) as follows.

11.16.1Confidentiality of the DIABCARD Server

As mentioned above, the DCS source code is not available and unauthorised start-ups cannot be prevented. A solution is to decrypt all files for the DCS in the DCC after the authentication dialogue has been passed successfully. Then, the server is started automatically

before the DCC appears. After closing the DCC, the DCS files are encrypted again. The DCS files are encrypted if no (authorised) user is logged in the DCC. Encryption and decryption is performed with a strong symmetrical key (DES3-EDE3-CBC, 168 bit) that is stored in the SC-PSE of each user for security and management reasons. The PSE is opened when the user has been successfully authenticated using the HPC. For file confidentiality, a different key is be used as for file integrity minimising key attacks.

For a higher level of security, a strong symmetric session key may be used that is changed after each operation (i.e. decrypt the DCS, trash the old session key, compute new session key, encrypt the DCS, store new key). However, the key is only changed on the current HPC not available for other users.

11.16.2Confidentiality of Paradox Database Table Data

Regarding the statements of Chapter 11.13, confidentiality is applied to the whole PDD encrypting the files belonging to each table. These are `.DB` for the Paradox table, `.PX` for the primary index of the Paradox table, `.XGn/.YGn` for the composite secondary index of the Paradox table. As explained above, a strong symmetric key is used for encryption/decryption stored in the SC-PSE. A different key is used for file confidentiality and integrity providing a higher level of security. Key changing as mentioned above may be performed, too. All database files are encrypted if there is no active connection by the DCS and decrypted after the user has been successfully authenticated to the DCS. It is possible to perform decryption/encryption for each table on demand (preferred), i.e. each single table is decrypted if accessed and encrypted if released. This is not yet practicable due to performance reasons, but gives a very high level of security protecting e.g. tables which are not opened (and therefore not locked) during the runtime of the DCS.

11.17 Audit

In sensitive environments, an audit checking the accountability including non-repudiation for any procedures is an inevitable functionality. Such an audit must be provided in a secure environment disabling any manipulation by the audited person or any unauthorised third party.

11.18 The Advanced DIABCARD Security Solution

Generally, the first implementation of the secure DIABCARD solution has been a success story. Meeting the genuine challenge of the European Commission, the feasibility of PDC applications based on the European security infrastructure has been demonstrated in 1999 in Magdeburg as well as in Spring 2000 in Munich.

In the context of new developments on both the technical and the legal field, however, a strong challenge has been discovered to improve the existing solution in order to meet the needs of health networks and to overcome the disadvantages of current installations mentioned. Still within the DIABCARD project framework, an advanced DIABCARD PDC has been completely specified and partially implemented in the sense of a feasibility study.

This advanced DIABCARD PDC contains a cryptographic co-processor by that way enabling all the services for communication security and application security provided by the deployment of cryptographic algorithms.

As another way assigned to be deployed in a German project for facilitating management and quality assurance in kidney transplantations, the German HPC specification [HCP-Protocol, 1999] defines already future interactions between HPC and PDC using card-verifiable certificates (see Chapter 11.19.2). By that way, some of the current application-mediated security services might be delegated to the card PSE.

11.18.1Additional Security Services of the Advanced DIABCARD

Exploiting key-related security services enabled by the advanced DIABCARD, strong authentication of the PDC against the DIABCARD workstation as well as access control services are provided. So, the advanced DIABCARD PDC enables also the access control services (step 1d.) and (step 1e.) shown in Figure 11.2.

Adopting the structure of data stored, storage capacity of the PDC and infrastructure of the DIABCARD environment, accountability services could be extended to the PDC including not only the processes of creation and updating of data, but also the processes of card-system interactions.

In the same context of key-related security services, integrity and confidentiality of data on the card level are included after establishing the environment needed. The audit of these card-related interactions will be realised in the future.

11.18.2Advanced Application Security Services

The misuse of the sensitive medical data by unauthorised persons invoking the database independent of the DIABCARD application must be excluded. To support multi-user facilities of the data in the database encrypted for providing confidentiality, the installation of a ticket server distributing a symmetric session key to users authorised through the strong authentication procedure based on the HPC identity key has been specified and tested as way of choice. This advanced solution is also applicable to general EHCR or other systems. Also the audit functionality has been combined with the ticket services and can be red only by the security administrator authorised through his/her smart card based certified strong authentication.

11.19 The DIABCARD Integration in Health Networks

Within the framework of the Bavaria Online Initiative of the German federal state Bavaria, this aforementioned advanced DIABCARD will be implemented in large scale demonstrators and evaluated during the next two years as one part of a Bavarian health network. This card will meet the challenges of enhanced security services such as strong authentication of the card holder, digital signature, and cipher functions managed and performed by three card-based asymmetric key pairs in addition to the former Patient Data Card services described above. Using for example the digital signature, the patient's consent can now be provided and verified electronically.

11.19.1The Next Generation DIABCARD Patient Data Card

As already indicated, several restrictions had to be taken into consideration during the process of designing and implementing the former DIABCARD PDC back in 1997/1998. On the one hand, the card has not been able to provide all the required security functions at this time. On the other hand, the application itself could be protected only by using a secure "shell" instead of securing the provision of services directly. Regarding security requirements, DIABCARD had to decide about an advanced strategy.

Starting with the advanced DIABCARD described above, the next generation DIABCARD will incorporate the functionality of both a Patient Identity Card (PIC) and a PDC. The functions of a PDC have been explained in detail already (see Chapter 11.3). On the other hand and from a security standpoint, a PIC shows similarities to the HPC card type described in Chapter 11.7. Thus, the holder of such a PIC is able to provide all security services required, as identification and authentication, digital signatures, and cipher functions. These services have been specified already so the new approach is able to completely follow existing European and national standards of smart cards.

As both the German legislation and the European regulations and recommendations require an evaluated card for enhanced security services, the DIABCARD PIC will follow their requests using certified products only. By the way, the specification of a new data card for diabetes patient could be seen as an approach towards a new generation of patient cards in Germany, as there is a strong need to improve the current health insurance card called "Krankenversichertenkarte" (KVK). The current version has been designed to store only administrative data (name, address, insurance number, insurance company, etc.). A second generation KVK needs to have several medical information items on card. Beside an emergency data set, the data structures might contain information about allergies, about specific medication, or about infectious diseases including HIV, up to a minimised version of an electronic patient record.

As said, several aspects have to be taken into account when designing the new DIABCARD PDC. The aforementioned legal situation in Europe will be considered. European law has become national law in most of the member states. The remaining states will follow soon. Nevertheless, in some countries national regulations exist with even higher legal demands which have to be followed. One example is the recent debate of the "German Digital Signature Law" versus "European Electronic Signature Standard Initiative". Germany has already defined a higher level of security requirements within its Digital Signature Law and the related act than the European Council. A process of adaptation is required here to achieve both a technical and an administrative interoperability. Nevertheless, the "Qualified Electronic Signature" which is equivalent to the former German legislation will be used in the sensitive healthcare environment only.

On the other hand, the growing need of patients to be properly informed to use their own right of self-determination can be discovered. Along with a growing mobility of people within Europe, there is a requirement for an extended electronic data exchange between different organisations directly or indirectly involved in the process of patients' care. Summarising these aspects, the new card has to cope both administrative data and medical data. It has to cover several interoperability aspects mentioned to make sure that it can be used in different countries. It has to fulfil more functions than being only a small part of a specific disease-related data set. In other words: DIABCARD is looking for a multi-functional smart card specification in a way that a combination of a Patient Identification Card and a Patient Data Card comes true.

This means furthermore that the new DIABCARD PDC needs to be able to offer several applications in parallel. From the card specification point of view, several "isolated" partitions on the card are required. As one can imagine each card-based application may need different security requirements and may of course thus have a different security policy at all. That's why each card application has to be accessable separately.

Taking both the memory card specification of the former DIABCARD PDC and the new requirements and conditions mentioned above as well as the availability of new card products into account, the new DIABCARD PDC will contain

- an emergency data set following European recommendations,
- the diabetes-related specific data set,
- the diabetes passport,
- an information data set about card holder (patient),
- and security-related functions including keys and probably certificates.

The first three information structures have already been specified by a former DIABCARD project phase as well as by other projects and initiatives (e.g. the G7 group, the WHO, etc.).

In the following, the application parts of the DIABCARD PDC will be described in a more detailed manner. As far as security functions and the data about the card holder are concerned please refer to the Health Professional Card section earlier in this paper.

In order to make such a card respectively specific parts of it readable by almost everyone interoperability aspects play an important role. This is especially true for the emergency data set because the new DIABCARD PDC needs to set up this service in a way that it is possible for everybody to get access in a case of emergency or even at home just to see what's on the card. Thus, this emergency data set has to be provided to all users without secure access means. There will be a need for just a parser or browser to read the data without having the chance to delete or update anything. The only security function requested is an integrity check of the data. Based on a successful data integrity check all persons and organisations can use the data being sure they are unchanged and thus valid.

When it comes to disease-specific data stored on the card, the new DIABCARD PDC will use both the diabetes passport and the diabetes data set specified in an earlier phase of the DIABCARD project. Taking the new cards with up to 32k RAM into account instead of the 8k cards used for the pure PDC version one can imagine that even here the data set definition can be extended. Nevertheless, the way these diabetes-related data are stored on the card is completely different from what has been said about the emergency data set. It is a must that the diabetes data set is accessable only by using the Health Professional's HPC and presenting a Personal Identification Number (known as PIN) as well as using the HP's appropriate attribute certificates. As far as the PIN itself is concerned the new card has to provide not only 4 digits but in minimum 6 to 8 digits, preferably alpha-numerical (similar to the difference between password and passphrase). Only by having successfully presented the authentication PIN to the card, an HP is able to get secure access to patient data. But this means new security requirements also towards the card reader used, as most of them do not even have a keyboard; and if yes, then it is a keyboard with digits 0 to 9 only.

A PIN is of course not an ideal solution for the issue of a secure identification and authentication process towards an application. It is simply a combination of possession (the card) and knowledge (the PIN). But as one can imagine such a PIN could easily be forgotten or even be "mixed up" with other PINs for e.g. a cheque card, a credit card, a debit card, etc. The only way to solve this problem is the substitution of this "knowledge" by another type of "possession": biometrics as, e.g., fingerprint, iris, face, voice or similar.

Last but not least it is the mechanism how to get data on cards that makes a difference. At the moment, DIABCARD is using a very specific application to read and write the aforementioned diabetes data in sequential records. This means, only the complete data record can be read or written. This is a typical disadvantage of nowadays card application solutions. Future cards should allow installing a relational or an object-oriented data base on the card so the HP is able to read and write data directly from and to data base thus enabling a technical access to data as easy as possible. A so-called Smart Card Query Language (SCQL) is already in use and could be extended to a real means for accessing card data. Along with this query language, both the data base definition language and, of course, the card operating system has to support direct access to any data on the card following the application security policy.

11.19.2Alternative Solutions for Access to Cards

As mentioned already, the German specification for an Electronic Doctor's License smart card ("Elektronischer Arztausweis") [Arbeitskreis, 1997] is proposing also a cheaper low-level and easy-to-use mechanism for a card-card interaction between HPC and PDC by the use of card-verifiable certificates (CVC). Herefore, only two security services are used. Firstly, the PDC has to prove its authenticity. Secondly, the HP – using his HPC – has to

prove his related access rights to read, write, or update PDC data. The authorisation procedure is therefore often based on attribute certificates.

When proving these access rights, an authentication procedure has to be performed so that in the PDC the related security status can be set. Using a symmetric algorithm, the group key needs to be successfully presented before any access to PDC data items is allowed. Assuming the use of asymmetric algorithms for the DIABCARD PDC access, the certificate holder authorisation certificate C.HP.AUT-CV needs to be successfully presented.

The authentication certificate is used in PK-based authentication procedures applied in any HPC-PDC interoperation. The principle structure of the card verifiable certificate used is shown in the subsequent figure. The sequence of data elements can be described by a headerlist as defined in ISO/IEC 7816-8 [ISO/IEC 7816-8]. This requires nonetheless a fixed length of each data element [Blobel and Pharow, 1997].

Certificate Content	Certificate Profile Identifier (1 B)	Certification Authority Reference (8 B)	Certificate Holder Reference (14 B)	Certificate Holder Authorization (x B)	OID.PK (x B)	PK (modulus tag '81', exponent tag '82') (x B)
Headerlist Content	'5F29 01'	'42 08'	'5F20 0E'	'5F4B 0x'	'06 0x'	'5F49 xx' || '81 xx || '82 xx'

Figure 11.3: Certificate Content and Certificate Headerlist

Hereby, the "Certificate Profile Identifier (CPI)" has the purpose to denote the exact structure of the CVC. It can be considered as an identifier of the card's internal headerlist describing the concatenation of the data elements including their length so that, e.g., the PK in the CVC can be found by the certificate verifying card (PDC). The "Certification Authority Reference (CAR)" has the purpose of identifying the certificate issuing CA with a distinguished name (DN) in such a way that it can be used as an authority key identifier for referencing the PK to be applied for the certificate verification. Therefore, the CAR consists of

- the CA name (country code according to ISO 3166 [ISO 3166] (2 Bytes, e.g. DE = Deutschland) followed by an acronym of the CA (3 Bytes, ASCII characters), an
- an extension for key referencing (3 Bytes).

The "Certificate Holder Reference (CHR)" has the purpose to denote the certificate holder uniquely in such a way that its DN can be used as a subject key identifier for referencing the PK of the certificate holder. The CHR thus consists of

- a CA Reference CAR (5 Bytes) || Extension for key referencing (3 Bytes), if the certificate holder is a CA, or
- the serial number of the card's processor chip (ICCSN, 14 Bytes), if the certificate holder is the card of a Health Professional.

The "Certificate Holder Authorisation (CHA)" has the purpose to denote the access rights of the Health Professional with respect to data stored in files in the patient data card. The meaning of CHA can be compared with a role based group key when applying symmetrical algorithms. The CHA consists of

- a prefix denoting the entity assigning the role ID, and
- the role identifier of the Health Professional.

Figure 11.4 shows CHA Role Identifiers relevant for physicians.

CHA Role ID	Meaning	Relevant for C.CA.AUT-CV	Relevant for C.HP.AUT-CV
'00'	No access right to data	x	
'01'	CHA Role ID for proving the access right of a physician		x

Figure 11.4: CHA Role ID Coding

The PK in a certificate consists of a concatenation of parameters. These parameters, which have a context specific tag, have to be coded as octet string. In the CVC verifying entity (i.e. in the PDC) the occurrence of such a parameter and its length can be described in the headerlist (Figure 11.4). The data to be signed are the certificate content. The hash function used and the digital signature input format are denoted by the object identifier (OID).

Summarising the intentions of a CVC and the related security services that can be provided, the new DIABCARD approach will strictly orient on a PDC with a cryptographic processor so that, e.g., full service signatures and strong authentication can be performed by the card.

11.20 Summary and Conclusions

By our knowledge nowhere else done in Europe, security services based on the European security infrastructure of Health Professional Cards (HPC) and related Trusted Third Party (TTP) services have been introduced securing Health Professional (HP) workstations dealing with chipcard-based health information systems. The basic application security services as authentication, authorisation, access control for users and their accountability as well as integrity and confidentiality have been implemented. Additionally, communication security services securing the communication between the workstation and a related departmental information system have been provided using secure FTP. Annex D gives a detailed description of this first European solution securing a PDC environment via HPCs.

At PDC side, CVC are discussed. For implementing advanced PDC security solutions, PKI-based certificates for patient's should be introduced enabling all security services discussed in the Health Professional context such as strong authentication, digital signature, and encoding/ decoding, however.

At this early stage, the DIABCARD applications are not yet prepared to deploy security services. Therefore concerning specific issues, the implementation is sometimes still a proprietary one.

12 A Future-Proof Concept for Distributed Intelligent Health Information Systems on the Internet

According to the generic component model (Chapter 4), all views, information content, functionality, implementation environment, and underlying technology but also the proper level of granularity might be modelled in a consistent way. In this way services and complexity of the running application component can be defined according to the application environment and the user needs. Services concern entry, processing, and presentation of data but also the enforcement of underlying policy for communication and co-operation. The generic component model enables claims change management (viewpoint of the system) and the resolution of the component's complexity by the transition to less complex sub-components as shown in Figure 4.6. Each specific model in the abstraction-granularity space reflects one specific archetype.

12.1 Design of Future-Proof Health Information Systems

Talking about a system's design, the specification of the system's structure and functionality must be provided. Furthermore, the specification and handling of data as well as the invocation of services at runtime has to be defined. Finally, the runtime environments must be specified and supported.

Within the components' appropriate level of granularity mentioned above, the platform-independent model of the system considered will be specified using UML. Additional descriptions of, e.g., behaviour and constraints can be defined using either UML or natural language.

To make the special views on the system (platform-specific models) visible and graphically manageable for lovers of abstraction, UML diagrams using an officially adapted platform-specific profile should be deployed. The second way using interface definitions in a concrete implementation technology (OMG, IDL, XMI, or Java) should be exploited rarely. The definition of behaviour and constraints expressing concepts and knowledge should be performed by ways preferred by common users. Therefore, the description of the components according to equation (4) can be established in archetype schemas using the XML standard set. Related to the granularity and technology viewpoints, mobile computing has to meet special requirements which are easily enabled by this dynamic selective approach of the proper state of the complex system.

The walk through the different RM-ODP views depends on the requirements the application has to meet. In that context, defining specifications and influencing specifications must be distinguished. Thereby, instantiations of models may change the constraints to be considered in the defining model which must be adapted recursively.

The specification of data, their import, export, and exchange should be performed using a technology-independent, long-term standard. Currently, the gold standard seems to be XML. The binding of functionalities and data will be performed using component certificates which are digitally signed.

12.2 Basic Packages of Future-Proof HIS

According to the Generic Components Model (Figure 4.8), specific components have to be aggregrated to packages related to requested services following specific concepts or strategies. Packages or aggregations of them enable the logic for business processes (business logic).

The basic packages of platform-independent models are shown in Figure 12.1. At the one hand, the archetype package is a recursive component of the architecture itself; at the other hand, it is a description of the concept and its underlying constraints. Therefore, some authors refer to archetypes as defining parts of semantics (in the figure asterix-labelled), other interprete them as rules and logic.

- Support package
  - External package
  - Mapping package
- Content-related package
  - Spatial package
  - Temporal package
  - Rule-related package
    - Logical package
    - Legal package
- Management package
  - Specification package
  - Revision and versioning package
  - Execution package
  - Navigation package
  - Status package
- Communication package
  - Extract package
  - Transaction package
- Semantic package
  - Archetype Package*
  - Data type packages
    - Basic
    - Text
    - Quantity
    - Date/time
    - Time specification package
    - Encapsulated data
    - Link data
    - Identifier / label data
  - Terminology package
- Containment package
  - Autorisation package
  - Decision package
  - Audit package
- Logic package

Figure 12.1: Basic Packages of Platform-independent Models

Knowledge-based active and/or interactive component systems have to follow constraint models (archetypes). For operational systems, at least some of the aforementioned basic packages have to be specified and implemented. Therefore, the semantic package at least containing data types, the support package to enable import of data from other resources inclusive queries via interfaces as well as mapping to other environments, the communication package allowing for export of data, and finally the semantic/logic package ruling constraints must be available to the management package. For the clinical practice guideline demonstrator (Chapter 12.8.1), these basic packages will be explained referring to the corresponding XML documents.

12.3 Tools Needed for Specifying and Running Future-Proof HIS

For specifying and implementing HIS, or especially, advanced EHR systems, tools have to be developed enabling modelling, managing, implementing, and maintaining the components needed. Such tools start with the UML expression of components, followed by the transfer of such graphical model into XML schemata. At runtime, these XML schemata will be instantiated according to the model instances reflecting both, the different domain models (constraint models) and the object model. The resulting schema must be translated into a runtime environment. Summarising the aforementioned definitions, at least following tools must be provided:

- Tools for graphical modelling of systems and components;
- Platform-specific[43] profiling tools;
- Tools for managing models and their schemata;
- Editors for authoring domain models, validators for interpreting them to create data, and browsers to use those data;
- Tools for mapping different languages (e.g. XML↔IDL, XML↔value, XML↔Java);
- Services for tailoring and managing the resulting components.

12.4 Meta-Model Transformation

Usually, analysis and specification of information systems is provided using advanced abstraction tools such as UML mentioned already in Chapter 4.1. Based on object-oriented meta-semantics, class models can be deployed for deriving XML vocabularies automatically. To bridge the gap between different graphical languages such as modelling languages like UML and the Meta-Object Facility (MOF) Specification but also other languages expressing syntax and semantics of systems, a unifying methodology must be introduced. UML has been introduced already, MOF defines a set of CORBA IDL interfaces that can be used to define and manipulate a set of interoperable metamodels and their corresponding models. The MOF provides the infrastructure for implementing CORBAbased design and reuse repositories. The MOF specifies precise mapping rules that enable the CORBA interfaces for metamodels to be automatically generated, thus encouraging consistency in manipulating metadata in all phases of the distributed application development cycle. The method of choice for expressing syntax and semantics of systems at any level of abstraction is the XML standard set. Figure 12.2 shows a general schema for representing complex system architectures using OMG's XML Metadata Interchange (XMI) standard [Jeckle, 2001]. The XMI specification supports the encoding of metadata consisting of both complete models and model fragments, as well as tool-specific extension metadata. Besides bridging different presentation languages and enabling the vocabulary generation by using XMI, the XML mapping specifications harmonise design, specification, and implementation environments.

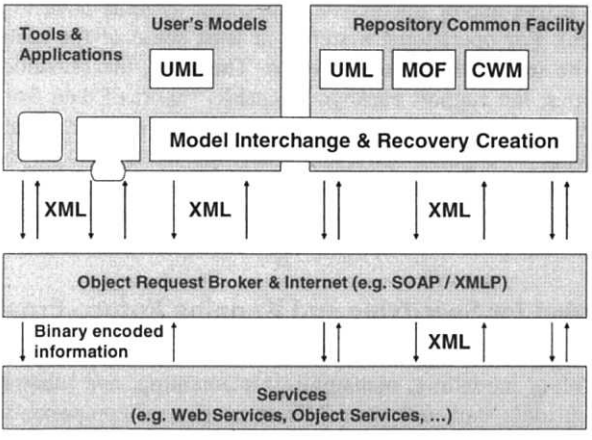


Figure 12.2: XML-Centric Architecture (nach [Jeckle, 2001])

[43] The term "platform" is used in a very generic way again.

XMI offers the generation of DTDs or XML schemata from UML/MOF models. Because XSD can express a schema for an XML schema, it can be used backwards for documenting the output of XMI processing and transposing it as XML input. For this purpose, an extended hierarchy of built-in types in XSD starting with a root type "anySimpleType" can be enriched by user-defined types or schemata as introduced in Chapter 5.1.4. The algorithm for transforming the output format at metadata level into the input format at schema level can be implemented using XSL transformation (XSLT).

The proposed way replaces proprietary tools such as HL7 RoseTree® by a generic straight forward methodology based on open standards.

12.5 HARP Based Implementation Tools

Within the HARP project, partners from Greece, Germany, Norway, United Kingdom, and the Netherlands have specified, developed and implemented the HARP Cross Security Platform (HCSP) for Internet based secure component systems as well as the development methodology and the development tools needed.

The HCSP is composed of:

- A *client environment* which is fully under server control and accessible only to principals holding the appropriate smartcard.
- An *application (central) server* as core of the server-centric approach. User tasks are delegated to servlets; therefore an application server must also host a web server.
- A *Web server* as 'entrance'-point for the user.
- A *policy server* providing policies and policy related functions.
- An *attribute certificate server* providing and managing attribute certificates.
- A *database server* storing all medical data. Control of access to data is policy-regulated.
- An *archive server* storing all messages communicated for accountability reasons.

For more details about HCSP see Chapter 12.7.

Regarding the implementation of an open, flexible, distributed, component-based architecture, important tools are those for mapping the concept specification and information exchange language/format with data specification formats, interface definition languages, legacy environment specifications, programming languages, etc. Using the example of HARP's component architecture, Figure 12.3 demonstrates HARP's implementation tools for the typical Internet environment. Therefore, Java has been used resulting in Java components (Java applets or servlets) which run on the systems communicating and cooperating. In this chapter, only components relevant for application functionalities will be discussed. The other components relevant for security services will be explained in detail in Chapter 12.7.

The HARP XML Data Translator Component (HXDT) is used on the server side as an XML interface component between the client applet and the servlet-based server. As main components simply speaking, it establishes an XML editor and XML parser. From the technical point of view, the HXDT consists of a package or a simple set of classes that provides two kinds of services [HARP_WWW]:

a) the functionality for extracting information from the XML document sent by the client applet. Specifically, it extracts information about the fields that must be updated in the database and the respective new values. The HXDT does not translate the XML document into SQL code; however, it provides all the necessary information in order to help the servlet side creating this code.

b) the functionality for creating an XML document containing the necessary information that has to be sent to client by the servlet-based server. The component is responsible for gathering all information provided by the server concerning database fields, respective values, read/write access rights etc. It constructs the appropriate XML document that will be sent to the client.

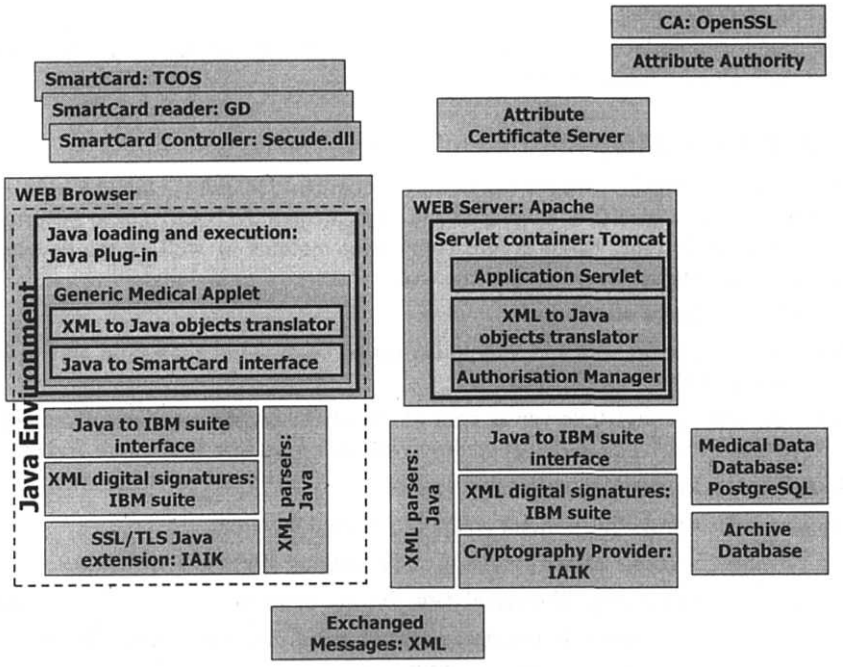


Figure 12.3: HARP Components for Generic Secure, Distributed Applications on the Internet [HARP_WWW]

The following paragraph describes some technical details concerning the main class (object) used, input output parameters/objects etc. which have been elaborated by the HARP partners of NTUA Athens (Greece) and Fraunhofer Institute FOKUS (Germany) with a modelling basis originated by the author. The specification uses the Java programming language for showing the component's functionality in technical terms. Defining the HARP demonstrator implemented at the Medical Informatics Department in Magdeburg, the specification might be changed for other purposes in different environments.

In both aforementioned cases, the servlet instantiates an object of the *XMLDataTranslator* class:

XMLDataTranslator dataTranslator=new XMLDataTranslator();

For extracting information from an XML document sent by a client to a server, the servlet calls the *parse* method of the *XMLDataTranslator* class in order to initiate the parsing procedure. The *parse* method has as input argument a *java.io.OutputStream* object that provides the XML document to be parsed. The method's return type is a *boolean* value indicating the success or failure of the parsing procedure. The *parse* method has the following declaration:

public boolean parseXMLDocument(OutputStream xmlDocumentStream)

Parsed data are encapsulated in an internal structure of the *XMLDataTranslator* class. The servlet accesses them via the *getFieldData* method. The method has the following signature:

public FieldData[] getFieldData()

FieldData is a custom helper class holding all the data regarding the SQL update procedures. The class functions as a three-dimensional or a more dimensions array.

The class provides the following methods.

public boolean hasMoreFields()

public void next()

public String getFieldName()

public String getFieldValue()

public int getValueType()

public String getFieldDataName()

The functionality provided by the aforementioned methods is obvious. As far as the last method is concerned, the return value will indicate the data type of the respective field value. Standard field types that map to respective SQL types (static Java constants) must be defined. For example, the *FieldData* class may contain the following declarations:

```
static int INTEGER=1;
static int FLOAT=2;
static int DOUBLE=3;
static int STRING=4;
static int DATE=5;
```

A sample code that can be used for the field/value retrieval procedure is the following:

```
String fieldname, fieldvalue;
while(fieldData.hasMoreFields()) {
        fieldname=fieldData.getFieldName();
        fieldvalue=fieldData.getFieldValue();
//      Execute the appropriate SQL update procedure
        fieldData.next();
}
```

An *XMLDataTranslator* object can contain one or more *FieldData* objects. A *FieldData* object may contain other *FieldData* objects too. Therefore, some Field Data objects can represent data categories (or empty tags in XML) whereas others contain actual data.

For creating an XML document to be sent by a server to a client, the servlet will instantiate an object of the *FieldData* class and associate it with the *XMLDataTranslator* object instance. The association of the two objects is done via the *setFieldData* method of the *XMLDataTranslator* class. The method has the following signature:

public void setFieldData(FieldData fieldData)

The following code snippet shows the instantiation and association steps:

```
FieldData fieldData=new FieldData(String fieldDataName);
dataTranslator.setFieldData(fieldData);
```

When used as a helper class for constructing XML documents, *FieldData* class will provide the following methods for entering field/value pairs:

public void next()

public void setFieldName(String fieldName)

public void setFieldValue(String fieldValue)

public void setValueType(int valueType)

After all necessary data has been entered; the *createXMLDocument* method of the *XMLTranslator* class should be called. The method has the following declaration:

public OutputStream createXMLDocument()

The XML document is received via a *java.io.OutputStream*.

Based on the principles presented, the HARP partners from National Technical University Athens developed first versions of appropriate tools administering servlets and defining XML schemata being mapped to Java clients' GUIs. Figure 12.4 demonstrates the NTUA Administration Tool.

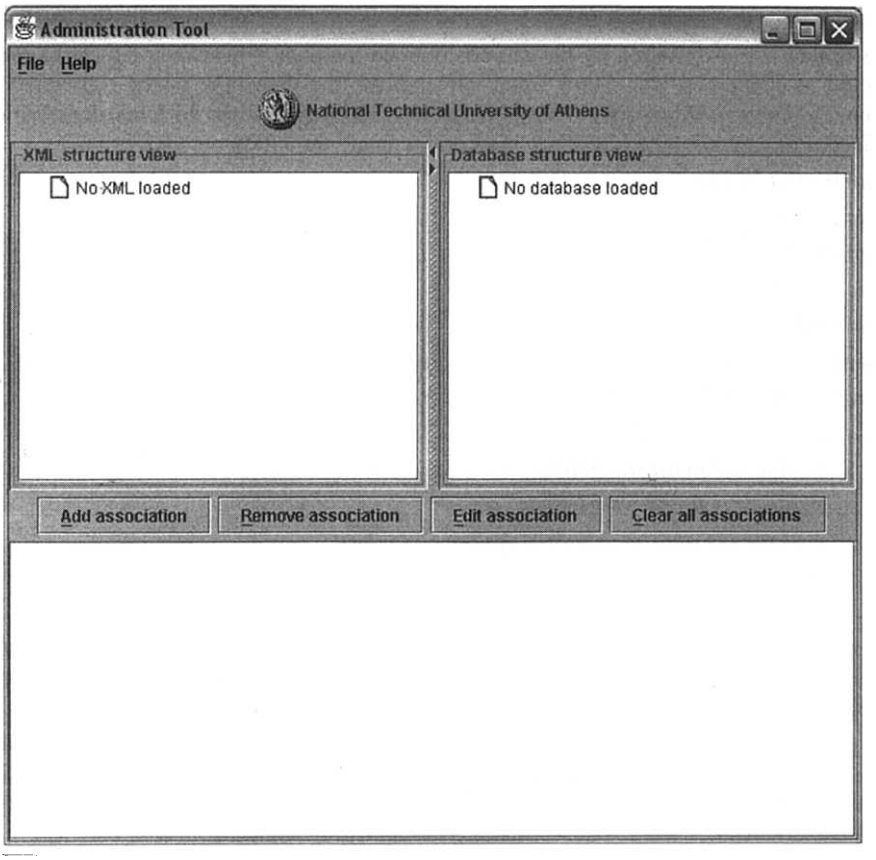


Figure 12.4: HARP Administration Tool [HARP_WWW]

Another tool serves to specify applets needed. It enables loading and editing of XML schemata to be used to create applets by translating the fixed XML scripts. Therefore, viewers for XML schemata as well as the resulting client application (GUI) have been integrated. The next figures give examples of using the HARP tools for establishing an Internet-based Clinical Studies application as a special kind of an advanced EHR.

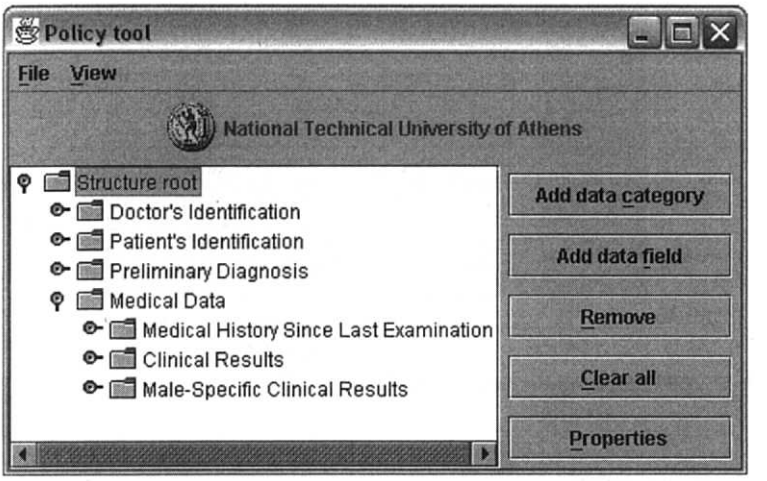


Figure 12.5: HARP Policy Tool Applied for Defining a Clinical Study Applet

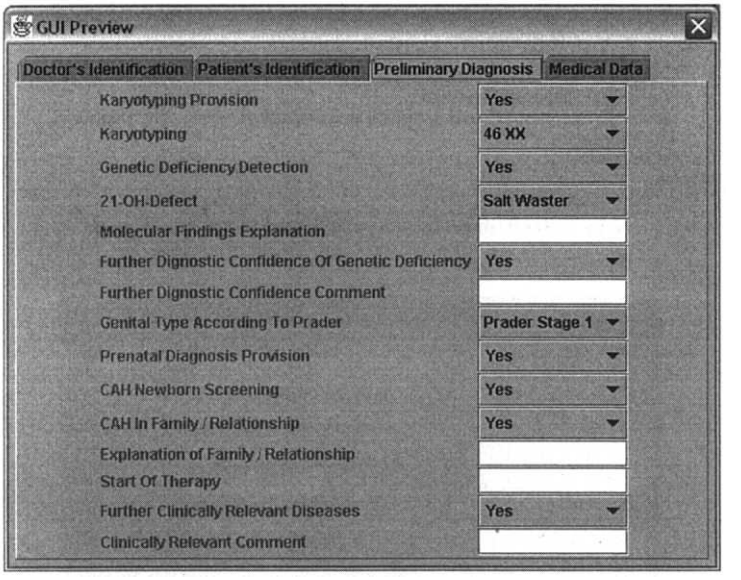


Figure 12.6: Examples of Clinical Study Applets

The automatic generation of applets as client components exemplified in Figure 12.6 is based on the systematic and automatic specification of XML schemata. Instantiating this schema, the XML specification is transferred into a Java applet at runtime. Figure 12.7 shows the generic secure interoperable applet. By that way, virtual applications occur.

Figure 12.8 and Figure 12.9 demonstrate both an XML message and the Java applet based GUI provided by the HXDT discussed above.

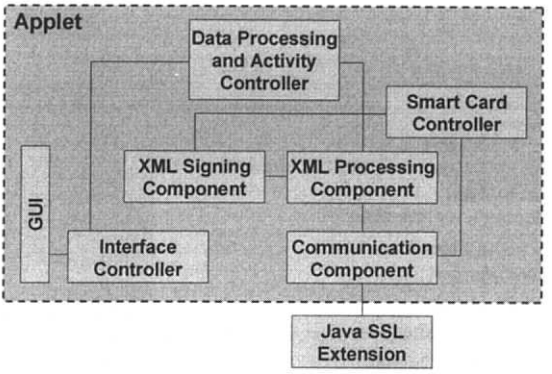


Figure 12.7: HARP Generic Applet Architecture [HARP_WWW]

```
<?xml version="1.0" encoding="utf-8" ?>
<reply status="OK">
  <PatientData>
    <Identification>
      <OrganizationId visible="true">14565</OrganizationId>
      <OrganizationName>Org. Name</OrganizationName>
      <PatientID>3456</PatientID>
      <UniversalPatientId>5678</UniversalPatientId>
      <PatientDateOfBirth>11/1/1980</PatientDateOfBirth>
      <PatientSex>Male</PatientSex>
      <PatientMultituplets>Second in the set of multituplets </PatientMultituplets>
    </Identification>
  </PatientData>
</reply>
```

Figure 12.8: XML Message Instantiating a Java Applet Shown in the Next Figure

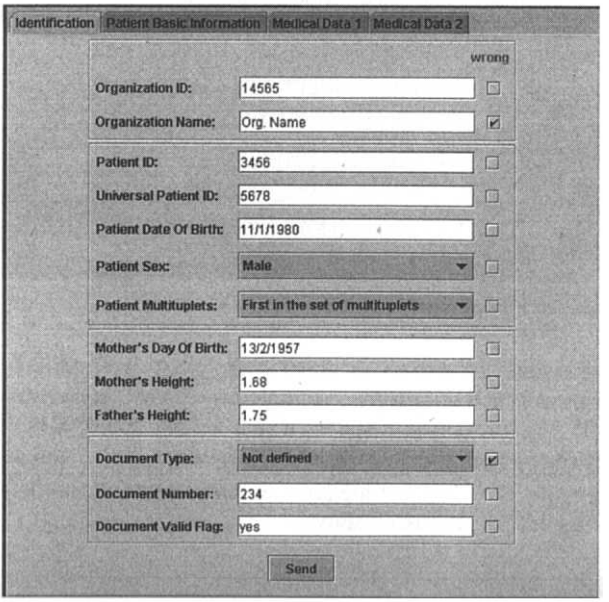


Figure 12.9: Java Applet Instantiated by the XML Message a Shown in the Figure Above

The servlet engine, which provides the specific XML schema according to the special functionality required, is connected to a policy server and the database serving with the data at instantiation of the schemata. The output is communicated via a regular Web server as shown in Figure 12.10.

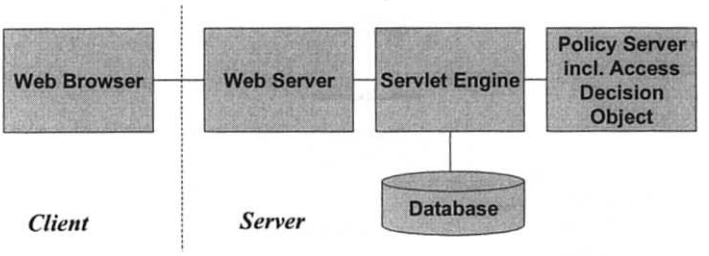


Figure 12.10: Generic HARP Architecture

12.6 The HARP Clinical Study Demonstrator

For demonstrating structure and functions of the HCSP as well as HARP's generic component architecture, a clinical studies application has been designed and implemented based on the aforementioned principles. Not dealing with the study design, a running study from the paediatric endocronilogy domain has been used. The application has been completely modelled using Rational Rose®. Starting with use case diagrams, sequence, activity, and package diagrams have been developed as shown in the following figures.

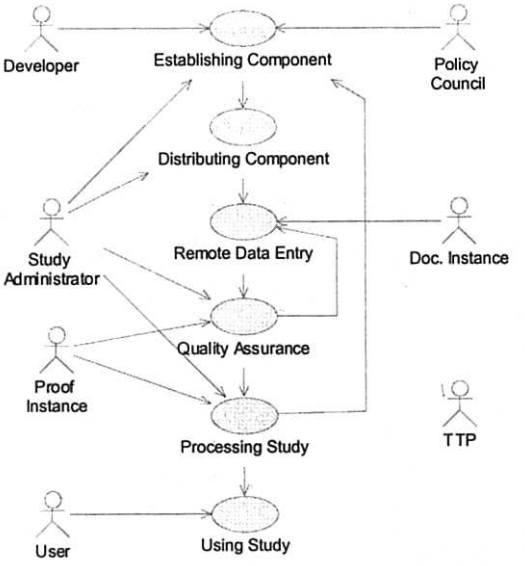


Figure 12.11: Clinical Study Use Case Diagram

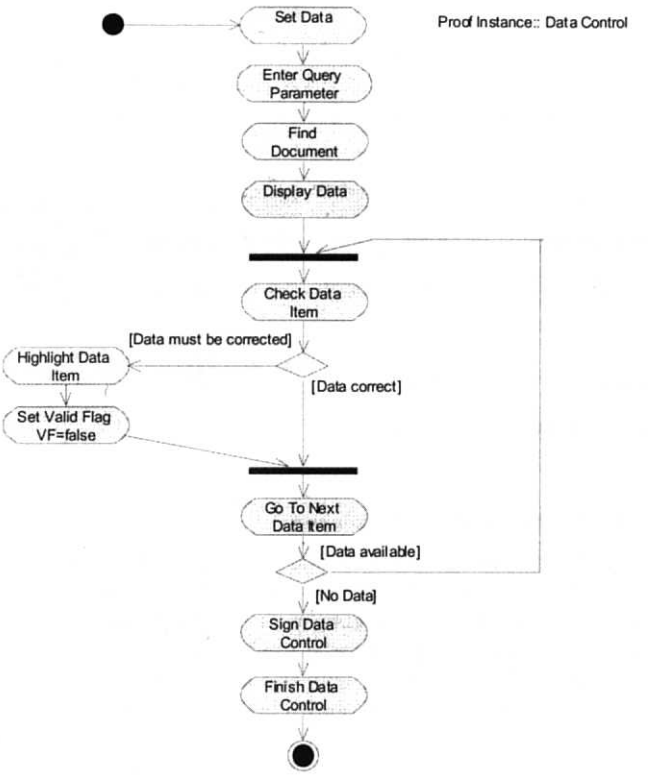


Figure 12.12: Clinical Study Activity Diagram Example

Figure 12.13 shows the resulting Clinical Study applet and Figure 12.14 presents the related package structure of the clinical study application.

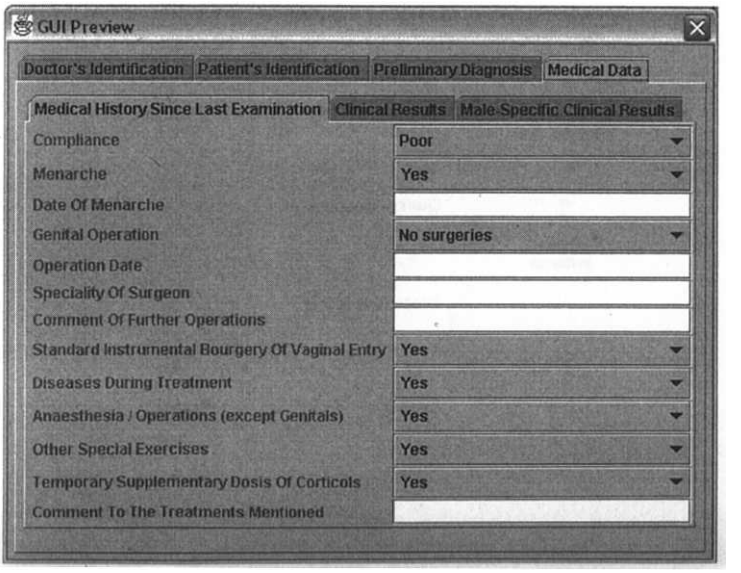


Figure 12.13: Examples of Clinical Study Applets

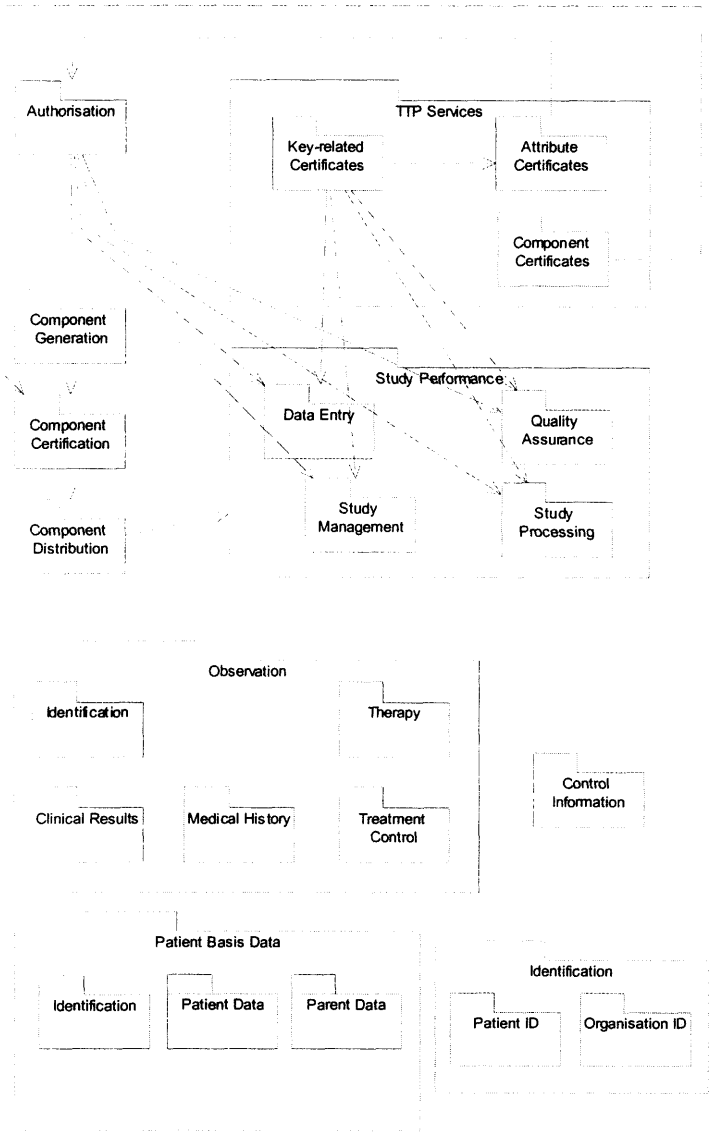


Figure 12.14: Package Diagram of the Clinical Study Application

12.7 HARP Cross Security Platform

12.7.1 The Need of Policy Enforcement

Implementing a security infrastructure mostly means to only provide communication security services as described in Chapter 6.4. The policy negotiated and agreed upon is mostly fixed in paper-based policy statements. Communication partners strongly authenticated

have to trust the other side in meeting the policy requirements if it uses the information exchanged. This is an important weakness of sensitive information systems, however. Application security services are managed (hopefully according to the harmonised policy) by the receiving side. Therefore, security breaches cannot be excluded, but they could be judged if they have been recognised. At worst, insiders perform security breaches having certain rights and extended knowledge about dealing with the system, but not having the rights they claim. Therefore, the enforcement of security policies regarding rights and duties of communication partners is an essential challenge for distributed interoperable information systems.

12.7.2 HARP Cross Security Platform Specification

The HARP project introduced in Chapter 12.5 does not only present a new architectural approach for component-based information including their implementation, but it offers also an enhanced security services environment including security policy enforcement.

To provide platform independence of solutions in HARP as a real three tiers architecture, the design pattern approach of developing a middleware-like common cross platform called HARP Cross Security Platform (HCSP) has been used. In HCSP, platform-specific security features have been isolated. Using an abstraction layer, communication in different environment is enabled. According to the component paradigm, an interface definition of a component providing a platform-specific service specifies how a client accesses a service without regard of how that service is implemented. So, the HCSP design isolates and encapsulates the implementation of platform-specific services behind a platform-neutral interface as well as reduces the visible complexity. Only some minor specifications have to be rewritten for each platform. The solutions concern secure authentication as well as authorisation of principals even not registered before, deploying proper Enhanced TTP (ETTP) services [HARP_WWW]. Especially, it helps to endorse policies by mapping them on processing components. Figure 12.15 demonstrates the HARP ETTP compared with a traditional TTP

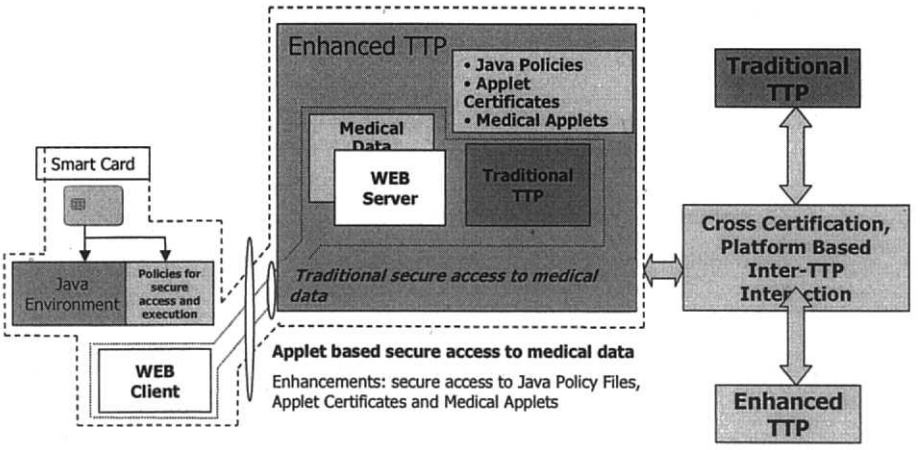


Figure 12.15: The HARP Project's Enhancement of TTP Services [HARP_WWW]

HARP's generic approach implements several basic principles.

HARP's security embedded into any application to be instantiated over the web-based environment outlined above follows object oriented programming principles. It is based on Internet technology and protocols solely. The trustworthiness needed has been provided by

applying only certified components which are tailored according to the principal's role. In fine-grained steps, it establishes its complete environment required avoiding any external services possibly compromised. After strong mutual authentication based on smartcards and TTP services, the security infrastructure components are downloaded and installed to be used for implementing the components needed to run the application as well as to transfer data input and output. The SSL protocol deployed to initiate secure sessions is provided by the Java Secure Socket Extension API. The applets and servlets for establishing the local client and the open remote database access facilities communicate using the XML (Extended Markup Language) standard set including XML Digital Signature. Because messages and not single items are signed, the messages are archived separately for accountability reasons meeting the legislation and regulations for health.

Policies are dynamically interpreted and adhered to the components. All components applied at both server and client site are checked twice against the user's role and the appropriate policy: first in context of their selection and provision and second in context of their use and functionality.

Applet security from the execution point of view is provided through the secure downloading of policy files, which determine all access rights in the client terminal. This has to be seen on top of the very desirable feature that the local, powerful, and versatile code is strictly transient and subject to predefined and securely controlled download procedures. All rights corresponding to predefined roles are subject to personal card identification with remote mapping of identity to roles and thereby to corresponding security policies with specific access rights.

For realising the services and procedures described, an applet consists of the sub-components GUI and interface controller, smartcard controller, XML signing and XML processing components, communication component applying the Java SSL (Secure Socket Layer) extension, and last but not least the data processing and activity controller. Beside equivalent sub-components and an attribute certificate repository at the server side, policy repository, policy solver and authorisation manager have been specified and implemented as a "light weight Resource Access Decision Service (RAD)" which has been explained in Chapter 8.3.2.

After exchanging certificates and establishing the authenticated secure session, servlet security is provided from the execution point of view through listing, selecting and finally executing the components to serve the user properly. By establishing an authenticated session that persists for all service selections, a single-sign-on approach can be realised.

In the server-centric approach, a web-accessible middleware has been chosen based on its support of basic security functionality, e.g., MICO/SSL., Apache Web server with mod_ssl, Apache JServ, and Apache Jakata Tomcat.

The basic components of the HARP Cross-Security Platform (Figure 12.16) have been introduced already in Chapter 12.5. This architecture enables the following generic scenario:

- The user connects to a dedicated web server via his browser and uses of course a secure protocol such as HTTPS. (step 1)
- The private key of the user is stored on a smartcard or in a software PSE (prerequisite for the mutual authentication in a SSL/TLS connection). (step 1)
- The web server may accept or deny a connection request based on its policy and the user public-key certificate presented. User and server authenticate each other with the mutual authentication scheme of the SSL/TLS protocol. The SSL/TLS protocol does not prescribe client authentication in order to establish a secure connection, but the policy defines this (i.e., the Web server is configured to request a client certificate). (step 1)

- The web site provides a Java applet execution policy that the user should install on his computer in order to allow the HARP applets to function without problems. This is again up to the site's policy to decide. Finally the applet is automatically downloaded. (step 2)
- The application applet is downloaded to the user's site and further tasks are initialised. The applet initiates a secure connection to the Web server in order to take advantage of the available services running within the server in form of servlets. (step 2)
- The identity (ascertained by the public-key certificate) and policy (for accessing data) retrieved from the policy server are used to identify the roles the user is able to take up. This is done via the Authorisation Manager (AM) and depends on the attribute certificates issued and made available by the Attribute Authority. (step 3, 4)
- Access to the database server is controlled by the role of the user, e.g. documentation instance, proof instance, student. The database is a relational one. (step 5)
- Correspondingly, on the client side, the presentation view of the application to the user is again controlled by his role; thus presented forms have shaded fields, i.e. fields the user is not allowed to change or see (due to policy) and a set of fields for input/output.
- The specific assignment of users to roles mentioned in the previous step uses attribute certificates which reside in an Attribute Authority. This is the appropriate approach to have the substantiation of roles well demarcated. As a consequence the effect of roles can be clearly separated from the development of the underlying application.

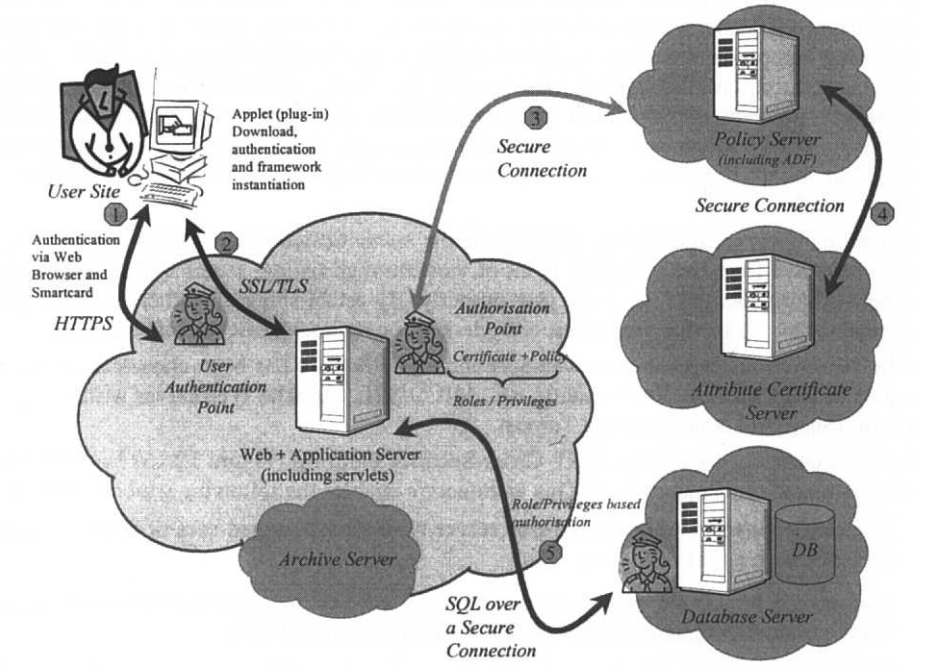


Figure 12.16:The HARP Cross Security Platform Architecture [HARP_WWW]

The sequence diagram for user authentication is presented in Figure 12.17.

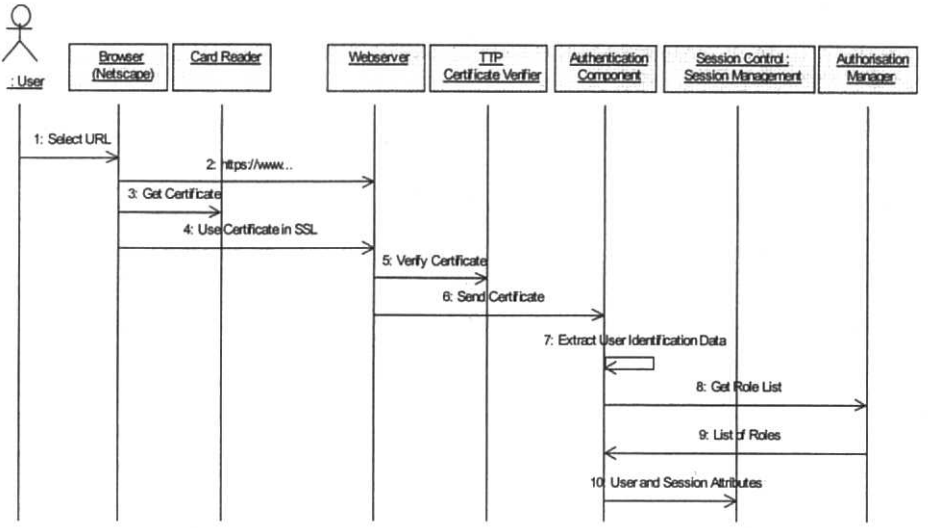


Figure 12.17: Authentication Sequence Diagram

The authentication sequence describes the authentication procedure.

1. *Select URL:* The User selects the URL of the target system (hospital, portal, ...) in the browser (Netscape Browser)
2. *https://www....:* The browser connects to the Web server. The Web server is configured such as to request a client certificate.
3. *Get Certificate:* The browser accesses the smartcard of the user to read the user public-key certificate. The user browser/system has to be configured for smartcard access, i.e. PKCS#11, OCF and dll-files have to be installed as required before the system is used (dynamic installations are a future enhancement if required).
4. *Use Certificate in SSL:* The browser SSL component transmits the user certificate to the server within the establishment phase of an SSL connection.
5. *Verify Certificate:* The certificate is verified within the SSL component of the Web server. This might be a local procedure, if all relevant verification information such as the CA certificates and CRL are already available in the Web server or this might be an online verification procedure with e.g. OCSP to a TTP.
6. *Send Certificate:* The X.509 certificate is extracted from the SSL component and handed over to the Authentication Component.
7. *Extract User Identification Data:* The unique user identification information is extracted from the certificate. This depends on the authentication policy and can e.g. be the Distinguished Name (DN) of the user contained in the certificate or the sequential number of this certificate in combination with the certificate issuer information.
8. *Get Role List:* The possible roles of the identified user are requested from the Authorisation Manager.
9. *List of Roles:* The list of roles is returned to the Authentication Component.

10. *User and Session Attributes:* Relevant user attributes and session data has to be kept and managed by the Session Control component. Based on these attributes the list of services a user is allowed to access and use may be requested.

Based on the attributes/privileges of the user a certain set of services is available. The components participating in the service selection use case are presented in the following sequence diagram.

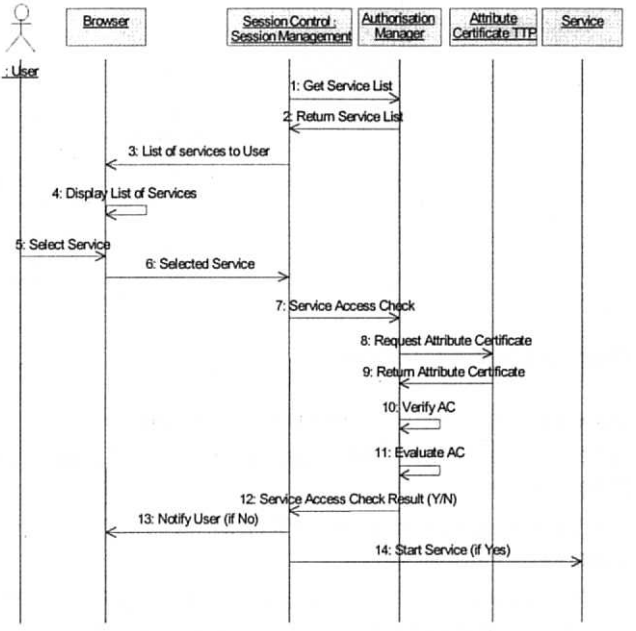


Figure 12.18: Service Selection Sequence Diagram

The service selection sequence describes the selection of a service.

1. *Get Service List:* The list of services accessible by the user is requested.
2. *Return Service List:* The list of services is returned.
3. *List of Services to User:* The list of services is returned to the browser (optional—due to the fact that within a dedicated trial environment only one service is available; an explicit selection by the user is not needed then).
4. *Display List of Services:* The browser displays the list of services (optional, see #3).
5. *Select Service:* The user selects a service (optional, see #3).
6. *Selected Service:* The service selection choice is transmitted to the Session Control component (optional, see #3).
7. *Service Access Check:* The access to the selected service has to be checked: "Will user U in circumstances X get access to service S_1?" (optional: based on the user identity and role only services are presented to the user, that are allowed to be executed. Based on certain policies, this service usage might depend on additional attributes/circumstances such as e.g. the time of day, the terminal equipment used etc. For simplicity the HARP demonstrator does not take into account these additional attributes).

8. *Request Attribute Certificate:* A request for available attribute certificate(s) is sent to the Attribute Certificate TTP (optional, see #7).
9. *Return Attribute Certificate:* The attribute certificate(s) is returned (optional, see #7).
10. *Verify AC:* Possibly a verification of the attribute certificate has to be performed, if not done by the Attribute Certificate TTP already (optional, see #7).
11. *Evaluate AC:* The attribute certificate is evaluated by the Authorisation Manager (optional, see #7).
12. *Service Access Check Result (Y/N):* The result of this evaluation (Yes: access allowed or No: access not allowed) is returned to Session Control component (optional, see #7).
13. *Notify User (if No):* If access is not allowed, the user has to be informed (optional, see #7).
14. *Start Service (if Yes):* If access is allowed, the selected service is started for the user (if the optional sequences are not executed, start service is always initiated if only one service is available).

If EHR systems or clinical practice guidelines have to meet the challenge of controlling processes and workflows and influencing them, systems have to react on specific events in an appropriate way. This reaction might happen by invoking specific specifications or proper services eventually programmed in special languages such as Java. Markup languages as expression tools for constraint models at meta-model level like XML provide first solutions for event control to complete their functionality in system design and implementation. One way to meet this challenge is the specification of events and their consequences using XML DOM Level 2 specifications. Therefore, the event control is related to a document tree defined in the DOM context. If a defined event occurs, it will be propagated down this tree, starting with a root element via nodes and elements and looking for a defined target element. The target element, also called listener, starts a specified action. On "its way", observer elements can mention this event and react with an own event triggered by a handler element which is connected to the observer. Such handler event could, e.g., prepare the target event's procedure. The capturing phase from root down can be inverted up to the root again by a bubbling phase, a corresponding specification of the target object assumed. If the event concerns the prescription of morphins for a patient, only authorised doctors should be able to perform this procedure. In this scenario, the observer element could check the authorisation of the user by requesting a password. An XML event module contains elements and attributes to be used for event control. Elements and attributes are constraint using a DTD specification as shown in Figure 12.19.

```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT  EventHandling (#PCDATA)>
<!ATTLIST  EventHandling pfx:Event          NMTOKEN          #REQUIRED
                         pfx:Observer       IDREF            #IMPLIED
                         pfx:Target         IDREF            #IMPLIED
                         pfx:Handler        %URI             #IMPLIED
                         pfx:Phase          (capture|default) #IMPLIED
                         pfx:Propagation    (stop|continue)  #IMPLIED
                         pfx:DefaultAction  (cancel|perform) #IMPLIED>
```

Figure 12.19:Part of an XML Event DTD

The event attribute valued by a XML nameToken specifies the event type. All the attributes must be uniquely idientified. The handler's reaction at the observer element level is specified at the given URI. The phase attribute defines, wether the phase path should be followed

down or also upwards. The propagate attribute defines the stop or the continuation of the path after the event was happening, if the defaultAction attribute specifies the reaction on the event at the target element. The observer element's reaction of requestinf the user's password might be expressed as follows (Figure 12.20).

```
<secret ref="/login/password">
        <caption>Please enter your password for authentication</caption>
        <info ev:Event="Attention">
                A password is required for authentication
        </info>
        <info ev:Event="Hint">
                Please select the drug for prescription
        </info>
        <info ev:Event="Cancellation">
                You are not authorised to prescribe morphins
        </info>
</secret>
```

Figure 12.20:Observer Object with Connected Event Handlers

The target event manged by the listener might be attributed as presented in Figure 12.21, reflecting the observer reaction.

```
<Listener Event="activate" Observer="button2" Target="link1" Handler="#info">
```

Figure 12.21:Script Snipet of the Listener Specification

The mentioned target object's declaration is finally given in .

```
<a ID="link1" href="sample.html">Select the drug needed</a>
```

Figure 12.22:Target Object Declaration

12.8 Decision Support Systems

In health, the need for offering guidance to the Health Professional is obvious. This concerns state of the art knowledge and procedures which means diagnosis and therapy. With the promotion of evidence-based medicine, the importance of guidelines is growing. The guidance is increasingly established in clinical practice guidelines (CPGs) helping to reduce inappropriate variations in diagnosis and treatment. It that context, the supportive character of CPGs must be emphasised, not deliberating Health Professionals from their responsibility for the individial patient with his individual case. Clinical guidelines' benefits, their limitations or even harm have often been discussed (see e.g. [Woolf et al., 1999]. For improving guideline compliance, computerised guidelines as well as interactive guideline systems adapting guidelines to the patient's requirements and conditions became the way of choice. So, the reuse of guidelines, dissemination and updates are facilitated. Patient-specific guideline knowledge is provided to the specific point and time of care. Interactive presentation of CPGs could also occur as recommendation by a critiquing or monitoring engine, as alerts or as reminders [Elkin et al., 2000]. For more references, see, e.g., [Ohno-Machado et al., 1998; Elkin et al., 2000].

12.8.1 Electronic Guideline Representation

In the literature, at least five different basic approaches for electronic representation of clinical guidelines can be found: Rule-based specifications, decision analysis representation of guidelines, state-transition networks and knowledge bases for establishing guidelines, guideline mark-up methodologies, multi-step guidelines. If the first three offer algorithms for guideline-controlled procedures for active decision support, the latter ones are based on documents offering document-related facilities to be used by the Health Professional for

making the right decision. For rule-based specification, an appropriate language for encoding the rules has to be introduced. Examples of such languages are the Arden Syntax for Medical Logic Modules (MLMs) [Hripcsak et al., 1994] following the HELP system's paradigm [Kuperman et al., 1991], or the G-CARE language [Overhage et al., 1995]. The decision analysis representation of guidelines is based on logical models and tools for knowledge representation and decision making as shown, e.g., in [Sanders et al., 2000]. The best-known example for guidelines established by markup methods is the Guideline Element Model (GEM) [Schiffman and Nath, 2000]. Closer adapting the Health Professionals' traditional thinking, the large group of multi-step guidelines model knowledge in an object-oriented way by complex combination of steps as a hierarchical set of nested guideline tasks. Important projects belonging to that group of guideline representation are the European Prestige project [Gordon and Veloso, 1999] or the UK projects Prodigy [Sugden et al. 1999] and PRO*forma* [Fox and Rahmanzadeh, 1998], but also the XML-based Clinical Practice Guidelines (xCPGs) [Hoelzer et al., 2001] authorised by the Giessen University. At the Stanford University, the intention-based language for CPG representation ASBRU has been created, which allows to explicitely express guideline intentions, patien tstatus, and prescribed actions dynamically [Shahar et al., 1998]. Originated by the same site, the component-based EON system for knowledge representation including domain ontology, eligibility criteria, abstraction definitions, guideline algorithms, revision rules, and a temporal query language has been developed [Musen et al., 1996].

For representing CPGs in maschine-readable format, the GuideLine Interchange Format (GLIF) has been instroduced by a group of acknowledged researchers of several US universities. Meanwhile, several authoring tools such as PROTÉGÉ [Grosso et al., 1999] and GEODE [Greenes et al., 1999] have been developed. Arden Syntax and GEM, but also PRO*forma* or the related knowledge representation frameworks of G-CARE, ASBRU, GLIF, etc., specify a formal structure for this knowledge representation, which is similar to the GEHR and openEHR approach focussing on data-driven architectural aspects. Progidy and xCPGs are more flexible allowing also narrative text as it is most provided by medical experts. Offering retrival criteria including the proper structuring of guideline output (e.g., presentation most important issues on top) will enable much higher quality and therefore acceptance in using clinical guidelines according to the document paradigm. Deploying XML methods for structuring information, establishing associations, and referencing recources, also the second group moves in the direction of active decision support as visible in recent papers about the exploitation of XML Topic Maps. For further references see, e.g. [Schweiger, 2002].

Within this chapter's approach offered for future proof HIS, active guidelines consist of constraint models describing data and operations, conditions, etc. required, recommended, or to be avoided, including certainty factors and other qualifiers. In that context, HARP-based systems are able

- to adapt their specificity to the data present, by that way dealing properly with incomplete or missing information,
- to react on user needs, requirements, external conditions by interactive combination of appropriate components,
- to follow predefined workflows (e.g. batch processing, rule-based processing) by concept-based predefined aggregation of components,
- to offer multiple and selective views on data according to domain-specific concepts and constraints,
- to provide flexible presentation of information,
- to allow intelligent mapping of external data and internally generated information.

For the simple example for an active and interactive diabetes-hyperlipidemia guideline including the basic packages only (Chapter 12.2), an XML document set can be specified. Independently defined by a group from Massachusetts General Hospital [Dubey and Chuch, 2000], this approach for guideline interchange and execution confirms the generic character of the HARP approach. The proposed system consists of an Environment XML document, a Data_Interface XML document, a Logic_Specification XML document, a pre-processor processing them to a mega XML tree, which is processed by the guideline engine using the mega XML tree to transform it to an adapted mega XML tree reflecting environmental data, patient-specific information as well as interactive entered data. The next figures present the sample documents for specifying and afterwards processing clinical guidelines in the way described before.

```
<ENVIRONMENT>
    <DATA>
        <DATUM ID="latest_Id" REQUESTED="yes|no" DATE="" SESSION_ID="">
            data (as XML)
        </DATUM>
    </DATA>
    <OUTPUT>
        <ASSERTION>
            <DATE>3/1/00</DATE>
            <SOURCE_INFO>
                <GUIDELINE_NAME></GUIDELINE_NAME>
                <SESSION_ID>11215 </SESSION_ID>
                <STEP_NAME>check_Id<STEP_NAME>
            </SOURCE_INFO>
            <LABELS>
            <CONTENT> content (as XML)</CONTENT>
        </ASSERTION>
    </OUTPUT>
    <SYSTEM_STATE>
        <STATUS>continue/stop/finished</STATUS>
        <NEXT_STEP></NEXT_STEP>
        <TIME_OF_LAST_STEP></TIME_OF_LAST_STEP>
        <TIME_OF_NEXT_STEP></TIME_OF_NEXT_STEP>
    </SYSTEM_STATE>
</ENVIRONMENT>
```

Figure 12.23:Sample ENVIRONMENT XML Document (after [Dubey and Chuch, 2000])

```
<DATA_INTERFACE>
    <DATUM ID="latest_Id">
        <DATA_TYPE>int|float|string|boolean</DATA_TYPE>
        <XSL_QUERY></XSL_QUERY>
        <CONSTRAINTS>
            <CONSTRAINT>age<50</CONSTRAINT>
        </CONSTRAINTS>
        <FORM_DATA></FORM_DATA>
        <EXTERNAL_SOURCE>
            <PARAMETERS>
                <PARAMETER_ID0"unitnumber"/>
            </PARAMETERS>
            <CODE>
                //Javascript code here
            </CODE>
        </EXTERNAL_SOURCE>
    </DATUM>
</DATA_INTERFACE>
```

Figure 12.24:Sample DATA_INTERFACE XML Document (after [Dubey and Chuch, 2000])

```
<LOGIC_SPECIFICATION>
    <TYPE>logic-specification</TYPE>
    <LABELS>
        <LABEL NAME="GUIDELINE NAME">hyperlipidemia</LABEL>
    </LABELS>
    <FIRST_STEP NAME=""/>
    <STEPS>
        <STEP NAME="">
            <BACKING>
                <REFERENCE></REFERENCE>
            </BACKING>
            <DATA><DATUM ID="ldl"7></DATUM>
            <CONDITIONAL>
                <PROSE></PROSE>
                <LOGICAL>((ldl<150)&&(trig<400))</LOGICAL>
            </CONDITIONAL>
            <TRUE>
                <OUTPUT>
                    <URL></URL>
                    <XSL_QUERY></XSL_QUERY>
                </OUTPUT>
                <NEXT_STEP NAME=""/>
            </TRUE>
            <FALSE>same child nodes as TRUE</FALSE>
            <UNKNOWN>same child nodes as TRUE</UNKNOWN>
        </STEP>
    </STEPS>
</LOGIC_SPECIFICATION>
```

Figure 12.25:Sample LOGIC_SPECIFICATION XML Document (after [Dubey and Chuch, 2000])

The different nests of the documents shown in the figures represent specific concepts and constraints, which rules the use of the clinical guidelines presented as examples.

12.8.2 Security Services for Clinical Guidelines

Clinical practice guidelines establish the specific set of security and safety requirements valid for any form, recommendation, instruction, order, etc. The content must be kept integer, the origin of the document has to be verifyable. On the other hand, the accessability and usability of the document is essential. Normally, there is no need for excluding specific domain-related users from access to the information. Excluding external users may be reasonable.

Regarding the security dimensions introduced in Chapter 6, requirements for availability, integrity, and authenticity are high, requierements for confidentiality are low. The emerging revision of CEN ENV 12924 "Medical Informatics – Security Categorisation and Protection for Healthcare Information Systems" meets these specific requirements for templates, guidelines, and similar "publicly" available as well as binding information [CEN ENV 12924].

12.8.3 Further XML-Related Security Specifications

Acknowledging the growing importance of platform-independent and programming-language-independent specification like Web services, thrustworthiness is a crucial requirement of the market. This trustworthiness comprises all communication security and application security services introduced. In that context, the communication security services authentication, integrity, confidentiality, and availability must be especially considered. Following, some basic services meeting these security requirements for Web services using the XML standard set will be discussed shortly.

XAML (Transaction Authority Markup Language) has been specified to define XML message formats and interaction models. By that way, the transaction operations *commit*, *cancel*, *retry*, *undo*, and *reverse* can be realised, which are needed to assure transactionality of Web services by TP (transaction processing) monitors. Transactionality assurance according to the XAML specification can be realised in three phases: request of a service, sending

a session ID for the available services to the requestor, realisation of transaction operations. Regarding application security services, SAML (Security Assertion Markup Laguage) has been specified partially realising the main access control concepts Access Matrix Model and Role-Based Access Control. SAML provides authentification, authorisation of access to Web services, and security of communicated data using defined XML data sets. If a request has been accepted as secure, it can be delegated according to the simple delegation model introduced, e.g., in the CORBA Security Service Specification. For securing SOAP messages, the security services S2ML and AuthXML have been included into the SAML specification. Confidentiality of XML messages can be provided in accordance with the XML Encryption Specification. Authentication service might be provided on the basis of PKI, which is managed following the XML Key Management Specification (XKMS). XKMS consists of two parts: the XML Key Information Service Specification X-KISS and the XML Key Registration Service Specification (X-KRSS). Both the X-KISS and the X-KRSS protocol do not X509 certificates. Because the specifications are still unstable, the valid specification should be checked at the W3C Web site [W3C_WWW]. As introduced in Chapter 6.13 and according the security policy agreed, appropriate security mechanisms, protocols and data (e.g. X509v3) should be introduced in the sensitive environment of health, complying with standards like ISO 17090 [ISO 17090].

12.9 Summary and Conclusions

Any EHR approach performed by the experienced players around the globe moves or will soon move towards a multi-model approach exercised by the HARP Consortium. At the moment, some of the specification teams have restricted views on specification issues only. However, there is an urgent need not only to describe the future but also to realise advanced products and tools for creating and managing them.

As mentioned in Chaper 5 already, the reference system is characterised by a reference model, several constraint models reflecting the different domains' knowledge, and the RM-ODP views for all those models.

The approach may also be used for dealing with Clinical Practice Guidelines and decision support systems. To promote interactive EHR systems as well as workflow management and controlling, XML-based event handling has been discussed and demonstrated.

Currently, the improvement of HARP's tools for modelling EHR systems, mapping the meta-models by XMI, defining different XML schemata using XSLT, and implementing as well as maintaining them with deployment of enhanced HARP tools presented is the main objective of the Magdeburg Medical Informatics Department and its international partners at UCL, GEHR and CORBA community, etc.

The generic, component-based, multi-model approach offered also enables specification, implementation and maintenance of decision support functionality.

The propsed approach for future-proof EHR systems has been practically demonstrated for clinical studies.

13 European Projects Contributing to the Paper

13.1 Introduction

The results presented in the paper have been elaborated amongst others within several projects funded by the European Commission. The projects have been established within two programmes of the European 4th Framework for Research and Development, the Telematics Applications Programme (TAP) and the Information Society Initiative for Standards (ISIS).

13.2 The DIABCARD Project

The DIABCARD-3 project[44] [DIABCARD_WWW] is a project of the European Commission Health Telematics Applications Programme. A specific requirement of the DIABCARD-3 project was the interoperability between all test sites, but also co-operation with the DIABCARE, NETLINK, and CARDLINK initiatives. The project is based on the foregoing work of the DIABCARD 1 and the DIABCARD-2 projects. As its predecessor projects DIABCARD-1 and DIABCARD-2, it aims at improved Diabetes Care using smartcard based information systems and technology. A chip card based medical information system (CCMIS), the patient data card (PDC), makes the up-to-date patient's record available whenever it is needed and thus replaces paper records. At the same time it offers security, data integrity and confidentiality. The use of the diabetes-specific PDC, the DIABCARD, improves the communication in routine diabetes care and thus DIABCARD becomes an important tool for quality assurance in chronic healthcare and in emergency care through read-access of the Cardlink card.

The objective of the DIABCARD-3 project was the specification of the information content and the methodology to support the care of diabetes patients by documentation and communication of essential data between the care providers involved. A Basic Data Set as well as the DIABCARD architecture have been specified and implemented in test sites of six European countries. An open DIABCARD workstation architecture [Böhm et al., 1997] was developed including the DIABCARD infrastructure needed. The concept of a portable patient record is realised by using the DIABCARD patient data cards (DIAB.PDC).

13.3 The HANSA Project

The HANSA[45] project was a project of the European Commission's 4th Framework "Telematics Applications Programme". It was dealing with DHE (Distributed Healthcare Environment) as distributed interoperable health information systems based on the European HISA (Healthcare Information Systems Architecture) standard. The approach included any health provider facilitating the *shared care* paradigm. Demonstration sites in several European countries provided pilots applying the DHE to show applicability of that architectural solution. The DHE was established as an integration plattform for newly developed as well as legacy departmental systems. Within the HANSA project, the author's work on the Generic Component Model has been performed.

13.4 The ISHTAR Project

Like DIABCARD, ISHTAR is a project of the European Commission's 4th Framework "Telematics Applications Programme". Like its predecessor project SEISMED, It deals

[44] A Chip Card Based Medical Information System for Diabetes Patients
[45] Healthcare Advanced Networking Architecture

with high level policies, threat and risk analysis as well as security requirements and solutions for health information systems including education and training [SEISMED, 1996]. However, it doesn't concern real implementations [ISHTAR_WWW; ISHTAR, 2001]. The security models used in this book are ISHTAR project results.

13.5 The TrustHealth Project

Like the projects mentioned before, the TrustHealth project [TRUSTHEALTH_WWW] is a project within the Health Telematics sector of the European Commission 4th framework programme Telematics Applications. The project aims to facilitate the establishment of trustworthy information systems in healthcare by providing a set of specifications for security services and interfaces, and a Trusted Third Party service infrastructure (TrustHealth-1). The successor project TrustHealth-2 aims at operational systems in some countries and publicly available specifications to demonstrate the feasibility and the cross-border interoperability of the solutions. A TrustHealth fundamental is the usage of a smartcard called Health Professional Card (HPC) or TrustHealth Health Professional Card (TH.HPC) as well as the implementation of Trusted Third Party (TTP) services. Originally, the DIABCARD-TrustHealth Extension was related to the TrustHealth-1 project results. Because it wouldn't be reasonable to introduce now solutions from 1997 which are not delivered and supported any more, the TrustHealth-2 technology is used to implement enhanced DIABCARD security solutions.

13.6 The EUROMED-ETS Project

The EUROMED-ETS[46] project [EUROMED_WWW] was part of European Commission's ISIS programme. This programme's challenge was the establishment of a security infrastructure for an Internet-based Pan-European Health Network. In 1997, between the project partners, the universities of Athens, Calabria and Magdeburg such a TTP needed could be implemented practically [Katsikas et al., 1998]. The secure clients and server specified have been evaluated using a sophisticated scheme. This scheme was also deployed to evaluate the DIABCARD-TrustHealth Extension project solutions.

13.7 The MEDSEC Project

Like the EUROMED-ETS project, the MEDSEC[47] project [MEDSEC_WWW] has been funded by the European Commission within the "Information Society Initiatives for Standards" (ISIS) programme and dealt with the review of existing and emerging standards in the healthcare domain identifying their gaps and assessing their applicability leading to the enhancement of security aspects. The Magdeburg Medical Informatics Department was responsible for analysis, specification and implementation of Electronic Data Interchange (EDI) security which is deployed in several projects such as TrustHealth-2 and RESHEN dealing with implementation and exploitation of the ONCONET [Blobel et al., 1998a,b; Blobel et al, 1999]. Recently, the main MEDSEC project results have been published in the IOS "Series in Health Technology and Informatics" [Allaert et al., 2002].

13.8 The HARP Project

Originally, the HARP project [HARP_WWW] aimed at the specification, implementation, and evaluation of a security platform for Web applications. The resulting HARP Cross Se-

[46] Trusted Third Party Services for Health Care in Europe
[47] Health Care Security and Privacy in the Information Society

curity Platform HCSP provided the basis for enhanced TTP services, especially establishing application security services based on certified components.

The architectural principles introduced could be enhanced too by using the HARP methodology as an environment for specifying, implementing, and maintaining components based EHR architectures based on multi-models and the use of XML as archetype description and exchange format. The HARP results have been demonstrated for a clinical study being a minimised EHR example.

13.9 The RESHEN Project

Running until end of 2002, the RESHEN project aims at the specification, implementation, and assessement of interoperable health applications based on an advanced PKI. In that context, the specification, implementation, operation, and evaluation of TTP services including cross certification services have to be installed at regional, national, and international scale. The project is part of the European Commission's "Best Practice Programme" (part of IST) involving partner from Greece, Finland, and Germany, the latter represented by the Magdeburg University Hospital.

13.10 German Partners

Beside the University of Magdeburg (Medical Informatics Department), the steady German partners involved in the realisation of the several projects' pilots presented are the GMD (now Fraunhofer Gesellschaft) Darmstadt as developer of the security toolkit SECUDE™ and Giesecke & Devrient Munich as provider of smart cards, card readers and card operating systems, the Physician Chamber Lower-Saxony, the Pysician Chamber Saxony-Anhalt, and the University of Goettingen (Medical Informatics Department) in the TrustHealth project as well as several Cancer Centre members as users. Further German main partners are the University of Giessen (Institute of Medical Informatics) in the HANSA project, GSF Munich in the DIABCARD project as well as the Fraunhofer Gesellschaft – FOKUS team within HARP.

14 Conclusions

Dealing with analysis and design of security enhanced distributed health information systems, the comprehensive paper tries to develop a systematic approach to meet this challenge and to provide support to the different user groups involved satisfying their specific need and expectations. Because it is more intelligible to summarise the results for sometimes very specific investigations, each chapter closes with related conclusions. Therefore, in the following only a condensed summary will be given providing a red thread through the different views culminating in a feasible methodology for analysis, design, and implementation of secure *shared care* information systems.

The *shared care* paradigm is the only response to the challenge the developed countries are confronted with. Looking for available solutions for health information systems, the communication and interoperation between applications needed in a distributed environment can be provided by two integration types: interfacing or integration. Only integration provides the interoperability in the sense of added value functionality. With different levels, integration can be provided by different architectural paradigms. Having started designing and implementing the Magdeburg University Hospital HICS as a very early client-server approach back in 1991, the author has introduced the concept of interoperable objects or "atomic components" for really integrated solutions.

To find the appropriate paradigm for analysis, design and implementation of security enhanced, open, and generic shared care information systems architectures, some of the most progressive integration platforms for health information systems available as products already or in the next future have been carefully investigated: CORBA, DHE, and HL7. Characterising the approaches considered, the paradigm of reusable objects could be selected. Likewise the competing ones, this paradigm, however, demonstrates dependencies on the underlying technology. Furthermore, ignoring concepts at other levels of granularity and abstraction, the objects are not really reusable in the context of business concepts and enhanced functionalities. This is especially obvious, if the approach concerns security issues with their social, legal, ethical, organisational, and technical relationships. Therefore, other concepts like component paradigms including but not restricted to objects have been introduced.

Considering the component paradigm which was originally developed for software engineering very carefully, a modelling approach can be defined which enables the different views on security enhanced shared care information systems with their strategic, business, social, organisational, and technological frameworks. Components are independent of the technological fundamentals. Component systems reflect different levels of abstraction for transferring to other underlying paradigms as well as different levels of granularities from a rough structure up to detailed lines of codes.

Regarding the component's state transition using process models, the abstract automaton approach and recursive functions, evidence could been brought in the feasibility of a consistent paradigm within the continuum of abstraction and formalisation to be considered in the context of our challenge. Bewaring the central characteristics of information, the different view and user needs can be separated, granulated and resolved step by step and piece by piece.

Based on these findings and the generic component model defined, the assurance for formalising the problems, requirements, and solutions using such methods originally created for software development as UML, etc., has been presented.

Rationally, the next step is to specify component models which might reflect the needs of, and are understood by, the different user groups involved in the process of design, imple-

mentation, and use of the information systems from lawyers and management staff up to physicians, nurses, members of the maintenance staff, system administrators and software developers as well as implementers. The resulting description and specifications have to be consistent.

Consequently, components models have been developed reflecting different levels of granularity as well as abstraction. The component approach has also been extended to manage security issues. One the one hand, different components at the conceptual level as communication and application security have been defined enabling the separate handling of the related issues as demonstrated in our pilots successfully implemented (e.g., secure EDI communication). On the other hand, different levels of abstraction as concepts, services, mechanisms, algorithms, data, and even protocols as well as products have been specified in a generic layered security model in a consistent way.

Following the approach developed by describing real systems with appropriate granularity and abstraction, Lego®-type basic elements for scenarios, also called use case types, could be defined for both medical and security related processes in the context of distributed health information systems. The investigations have been performed on the basis of an extended analysis of real-life scenario including international aspects. Using these use case types (6 for medical scenarios and 8 for security-related scenarios), any complex system may be characterised by combining these use case types in a proper way. Because the use case types and their instantiations are components, they can be handled separately facilitating design and implementation of complex and costly systems.

Due to the importance of EHR for any medical scenario, EHR systems can be defined as basis application for any health information system architecture. So, EHR architectures have been considered in a special chapter introducing especially the newer concepts and comparing them for deriving a harmonised future solution. The HARP project originally dedicated to security issues has been developed as such future-proof EHR paradigm. It provides all ISO RM-ODP views, by that way enabling specification, implementation, and maintenance of EHR, EHR architecture, and EHR systems. The constraint models are specified using graphical (UML) and textual (XML standard set) means. The future-proof and tool-based architecture mentioned has been partially demonstrated.

Based on the approach presented, existing environments and solutions have been investigated and improved regarding secure system architecture. In that context, CORBA and EDI communication have got special attention.

Nowadays security solutions, which are often based on cryptographic algorithms, require a corresponding security infrastructure dealing with the keys and providing Trusted Third Party services needed. Within the framework of the European projects we have been involved in, the European security infrastructure based on Health Professional Cards and TTPs has been developed and described, especially reflecting the different pilots and trials the Magdeburg Medical Informatics Department is participating. The open challenge of providing appropriate application security services could be overcome by the HARP Croos Security Platform with enhanced TTP services for authorisation, assignment, privilege management, etc. Furthermore, the HARP project has been moved towards a comprehensive, scalable, portable, flexible, interoperable, and secure EHR approach improving novadays EHR initiatives.

Summarising the results it can be stated, that based on generic architecture and security models for health information systems enabling communication and co-operation to meet the *shared care* paradigm, the formalisation, analysis, design, specification, implementation, use, and maintenance of security enhanced information systems in the healthcare domain can be supported efficiently. Because the component-based approach separates the different levels of granularity and abstraction facilitating the different user group views

within a homogeneous framework, the methodology promotes open solutions independent of the underlying technology. However it must be clearly mentioned, that any information system solution depends on the political, social and behavioural environment and especially on the acceptance by the users. Therefore, interaction with as well as inclusion of all user groups, education, training and – regarding security – the improvement of awareness are basic conditions to be established for a successful work on our field.

15 Definition and Interpretation of Basic Terms Used

The following list gives an overview about security-related services, mechanisms, algorithms, and data used in the monograph. As far as possible, standardised terms are used. Otherwise, definitions agreed about in European projects concerning security are employed.

Aborted connection
Documented agreements containing technical specifications or other precise criteria to be used consistently as rules, guidelines, or definitions of characteristics to ensure that materials, products, processes, and services are fit for their purposes
[ISO/IEC 2382-8]

Abstract security mechanisms
Security mechanism described in a generalised fashion, without specific choices made for algorithms.
[CEN ENV 13608]

Access control
Means of ensuring that the resources of a data processing system can be accessed only by authorised entities in authorized ways.
[ISO/IEC 2382-8]

Access control list
A list of entities, together with their access rights, which are authorised to have access to a resource
[ISO/IEC 2382-8]

Access level
The level of authority required from an entity to access a protected resource
[ISO/IEC 2382-8]

Access period
A period of time during which specified access rights prevail
[ISO/IEC 2382-8]

Access permission
All of a subject's access rights with respect to some object
[ISO/IEC 2382-8]

Access right
Permission for a subject to access a particular object for a specific type of operation
[ISO/IEC 2382-8]

Access type
A type of operation specified by an access right
[ISO/IEC 2382-8]

Accountability
Ensures that the actions of an entity may be traced uniquely to the entity.
[ISO 7498-2]

Analytical attack
An attempt to break a code or to find a key using analytical methods (e.g. statistical analysis of patterns, discovering flaws in an encryption algorithm)
[ISO/IEC 2328-8]

Assurance
Confidence that an entitiy meets its security objectives
[ISO/IEC CD 15408-1]

Asymmetric authentication method
A method of authentication, in which not all authentication information is shared by both entities
[ISO/IEC 10181-2]

Asymmetric cryptographic algorithm
Asymmetric encryption algorithm
Algorithm for performing encipherment or the corresponding decipherment in which the keys used for encipherment and decipherment differ.
[ISO/IEC 9798-1]

Asymmetric cryptographic technique
A cryptographic technique that uses two related transformations, a public transformation (defined by the public key) and a private transformation (defined by the private key). The two transformations have the property that, given the public transformation, it is computational infeasible to derive the private transformation
[ISO/IEC 11770-1]

Asymmetric signature system
A system based on asymmetric techniques whose private transformation is used for signing and whose public transformation is used for verification
[ISO/IEC 9798-1]

Attributability
The property that ensures that events that occur in the system, and are traced to their authors by the system, can be successfully attributed to the corresponding security subjects.
[CEN ENV 13608]

Audit
An (external) investigation to determine compliance to specifications, standards and pre-determined agreements
[ISO/IEC 2328-8]

Auditability
The property that ensures that events that occur in the system and are traced by the system can be reliably attributed to the corresponding security subjects, and authors of these events.
[CEN prENV 13608]

Authentication
The provision of assurance of the claimed of an entity
[ISO/IEC 10181-2]
Process of reliably identifying security subjects by securely associating an identifier and its authenticator
[ISO 7498-2]
NOTE See also data origin authentication and peer entity authentication.

Authenticator
Means used to confirm the identity or to verify the eligibility of a station, originator, or individual
[NCSC TG-004]

Authorisation
Authorisation is the process of managing access policies and privileges to resources by authenticated principals.
The granting of rights, which includes the granting of access based on access rights
[ISO/IEC 2328-8]

Availability
Property of being accessible and useable upon demand by an authorised entity.
[ISO 7498-2]

Biometrics
Measurable, unique physical characteristic or personal trait used to recognise the identity, or verify the claimed identity, of an enrollee

	[ANSI/SIA 3]
	Pertaining to the use of specific attributes that reflect unique personal characteristics, such as a fingerprint, an eye blood-vessel print, an iris print, or a voice print, to validate the identity of entities.
	[ISO/IEC 2328-8]
Brute-force attack	A trial-and-error attempt to violate computer security by trying possible values of passwords or keys (contrast with analytical attack)
Card accepting device	Device used to interface with the ICC during a session
CAD	[ISO 10202]
Cardholder	The person to whom the card has been issued
Certificate	An entity's data rendered unforgeable with the private or secret key of a certification authority [ISO/IEC 13888-1]
Certificate distribution	Act of publishing certificates and transferring certificates to security subjects. [TH]
Certificate generation	Act of creating certificates. [TH]
Certificate management	Procedures relating to certificates: certificate generation, certificate distribution, certificate archiving.
Certificate serial number	An integer value, unique within the issuing CA, which is unambiguously associated with a certificate issued by threat CA
Certificate user	An entity that needs to know, with certainty, the public key of another entity. [ISO 9594-8]
Certificate verification	Verifying that a certificate is authentic. [TH]
Certification authority	Authority trusted by one or more users to create and assign certificates. Optionally the certification authority may create the users' keys. [ISO 9594-8]
Ciphersuite	An encoding for the set of bulk data cipher, message digest function, digital signature algorithm and key exchange algorithm used within the negotiation phase of TLS.
Ciphertext	Data produced through the use of encipherment. The semantic content of the resulting data is not available. [ISO 7498-2]
Communication	Exchange of information between principals.
Communication security	Concept for security; providing policy and services for secure communication.
Confidentiality	Protects against information being is not disclosed or revealed to unauthorised principals. [ISO 7498-2]
Consent	Voluntary agreement with what is being done or proposed (epressed or implied)

	[CIHI]
Countermeasure	An action, device, procedure, technique, or other measure that is designed to minimise vulnerability [ISO/IEC 2328-8]
Cryptographic algorithm	Algorithm which transforms data in order to hide or reveal its information content and which uses at least one secret parameter. (In the case of an asymmetric algorithm the data is hidden using a public parameter and revealed using a secret parameter). [ISO/IEC 9979: 1991]
Cryptographic Check Value	Information which is derived by performing a cryptographic transformation on data.
Cryptography	The discipline which embodies principles, means, and methods for the transformation of data in order to hide its information content, prevent its undetected modification and/or prevent its unauthorised use. [ISO 7498-2]
Cryptosystem	A collection of transformations from plain text into ciphertext and vice versa, the particular transformation(s) to be used being selected by keys. The transformation are normally defined by a mathematical algorithm [ISO/IEC 9594-8] The documents, devices, equipment, and associated techniques that are used together to provide a means of encryption or decryption [ISO/IEC 2328-8]
Data integrity	The property that data has not been altered or destroyed in an unauthorised manner. [ISO 7498-2]
Data Origin Authentication	A principal claiming to be the originator of some data includes its identity along with that data glued together using the integrity service. The corroboration that the source of data received is as claimed. [ISO 7498-2]
Decryption	The reversal of a corresponding reversible encipherment. See decipherment. [ISO 7498-2]
Digital signature	Data appended to, or a cryptographic transformation (see cryptography) of a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery e.g. by the recipient. [ISO 7498-2]
Encipherment	The cryptographic transformation of data (see cryptography) to produce ciphertext. [ISO 7498-2]
Encryption	The cryptographic transformation of data (see cryptography) to produce ciphertext.

	See encipherment. [ISO 7498-2]
End-user's security needs	Security requirements from the end user's domain specific viewpoint. [CEN ENV 13608]
Forward secrecy	Technique of ensuring that the communicated data is only decipherable for a limited time span by the communicating parties. After that time the communicating parties typically achieve forward secrecy by destroying cryptographic keys. This prevents an attacker from coercing the communicating parties into decrypting old ciphertext. [CEN ENV 13608]
Generic security functionalities	Set of semi-formal security functionalities.
Human-intrinsic threats	Security threats arising from human involvement in the system.
Health professional card **HPC**	A card issued to a person working professionally in the provision of health services which is used as a security device to provide secure user authentication and possibly other security services
Hashing algorithm	An algorithm used to perform a (mathematical) function that maps values from a (possibly very) large set of values into a smaller range of values [ISO 10181-1]
Identification	Unique name of a principal. Identification - Process that enables recognition of an entity by an IT product. [FC v.1]
Identifier	Piece of information used to distinguish an object including a computer systems user from other objects of the same class.
Integrated circuit card **IC Card** **ICC**	ID-1 card type (as specified in ISO 7810, ISO 7811 parts 1 to 5, ISO 7812, and ISO 7813) into which has been inserted one or more integrated circuits (ICs). [ISO 7816-1]
Integrity	Ensuring consistency of data detecting unauthorised creation, alteration, or destruction of data. [ITSEC]
Key	Sequence of symbols that controls the operations of encipherment and decipherment. [ISO 7498-2]
Key certification	Digitally signing a cryptographic key to indicate to third parties the identity or other attribute of the key owner.
Key distribution	Process of publishing or transferring to other security subjects a cryptographic key. [TH]
Key exchange algorithm	An algorithm used to derive a shared secret over an open communications channel. [TH]

Key generation	Process of creating a cryptographic key. [TH]
Key management	The generation, storage, distribution, deletion, archiving and application of keys in accordance with a security policy. [ISO 7498-2]
Masquerade	A principal pretends to be a different one.
Message authentication	Ensuring typically with a message authentication code, that message received matches the message sent [ISO/IEC 2328-8]
Message Authentication Code (MAC)	Data derived from a message using symmetric cryptography techniques and a secret key to provide authenticity of integrity and origin.
Message digest	See one-way hash function.
Message recovery	Process of a third party decrypting an encrypted message.
Non-Repudiation of Origin	This service is intended to protect against the originator's false denial of having created the content of a message and of having sent a message [ISO/IEC13888-1]
Non-Repudiation of Receipt	This service is intended to protect against a recipient's false denial of having received a message [ISO/IEC 13888-1]
Notarisation	The registration of data with a trusted third party that allows the later assurance of the accuracy of its characteristics such as content, origin, time, and delivery [ISO/IEC 7498-2]
One-way function	A (mathematical) function that is easy to compute but, when knowing a result, it is computationally infeasible to find any of the values that may have been supplied to obtain it. [ISO 10181-1]
One-way hash function	A (mathematical) function that is both a one-way function and a hash function. [ISO 10181-1]
Peer entity authentication	The corroboration that a peer entity in an association is the one claimed. [ISO 7498-2]
Personal identification number **PIN**	The 4 to 12 character alphanumeric code or password possessed by the system user for verification of identity. [ISO 9564-1]
Plaintext	Intelligible data, the semantic content of which is available. [TH]
Policy	A set of rules that specifies the procedures and mechanisms required to maintain the security of a system, and the security objects and security subjects under the purview of the policy. [ECMA]
Principal	Generally, the party involved in communications and co-operations like user, application, system, etc. In the present scope: system or application.

Privacy	Freedom from intrusion into the private life or affairs of an individual when that intrusion results from undue or illegal gathering und use of data about that individual [ISO/IEC 2328-8]
Private key	Key that is used with an asymmetric cryptographic algorithm and whose possession is restricted (usually to only one entity). [ISO 10181-1]
Provability	The property that ensures that events that occur in the system, and are traced, audited and attributed to their authors by the system, can be legally proved as authentic. [CEN prENV 13608]
Public key	Key that is used with an asymmetric cryptographic algorithm and that can be made publicly available. [ISO 10181-1]
Public key certificate **User certificate** **Certificate**	The public keys of a user, together with some other information, rendered unforgeable by encipherment with the private key of the certification authority which issued it. [ISO 9594-8] NOTE such kinds of certificates might be dedicated, on the basis of public key certification techniques, to attributes (i.e., attribute certificate), or digital signatures (i.e., signature certificate).
Public key infrastructure PKI[48]	System of digital certificates, Certificate Authorities, and other registration authorities that verify and authenticate the validity of each party involved in an Internet transaction.
Secret key	Key which is kept securely and only disclosed to parties intended to have access to data protected by it. [TH]
Security	Concept schema consisting of security, safety, and quality of data and procedures.
Security enforcement procedure	Coherent and complete package of organisational, physical or technical rules intended to be used to verify the correct enforcement of the security policy. [CEN ENV 13608]
Security mechanism	A formal specification describing a methodology for implementing a set of security functions to provide security ser-

[48] **public key infrastructure**
an infrastructure with the components below used in the relation between a key holder and a relying party including a Certification Authority that allows a relying party to use a certificate relating to the key holder for at least one application using a public key dependent security service.

- a certificate data structure according to one standard that allows the understanding of those attributes necessary for the application. (Note. the certificate of the key holder may contain additional optional non-critical attributes not understood by the relying party)
- means for the relying party to obtain current information on the revocation status of the certificate. (Note this includes technical means (CRLs or OCSP) and agreements to allow access, possibly at a charge)
- publication of certification policy and a certification practice statement used by the included CAs

a CA or several CAs that issue certificates to the key holder and provide relying parties with revocation information. (Note many PKIs may provide certificates in directories accessible by relying parties but this is not essential for the definition, the CA issues the certificate to the key holder that by some means makes this available to the relying party in the specific application)

vices.

Security policy The set of laws, rules, and practices that regulate how an organisation manages, protects, and distributes sensitive information.
[TCSEC]

Security protocol Formal detailed specification describing the implementation of a set of security functions
[TH]

Security service A service, provided by a layer of communicating open systems, which ensures adequate security of the systems or of data transfers.
[ISO 7498-2]
Service enforcing a security concept within the security policy.

Security subject
Subject Active entity, generally in the form of a person, process or device that causes information to flow among objects or changes the system state. Technically, a process/domain pair
[TCSEC]
NOTE According to the Object-Oriented paradigm, a subject is usually called a principal.

Strong Authentication Authentication by means of cryptographic techniques.
[ISO 7498-2]

System-intrinsic threats Security threats arising from the system.

Third party Party other than data originator, or data recipient, required to perform a security function as part of a communication protocol.
[CEN ENV 13608]

Traceability The property that ensures that events that occur in the system, can be traced by the system.
[CEN ENV 13608]

Trusted third party
TTP Security authority, or its agent, trusted by other entities with respect to security related activities. (In the context of ISO/IEC 13888, a trusted third party is trusted either by the originator, the recipient, and/or the delivery authority for the purposes of non-repudiation, and by another party such as the adjudicator).
[ISO/IEC 13888-1: 1997]
A security authority or its agent, trusted by other principal with respect to security-related activities.

User certificate
Public key certificate
Certificate The public keys of a user, together with some other information, rendered unforgeable by encipherment with the private key of the certification authority which issued it.
[ISO 9594-8]

16 References

[Abrams et al., 1995]
Abrams, M.D., Jajodja, S., Podell, H.J. (edrs.) (1995) Information Security. An Integrated Collection of Essays. IEEE Computer Society Press, Los Alamitos, California.

[Allaert et al., 2002]
Allaert, F.A., Blobel, B., Louwerse, C.P., Barber, B. (edrs.) (2002) Security Standards for Healthcare Information Systems - A Perspective from the EU ISIS MEDSEC Project. Series in Health Technology and Informatics Vol. 27. IOS Press, Amsterdam.

[Aoyama, 1998]
Aoyama, M. (1998) New Age of Software Development: How Component-Based Software Engineering Changes the Way of Software Development. 1998 International Workshop on Component-Based Software Engineering. Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA 1998. http://www.sei.cmu.edu/cbs/icse98

[Arbeitsgemeinschaft, 1996]
AG "Karten im Gesundheitswesen", GMD-Forschungszentrum Informationstechnik GmbH (1996) Multifunktionale KartenTerminals (MKT) für das Gesundheitswesen und andere Anwendungsgebiete. Spezifikation Version 1.0.

[Arbeitskreis, 1997]
AG "Karten im Gesundheitswesen", AK "Health Professional Cards" (1997) Deutscher Modellversuch "HPC". Version Dezember 1997.

[ASTM, 2001]
ASTM E31.20: Privilege Management Infrastructure, Working Draft 0.6, November 2001

[Barber et al., 1996]
Barber, B., Bleumer, G., Davey, J., Louwerse, C.P. (1995) How to Achieve Secure Environment for Information Systems in Medicine, in *MEDINFO '95* (edrs. R.A. Greenes, H.E. Peterson, D.J. Protti), pp 635-639. North-Holland, Amsterdam-London-New_York-Tokyo.

[Barkley et al.]
Barkley, J.F., Kuhn, D.R., Rosenthal, L.S., Skall, M.W., Cincotta, A.V.: Role-Based Access Control for the Web. http://hissa.ncsl.nist.gov/rbac/cals-paper.html.

[Baum-Waidner et al., 1998]
Baum-Waidner, B., Blobel, B., Ottes, F., Louwerse, K., Krohn, R., Bleumer, G. (1998) Security Requirements of a HIS-Architecture. ISHTAR Project HC 1028, Deliverable 23 (Final), February 1998.

[Baur et al., 1996]
Baur, H.J., Saurbier, F., Engelmann, U., Schröter, A., Baur, U., Meinzer, H.-P. (1996) Aspects of Data Security and Privacy in Teleradiology, in *CAR '96, Computer Assisted Radiology, Proceedings of the International Symposium on Computer and Communication Systems for Image Guided Diagnosis and Therapy*. Paris, June 96.

[Beale, 2001]
Beale, T. (2001) An Interoperable Knowlegde Methodology for Future-proof Informations Systems, Revision 2.1 Draft A.
http://www.deepthought.com.au

[Bergner et al., 1998]
Bergner, K., Rausch, A., Sihling, M. (1998) Componentware – The Big Picture. 1998 International Workshop on Component-Based Software Engineering. Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA. http://www.sei.cmu.edu/cbs/icse98

[Blobel, 1993]
B.Blobel: Die Realisierung eines integrierten Krankenhausinformationssystems - pro oder contra HL7? in *Europäische Perspektiven der Medizinischen Informatik, Biometrie und Epidemiologie* (Hrsg. J. Michaelis, G. Hommel, S. Wellek). Medizinische Informatik, Biometrie und Epidemiologie Band 76, 384-387. MMV Medizin Verlag München.

[Blobel, 1996a]
Blobel, B. (1996) Open Information Systems and Data Security in Medicine, in *Towards Security in Medical Telematics* (edrs. B. Barber, A. Treacher and C.P. Louwerse), pp 168-182. Series in Health Technology and Informatics Vol. 27. IOS Press, Amsterdam.

[Blobel, 1996b]
Blobel, B. (1996) Clinical Record Systems in Oncology. Experiences and Developments on Cancer Registries in Eastern Germany, in *Preproceedings of the International Workshop "Personal Information - Security, Engineering and Ethics"* pp 37-54, Cambridge, 21-22 June, 1996, also published in *Personal Medical Information - Security, Engineering, and Ethics* (edr. R.Anderson), pp 39-56. Spinger, Berlin 1997.

[Blobel, 1996c]
Blobel, B. (1996) A Regional Clinical Cancer Documentation System for an Optimal Shared Health Care in Cancer, in *Medical Informatics Europe '96* (edrs. J. Brender, J.P. Christensen, J.-R. Scherrer and P. McNair), pp 1019-1026. Series in Health Technology and Informatics Vol. 34. IOS Press, Amsterdam.

[Blobel, 1996d]
Blobel, B. (1996) Modelling for Design and Implementation of Secure Health Information Systems. International Journal of Bio-Medical Computing **43**, S23-S30.

[Blobel, 1997a]
Blobel, B. (1997) Security Requirements and Solutions in Distributed Electronic Health Records, in *Information Security in Research and Business* (edrs. L.Yngström and J.Carlsen), pp 377-390. Chapman & Hall, London.

[Blobel, 1997b]
Blobel, B (1997) Treats and Solutions for Data Protection and Data Security in Health Care Information Systems. Toward An Electronic Patient Record, Volume 5, Issue 8, pp 1-16.

[Blobel, 2000a]
B.Blobel (2000) Securely Communicating and Co-operating Health Information Systems – The Basis for Shared Care. World Markets Series BUSINESS BRIEFING: Global Health Care, September 2000, pp 35-38.

[Blobel, 2001]
B.Blobel (2001) Data Protection and Data Security in Shared Care Information Systems in *Recent Health Policy Innovations in Social Security* (edrs. A. Ron and X. Scheil-Adlung), pp. 301-334. International Social Security Series, Vol. 5. Transaction Publishers, New Brunswik (U.S.A.) and London (U.K.).

[Blobel and Holena, 1996]
Blobel, B., Holena, M. (1996) Advanced Healthcare System Architecture Using Middleware Concepts - A Comparative Study. Deliverable of the HC 1019 Telematics Project HANSA, July 1996.

[Blobel and Holena, 1997]
Blobel, B., Holena, M. (1997) Comparing middleware concepts for advanced healthcare system architectures. International Journal of Medical Informatics **46**, pp. 69-85.

[Blobel and Holena, 1998]
Blobel, B., Holena, M. (1998) CORBA Security Services for Health Information Systems. International Journal of Medical Informatics **52** 1-3, pp 29-38

[Blobel and Katsikas, 1998]
Blobel, B., Katsikas, S.K. (1998) Patient data and the Internet - security issues. Chairpersons' introduction. International Journal of Medical Informatics **49**, pp. S5-S8

[Blobel and Pharow, 1997a]
Blobel, B., Pharow, P. (1997) Experiences with Health Professional Cards and Trusted Third Party Services Providing Security in Distributed Electronic Records in Oncology. Proceedings of the Conference „Toward An Electronic Health Record Europe '97", pp 29-39. 20-23 October 1997 London.

[Blobel and Pharow, 1997b]
Blobel, B., Pharow, P. (1997) Security Infrastructure of an Oncological Network Using Health Professional Cards, in *Health Cards '97* (edrs. L.van den Broek and A.J. Sikkel), pp 323-334. Series in Health Technology and Informatics, Vol. 49.IOS Press, Amsterdam.

[Blobel and Pharow, 1998]
Blobel, B., Pharow, P. (1998) Results of European Projects Improving Security of Distributed Health Information Systems, in *MEDINFO '98* (edrs. B. Cesnik, A.T. McCray, J.-R. Scherrer), 1119-1123. IOS Press Amsterdam, Berlin, Oxford, Tokyo, Washington DC.

[Blobel and Pharow, 1999]
B.Blobel, P.Pharow (1999) Secure Communication and Co-operation in Open Networks, in *3rd IMACS / IEEE CSCC '99 International Multiconference.* Athens, Greece.

[Blobel and Roger-France, 1998]
Blobel, B., Roger-France, F. (1998) Healthcare Security View Based on the Security Services Concept. ISHTAR Project HC 1028, Deliverable, August 1998.

[Blobel and van Eecke, 1999]
Blobel, B., Eecke, P. van (1999) TrustHealth-2 –Impact of rules, regulations, laws and legislation in the demonstrator applications. TrustHealth-2 Project HC 4023, Deliverable 1.2, August 1999.

[Blobel et al., 1997]
Blobel, B., Bleumer, G., Müller, A., Louwerse, K., Flikkenschild, E., Ottes, F. (1997) Current Security Issues Faced by Health Care Establishments. ISHTAR Project HC 1028, Deliverable 09 (Final), February 1997.

[Blobel et al., 1998a]
Blobel, B., Spiegel, V., Krohn, R., Pharow, P., Engel, K. (1998) Standard Guide for Specifying EDI (HL7) Communication Security. ISIS MEDSEC Project, Deliverable 30, August 1998, http://www.math.aegean.gr/medsec/d1.html

[Blobel et al., 1998b]
Blobel, B., Spiegel, V., Krohn, R., Pharow, P., Engel, K. (1998) Standard Guide for Implementing EDI Communication Security. ISIS MEDSEC Project, Deliverable 31, August 1998, http://www.math.aegean.gr/medsec/d1.html

[Blobel et al., 1999]
Blobel, B., Pharow, P., Engel, K., Spiegel, V., Krohn, R. (1999) Communication Security

in Open Health Care Networks, in *Medical Informatics Europe '99* (edrs. P. Kokol, B. Zupan, J. Stare, M. Premik, R. Engelbrecht), pp 291-296. Series in Health Technology and Informatics Vol. 68. IOS Press, Amsterdam.

[BMG, 1995]
BMG: Bericht der Weisen 1995

[Böhm et al., 1997]
Böhm, V., Engelbrecht, R., Sulzmann, R., Moser, W., Sembritzki, J. (1997) The DIABCARD Server Architecture, in *Health Cards '97* (edrs. L.van den Broek and A.J. Sikkel), pp 257-261. Series in Health Technology and Informatics, Vol. 49.IOS Press, Amsterdam.

[Booch, 1994]
Booch, G. (1994) Object-oriented Analysis and Design. With Applications. Second Edition. Addison-Wesley Publishing Company, Reading, NY.

[Brannigan and Beier, 1995]
Brannigan, V.M., Beier, B.R. (1995) Patient Privacy in the Era of Medical Computer Networks: A New Paradigm for a New Technology, in *MEDINFO '95* (edrs. R.A. Greenes, H.E. Peterson, D.J. Protti), pp 640-643. North-Holland, Amsterdam-London-New York-Tokyo.

[Brown, 1996]
Brown, A. W. (1996) Component-Based Software Engineering, IEEE CS Press.

[Brown, 1998]
Brown, A. (1998) From Component Infrastructure to Component-Based Development. 1998 International Workshop on Component-Based Software Engineering. Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA.
http://www.sei.cmu.edu/cbs/icse98

[BSI, 1995]
Bundesamt für Sicherheit in der Informationstechnik (1995) IT-Grundschutzhandbuch - Massnahmenempfehlung für den mittleren Schutzbedarf. Schriftenreihe zur IT Sicherheit, Bundesanzeiger Verlag GmbH, Germany.

[Castano et al., 1995]
Castano, S., Fugini, M., Martella, G., Samarati, P. (1995) Database Security. Addison-Wesley Publishing Company, Wokingham.

[CEN, 1995]
CEN(1995). Healthcare Information System Architecture. Draft European Prestandard prENV 1995-12-12, Project Team PT1-013, Comité Européen de Normalisation, Brussels, Belgium.

[CENTC215_WWW]
http://www.centc215.org

[CM, 1997]
Committee of Ministers (1997) European Recommendation No. R(97)5 of the Committee of Ministers to Member States on the Protection of Medical Data (and Genetic Data). CJ-PD (97). Strasbourg.

[Cooper,1996]
Cooper, P. (1996). Integration: An Analysis of the Integration Methods in the Context of the Electronic Patient Record, in *Proceedings of the Symposium Toward An Electronic Health Record Europe '96*, London, UK, pp. 202-206.

[CE, 1993]
Council of Europe (1993) Infosec Business Advisory Group. The IBAG Framework for Commercial IT Security, V2.0. Frankfurt.

[CE, 1995]
Council of Europe (1995) Directive 95/46/EC, On the Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of such Data (OJ L281/31-50, 24 October 1995). Strasbourg.

[CE, 1999]
Council of Europe (1999) Council of Europe 99/93/EC: Directive on Electronic Signatures. Strasbourg.

[CE; 2000]
Council of Europe (2000) Council of Europe 2000/31/EC: Directive on certain legal aspects of E-commerce

[CERT]
CERT Co-ordination Centre (Computer Emergency Response Team): Security Improvement, Tech Tips, Alerts and Advisories.
http://www.cert.org/nav/securityimprovement.html,
ftp://ftp.cert.org/pub/tech_tips/FTP_PORT_attacks,
http://www.cert.org/advisories/CA-97.27.FTP_bounce.html,
http://www.cert.org/advisories/
CA-95.01.IP.spoofing.attacks.and.hijacked.terminal.connections.html).

[CHIME_WWW]
CHIME_UCL: University College of London. Electronic Healthcare Support Action.
http://www.chime.ucl.ac.uk/HealthI/EHCR-SupA

[CORBA, 2000]
CORBA Revised Specifications
http://www.omg.org

[CORBA_COAS, 2001]
CORBA Clinical Observations Access Service Specification
http://www.omg.org

[CORBA_LQS, 2000]
CORBA Lexicon Query Service Specification, Version 1.0
http://www.omg.org

[CORBA_PIDS, 2001]
CORBA Person Identification Service Specification, Version 1.1
http://www.omg.org

[CORBA_RADS, 2001]
CORBA Resorce Access Decision Facility Specification, Version 1.0
http://www.omg.org

[CORBA_SIS, 2001]
CORBA Common Secure Interoperability V2 Specification
http://www.omg.org

[CORBA_SSS, 2001]
CORBA Security Service Specification V 1.7
http://www.omg.org

[CTR, 1996]
CTR Report No 8 (1996) Security Issues for the Internet and the World Wide Web. Computer Technology Research Corp., Charleston.

[Cutter, 1999]
Cutter's Consortium (1999) Executive Summary. Executive Report, Vol. 2, No. 2

[Deibel and Greenes, 1995]
Deibel, S.R.A., Greenes, R.A. (1995). Arachne: A Toolset for the Development of Clinical Workstations from Distributed Components. Technical Report DSG-AR-1.1.

[Deibel and Greenes, 1996]
Deibel, S.R.A., Greenes, R.A. (1996). Constructing Advanced Health Care Systems from CORBA components. Focus **3-4**, pp. 16-18.

[Demmer et al., 1998]
Demmer, T., Neufeld, E., Steyer, A. (1998) Diabcard Server Description DCS_1998, Version 1.01, 30.10.1998, ACG-SmartGate Software GmbH, Köln.

[Der Deutsche Bundestag, 1997]
Der Deutsche Bundestag: IuKGD (1997) The German „Information and Communication Services Law" including the „Digital Signature Law". http://www.iukdg.de

[Der Deutsche Bundestag, 2001a]
Der Deutsche Bundestag (2001) The German Law Governing Framework Conditions for Electronic Signatures and Amending Other Regulations (Gesetz über Rahmenbedingungen für elektronische Signaturen und zur Änderung weiterer Vorschriften – SigG): English version, May 16th, 2001.
http://www.iid.de/iukdg/gesetz/engindex.html

[Der Deutsche Bundestag, 2001b]
Der Deutsche Bundestag (2001) The German Electronic Signature Ordinance (Signaturverordnung – SigV): English version, November 2001.
http://www.iid.de/iukdg/gesetz/engindex.html

[DIABCARD_WWW]
DIABCARD - Improved Communication in Diabetes Care Based on Chipcard Technology.
http://www-mi.gsf.de/diabcard

[Diffie and Hellman, 1976]
Diffie, W., Hellman, M. (1976) New directions in cryptography. IEEE Transactions on Information Theory, v. IT-22, n.6, Nov 1976, pp. 644-654.

[DIN_WWW]
http://www.din.org

[Dubey and Chuch, 2000]
Dubey, A.K., Chuch, H.C. (2000) An XML-based Format for Guideline Interchange and Execution, in *Proceedings of the AMIA Annual Symposium 2000*, 8(1), pp. 205-209.

[EC, 1991]
European Commission (1991) European Communities - Commission: ITSEC: Information Technology Security Evaluation Criteria; (Provisional Harmonised Criteria, Version 1.2, 28 June 1991). Office for Official Publications of the European Communities, Luxembourg, 1991.
http://www.itsec.gov.uk/docs/pdfs/formal/ITSEC.PDF

[EC, 1994]
Commission of the European Communities (CEC): Information Technology Security Evaluation Manual, ITSEM, Brussels and Luxembourg 1994, DGXIII, ISBN 92-826-7087-2.

[EC, 1998]
European Commission (1998) Resolution on the communication from the Commission to the Council, the European Parliament, the Economic and Social Committee and the Committee of the Regions on "A European initiative in electronic commerce", A4-0173/98, Resolution of 14 April 1998.

[EC, 1999]
European Commission (1999) Proposal for a European Parliament and Council Directive on certain legal aspects of electronic commerce in the internal market, COM (1998) 586 final, 98/0325 (COD), O.J. 1999/C 30/04.

[ECMA-219]
Standardising Information and Communication Systems. Authentication and Privilege Attribute Security Application with related Key Distribution Functions – Part 1, 2 and 3.

[Elkin et al., 2000]
Elkin, P.L., Peleg, M., Lacson, R., Bernstam, E., Tu, S.W., Boxwala, A., Greenes, R., Shortliffe, E.H. (2000) Toward Standardization of Electronic Guideline Representation. MD Computing **17** (6), pp. 39-44.

[Ellsässer and Köhler, 1993]
Ellsässer, K.-H., Köhler, C.O. (1993) Shared Care: Concept of a Distributed Care - Short and Long Time Perspectives in Europe (in German). Informatik, Biometrie und Epidemiolo-gie in Medizin und Biologie **24**, H.4, S. 188-198.

[Engelbrecht et al., 1997]
Engelbrecht, R., Böhm, V., Hildebrand, C., Moser, W., Landgraf, R., Hierl, F., Töppel, S., Blobel, B., Diedrich, T. (1997) ByMedCard - An Electronic Patient Record with Chip Card Functionality, in *Health Cards '97* (edrs. L. van den Broek, A.J. Sikkel), pp 313-317. Series in Health Technology and Informatics Vol. 49. IOS Press, Amsterdam.

[Eriksson and Penker, 1998]
Eriksson, H.-E., Penker, M. (1998) UML Toolkit. John Wiley & Sons, Inc., New York.

[ETSI, 2001]
ETSI TS 101 903: XML Advanced Electronic Signatures (XAdES), Draft V 0.0.8, 2001-07

[EUROMED_WWW]
EUROMED-ETS - Trusted Third Party Services for Health Care in Europe. Project Description, Partners, Deliverables, Work Items. http://euromed.ece.ntua.gr

[Ferrara, 1995a]
Ferrara, F.M. (1995) DHE General overview. Gestione Sistemi per l'Informatica, Rome, Italy.

[Ferrara, 1995b]
Ferrara, F.M. (1995) HANSA Project Highlights, Version 1.0. Gestione Sistemi per l'Informatica, Rome, Italy.

[Ferrara, 1995c]
Ferrara, F.M. (1995) Specification of the DHE middleware, Version 1.0. Gestione Sistemi per l'Informatica, Rome, Italy.

[Ferrara, 1995d]
Ferrara, F.M. (1995) The open architecture for healthcare information systems, Version 1.0. Gestione Sistemi per l'Informatica, Rome, Italy.

[Ferrara, 1996]
Ferrara, F.M. (1996) The DHE middleware information view, Version 1.1. Gestione Sistemi per l'Informatica, Rome, Italy.

[Ferrara and Sottile, 1995]
Ferrara, F.M., and Sottile, P.A. (1996) The DHE middleware Functional view, Version 0.5. Gestione Sistemi per l'Informatica, Rome, Italy.

[Flikkenschild et al., 1996]
Flikkenschild, E., Sluijs, P.v.d., Buis, E., Verhage, J. (1996) SIDERO: a relational Database Application for Security Practitioners, supporting the implementation of SEISMED guidelines in Health Care Institutions. AIM SEISMED Deliverable. Leiden 1996.

[Ford, 1994]
Ford, W. (1994) Computer Communications Security. PTR Prentice Hall, Englewood Cliffs, New Jersey.

[Forslund, 1996]
Forslund, D. (1996). TeleMed. Los Alamos National Laboratory, University of California, Los Alamos, New Mexico.

[Forslund and Kilman, 1996]
Forslund, D., Kilman, D. (1996) The Virtual Patient Record: A Key to Distributed Healthcare and Telemedicine. *Focus* **3-4**, pp. 14–16.

[Fox and Rahmanzadeh, 1998]
Fox, J., Rahmanzadeh, A. (1998) Disseminating medical knowledge: the PROforma approach. Artificial Intelligence in Medicine **14**, pp. 157-181.

[Frost and Allen, 1997]
Frost, S., Allen, P. (1997) Businessorientierte Komponenten-Modellierung. OBJEKTspektrum **3**, S. 32-39.

[Garets, 2001]
Garets, G. (2001) Gartner's Vision for Healthcare: The Next 10 Years. Presentation at the HL7 Plenary and Working Group Meeting in Orlando, FL.

[G-CPR_WWW]
Veterans Affairs: Governmental Computerised Patient Record, Washington DC, USA
http://www.gcpr.gov

[GEHR_WWW]
Good Electgronic Health Record Australia
http://www.gehr.org

[George, 1996]
George, J. (1996) TeleMed Virtual Patient Record System. Los Alamos National Laboratory, University of California, Los Alamos, New Mexico.

[GNDS_AG]
GMDS Memorandum "Anwenderkriterien zur Einführung von Karten im Gesundheitswesen" der AG "Anwenderkriterien" der GMDS e.V., internal materials

[Gogou et al., 1998]
Gogou G, Mavromatis A, Maglaveras N, Pappas C, Engelbrecht R (1998) Demonstration of the regional DIABCARD System: Implementation Site Thessaloniki – DIABCARD Core System DCC_1998 Technical Report, DIABCARD Deliverable 10.3, Public Report, July 1998

[Gordon and Veloso, 1999]
Gordon, C., Veloso, M. (1999) Guidelines in Healthcare: the experience of the Prestige project, in *Medical Informatics Europe '99* (edrs. P. Kokol, B. Zupan, J. Stare, M. Premik, R. Engelbrecht), pp 733-738. Series in Health Technology and Informatics Vol. 68. IOS Press, Amsterdam.

[Greenes et al, 1999]
Greenes, R.A., Boxwala, A., Sloan, W.N., Ohno-Machado, L., Deibel, S.R. (1999) A framework and tools for authoring, editing, documenting, sharing, searching, navigating, and executing computer-based clinical guidelines, in *Proceedings of the AMIA Annual Symposium 1999*, pp. 261-265.

[Grosso et al., 1999]
Grosso, W.E., Eriksson, H., Fergerson, R., Gennari, J.H., Tu, S.W., Musen, M.A. (1999) Knowledge Modeling at the Millenium (The Design and Evolution of PROTÉGÉ-2000), in *The 12th Banff Accquisition for Knowledge-based Systems Workshop* (edrs. B.R. Gains, R. Kremer, M. Musen, Vol. 7-4, pp. 1-36. Banff, Canada

[Han, 1998]
Han, J. (1998) Characterization of Components. 1998 International Workshop on Component-Based Software Engineering. Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA 1998.
http://www.sei.cmu.edu/cbs/icse98

[HANSA_WWW]
HANSA – Healthcare Advanced Networking Systems Architecture.
http://www.ehto.be/projects/hansa

[Heitmann, 2001]
Heitmann, K. (2001) Introduction to XML. HL7 Plenary and Working Group Meeting Salt Lake City, October 2001.

[Hermanns et al., 1999]
Hermanns, J., Jänsch, C., Tensi, T., Toeniessen, F. (1999) Architekturmanagement in Großunternehmen. OBJEKTspektrum **4**, 26-34.

[HL7_WWW]
Health Level Seven, Inc, Ann Arbor, MI, USA
http://www.hl7.org

[Hoelzer et al., 2001]
Hoelzer, S., Schweiger, R.K., Boettcher, H.A., Tafazzoli, A.G., Dudeck, J. (2001) Value of XML in the implementation of clinical practice guidelines – the issue of content retrieval and presentation. Med. Inform. **26**, 2, pp. 131-146.

[HCP-Protocol, 1999]
HPC (1999) The German HPC Specification for an electronic doctor's licence. Version 1.0, July 1999.
http://www.hcp-protocol.de

[Hripcsak et al., 1994]
Hripcsak, G., Ludemann, P., Pryor, T.A., Wigertz, O.B., Clayton, P.D. (1994) Rationale for the Arden Syntax. Computers in Biomedical Research **27** (4), pp. 291-324.

[Hruschka, 1998]
Hruschka, P. (1998) Ein pragmatisches Vorgehensmodel für UML. OBJEKTspektrum 2, 34-45.

[HR-XML_WWW]
Human Resources Consortium
http://www.hr-xml.org

[Hutt et al., 1995]
Hutt, A.E., Bosworth, S., Hoyt, D.B. (edrs.) (1995) Computer Security Handbook. John Wiley & Sons, Inc., New York.

[IETF/RFC 2246]
Transport Layer Security (TLS) Protocol, Version 1.0 [for client authentication using Transport layer security over TCP/IP]

[IETF/RFC 2459]
Internet X.509 Public key Infrastructure Certificate and Certificate Revocation List (CRL) Profile

[IETF/RFC 2527]
Internet X.509 Public key Infrastructure Certificate Policy and Certification Practices ('PKIX framework ')

[IETF/ SSLv3]
IETF: Secure Sockets Layer, version 3

[IPv6]
http://www.innergy.com/foundation/ipv6.shtml
Security Architecture for the Internet Protocol (draft-ietf-ipsec-arch-sec), Internet Draft (Network Working Group), S.Kent, R.Atkinson, July 1998.
http://www.ietf.org/html.charters/ipsec-charter.html

[ISHTAR_WWW]
ISHTAR - Implementing Secure Health Telematics Applications in Europe.
http://www.ehto.be/projects/ishtar

[ISHTAR, 2001]
The ISHTAR Consortium (edr.) (2001) Implementing Secure Health Telematics Applications in Europe. Series Studies in Health Technology and Informatics, Vol. 66. IOS Press, Amsterdam.

[ISOTC215WG4_WWW]
ISO Technical Committee "Health Informatics", WG4 "Security"
http://www.medis.or.jp/iso/tc215wg4.html

[Jacobson et al., 1992]
Jacobson, I., Christerson, M., Jonsson, P., Övergaard, G. (1992) Object-oriented Software Engineering. Addison-Wesley Publishing Company, Reading, NY.

[Jeckle, 2001]
Jeckle, M. (2001) Practical usage of W3C's XML-Schema and a process for generating schema structures from UML models. SSGRR, pp. 1-18.

[JWD-CDM, 1996]
JWG-CDM (1996). Trial-Use Standard for Health Care Data Interchange – Information Model Methods. Data Model Framework. IEEE P1157 Medical Data Interchange Working Group – Joint Working Group for a Common Data Model.

[Katsikas et al., 1998]
Katsikas, S.K., Spinellis, D.D., Iliadis, J., Blobel, B. (1998) Using Trusted Third Parties for secure telemedical applications over the WWW: The EUROMED-ETS approach. International Journal of Medical Informatics **49,** pp. 59-68

[Kluge, 1995a]
Kluge, E.-H.W. (1995) Patients, Patient Records, and Ethical Principles, in *MEDINFO '95* (edrs. R.A. Greenes, H.E. Peterson, D.J. Protti), pp 1596-1600. North-Holland, Amsterdam-London-New York-Tokyo.

[Kluge, 1995b]
Kluge, E.-H.W. (1995) Health information, the fair information principles and ethics, in *Yearbook of Medical Informatics* (edrs. J.H. van Bemmel and A.T. McCray), pp 255-264. Schattauer, Stuttgart.

[Kruchten, 1998]
Kruchten, P. (1998) Modeling Component Systems with the Unified Modeling Language. 1998 International Workshop on Component-Based Software Engineering. Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA.
http://www.sei.cmu.edu/cbs/icse98

[Kuperman, G.J., Gardner, R.M., Pryor, T. (1991) HELP: A Dynamic Hospital Information System. Springer, New York.

[Laske, 1995]
Laske, C. (1995) Legal Issues in Medical Informatics: A Bird's Eye View, in *Towards Security in Medical Telematics – Legal, and Technical Aspects* (edrs. B. Barber, A. Treacher and C.P. Louwerse), pp. 53-78. Studies in Health Technology and Informatics, Vol. 27. IOS Press, Amsterdam.

[Leguit, 1992]
Leguit, F. (1992) Interfacing Integration, in *Hospital Information Systems* (eds. A.R. Bakker et al.), pp. 141-148. North-Holland, Amsterdam.

[MedRecInst_WWW]
Medical Record Institute, Newton, MA, USA
http://www.medrecinst.com

[MEDSEC_WWW]
MEDSEC - Health Care Security and Privacy in the Information Society.
http://www.math.aegean.gr/medsec

[Menezes et al., 1997]
Menezes, A.J., Oorschot, P.C. van, Vanstone, S.A. (1997) Handbook of Applied Cryptography. CRC Press, Boca Raton.

[Musen et al., 1996]
Musen, M., Tu, S.W., Das, A., Shahar, Y. (1996) EON: A component-based approach to automation of protocol-directed therapy.JAMIA **3**, pp. 367-388.

[O, 1999]
O, Y.-L. (1999) A Life-cycle based Authorisation Expert Database System in Artificial

Intelligence in Medicine (eds: W. Horn et al.), pp. 153-157. LNAI Vol. 1620. Springer, Berlin.

[OASIS_WWW]
Organization for the Advancement of Structured Information Standards
http://www.oasis-open.org

[Ohne-Machado et al., 1998]
Ohno-Machado, L., Gennari, J.H., Murphy, S., Jain, N.L., Tu, S.W., Oliver, D.E., Pattison-Gordon, E., Greenes, R., Shortliffe, E.H., Barnett, G.O. (1998) The Guideline Interchange Format: A Model for Representing Guidelines. JAMIA **5** (4), pp. 357-372.

[OMG, 1991]
OMG (1991) Object Services Request for Information. Object Management Group, Inc., Framingham, December 1991.

[OMG, 1994]
OMG (1994) White paper on security. Object Management Group, Inc., Framingham, April 1994.

[OMG, 1995a]
OMG (1995) Common Facilities Architecture. Revision 4.0, Object Management Group, Inc., Framingham, November 1995.

[OMG, 1995b]
OMG (1995) The Common Object Request Broker: Architecture and Specification. Revision 2.0, Object Management Group, Inc., Framingham, Massachusetts.

[OMG, 1995c]
OMG (1995) The CORBA Security Specification. OMG Doc.No. 95-12-01.

[OMG, 1996a]
OMG (1996) CORBA Services: Common Object Services Specification. Revised Edition, Object Management Group, Inc., Framingham, Massachusetts.

[OMG, 1996b]
OMG (1996) CORBAmed Request for Information. Object Management Group, Inc., Framingham, Massachusetts.

[OMG, 1996c]
OMG (1996) CORBAmed White Paper. Revision 1, Object Management Group, Inc., Framingham, Massachusetts.

[OMG, 1996d]
OMG (1996) Patient Identification Services Request for Proposal. Object Management Group, Inc., Framingham, November 1996.

[OMG, 1997a]
OMG (1997) CORBA Services: Common Object Services Specification, Revised Edition. Object Management Group, Inc., Framingham, November 1997.

[OMG, 1997b]
OMG (1997) Lexicon Query Services Request for Proposal. Object Management Group, Inc., Framingham, January 1997.

[OMG, 1997c]
OMG (1997) The CORBA Security Specification. Framingham: Object Management Group, Inc., 1995, 1997

[OMG, 1998a]
OMG (1998) The Common Object Request Broker: Architecture and Specification, Revision 2.2. Object Management Group, Inc., Framingham, February 1998.

[OMG, 1998b]
OMG (1998): Business Object Component Architecture (BOCA). Object Management Group, Inc., Framingham, Massachusetts.

[OMG COSS14]
OMG (1997) Common Object Services Specification: Chapter 14: Time Services

[OMG COSS15]
OMG (1997) Common Object Services Specification: Chapter 15: Security Services

[Overhage, J., Mamlin, B., Warvel, J., Tierney, W., McDonald, C. (1995) A tool for provider interaction during patient care: G-CARE, in *Proceedings of the AMIA Annual Symposium 1995*, pp. 178-182.

[Patel and Kantzavelou, 1995]
Patel, A., Kantzavelou, I. (1995) Implementing Network Security in Health Care Information Systems, in *MEDINFO '95* (edrs. R.A. Greenes, H.E. Peterson, D.J. Protti), pp 671-674. North-Holland, Amsterdam-London-New York-Tokyo.

[Pharow and Blobel, 1999]
Pharow, P., Blobel, B. (1999) Trusted Third Party Services for Internet Security in *Recent Advances in Signal Processing and Communications* (edr. N.E. Mastorakis), pp 379-385. World Scientific and Engineering Society Press.

[Pommerening_WWW]
Pommerening, K.: http://www.uni-mainz.de/~pommeren/

[Port, 1998]
Port, D. (1998) Unification of Components and Objects Through Abstractions. 1998 International Workshop on Component-Based Software Engineering. Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA. http://www.sei.cmu.edu/cbs/icse98

[ProtStack]
http://www.med2000.com:457/NetAdminG/netoverN.stack.html
http://www.chips.ibm.com/techlib/micronews/vol3_no1/kellow.htm

[Pyarali et al., 1996a]
Pyarali, I., Harrison, T.H., Schmidt, D.C. (1996). An Object-Oriented Framework for High-Speed Medical Imaging. Focus **3-4**, pp. 18-21.

[Pyarali et al., 1996b]
Pyarali, I., Harrison, T.H., Schmidt, D.C. (1996) Design and Performance of an Object-Oriented Framework for High-Speed Electronic Medical Imaging in *Proceedings of the USENIX COOTS Conference*, Toronto, June 1996.

[Quatrani, 1998]
Quatrani, T (1998) Visual Modeling with Rational Rose and UML. Addison-Wesley, Reading.

[RESHEN, 2002]
RESHEN Project IST-2000-25354 Deliverable 2.2: Legal and policy issues of PKI adoption in health telematics applications in Greece, Germany and Finland (ed. Z. Kardasiadou).

[Rishel, 1996]
Rishel, W. (1996) HL7 with CORBA and OLE: Software Components for Healthcare. AMIA, Inc., 0195-4210/96, pp. 95-99.

[Rumbaugh et al., 1991]
Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorensen W. (1991) Object-Oriented Modeling and Design. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

[Saleck, 1997a]
Saleck, T. (1997) Komponenten und Skalierbarkeit: Softwareteile in allen Grössen. DATENBANK FOKUS **4**, S. 20-27.

[Saleck, 1997b]
Saleck, T. (1997) Entwurfsmuster als Firmen-Paradigma: Keine blossen IT-Konstrukte. Objekt Fokus **3**, S. 8-15.

[Sanders et al., 2000]
Sanders, G., Nease, R., Owens, D. (2000) Design and pilot evaluation of a system to develop computer-based site-specific practice guidelines from decision models. Medical Decision Making **20** (2), pp. 145-159.

[Schadow et al., 1998]
Schadow, G., Tucker, M., Rishel, W. (1998) Secure HL7 Transactions using Internet Mail (draft-ietf-ediint-hl7), Internet Draft (EDIINT Working Group), July 21, 1998. http://www.ietf.org/internet-drafts.

[Siegel, 2001]
Siegel, J. (2001) Quick CORBA® 3. John Wiley & Sons, New York, Chichester, Weinheim, Brisbane, Singapore, Toronto.

[Schiffman and Nath, 2000]
Schiffman, R., Nath, S. (2000) A Preliminary Evaluation of Guideline Content Mark-up Using GEM – An XML Guideline Element Model, in *Proceedings of the AMIA Annual Symposium 2000.*

[Schneier, 1996]
Schneier, B. (1996) Applied Cryptography. John Wiley & Sons, Inc., New York.

[Schweiger, 2002]
Schweiger, R. (2002) XML Topic Maps. Workshop "XML – Advanced Concepts". 22-23 April 2002, Rauischholzhausen, Germany.
http://www.hl7.de

[SEISMED, 1996]
The SEISMED Consortium (edr.) (1996) Data Security for Health Care. Series of Studies in Health Technology and Informatics Vols. 31 - 33. IOS Press, Amsterdam.

[Selic and Rumbaugh, 1998]
Selic, B., Rumbaugh, J. (1998) Die Verwendung von UML für die Modellierung komplexer Echtzeitsysteme. OBJEKTspektrum **4**, 24-36.

[Siegel, 2001]
Siegel, J. (2001) Quick CORBA® 3. John Wiley & Sons, Inc., New York, Chichester, Weinheim, Brisbane, Singapore, Toronto.

[SSL]
http://www.rsa.com/rsalabs/faq/html/5-1-2.html
http://home.netscape.com/eng/ssl3/ssl-toc.html

[Sugden et al., 1999]
Sugden, B., Purves, I.N., Booth, N., Sowerby, M. (1999) The PRODIGY Project – the Interactive Development of the Release One Model, in *Proceedings of the AMIA Annual Symposium 1999*, pp. 359-363.

[Stallings, 1995]
Stallings, W. (1995) Network and Internet Security. Principles and Practice. Prentice Hall, Hemel Hempstead.

[Stuart, 2001]
Stuart, I. (2001) XML Schema, a brief introduction.
http://lucas.ucs.ed.ac.uk/xml-schema/

[Sulzmann, 1998]
Sulzmann, R. (1998) DiabCard Interface Description DDATA_1998, Version 1.6, 19.05.1998, IBM Deutschland Entwicklung GmbH, Böblingen.

[TIMEPROOF_WWW]
timeproof® Time Signature Systems Hamburg
http://www.timeproof.de

[TLS]
The TLS Protocol Version 1.0 (draft-ietf-tls-protocol), Internet Draft (Transport Layer Security Working Group), T.Dierks, C.Allen, November 1997.
http://www.ietf.org/html.charters/tls-charter.html

[TRUSTHEALTH_WWW]
TRUSTHEALTH - Trustworthy Health Telematics. Project Description, Partners, Deliverables, Work Items.
http://www.ehto.be/projects/trusthealth.

[Varadharajan and Hardjono, 1996]
Varadharajan, V., Hardjono, T. (1996) Security Model for Distributed Object Framework and its Applicability to CORBA, in *Information Systems Security* (edrs. S.K. Katsikas and D. Gritzalis), pp. 452-463.Chapman & Hall, London.

[Velde, 1992]
Velde, R. van de (1992) Hospital Information Systems - The Next Generation. Springer-Verlag, Berlin.

[Velde, 2000]
Velde, R. van de (2000) Framework for a clinical information system. International Journal of Medical Informatics **57,** pp. 57-72

[Veluwen, 1999]
Veluwen, A. van (1999) Komponentenarchitekturen. OBJEKTspektrum **4** (1999), 36-40.

[Vlist, 2002]
Vlist, E, van der (2002) Relax NG, Compared. Published on XML.com
http://www.xml.com/pub/a/2002/01/23/relaxing.html

[W3C_WWW]
World Wide Web Consortium
http://www.org

[Walsh, 2001]
Walsh, N. (2001) Understanding XML Schemas.
http://www.xml.com./

[Weed, 1970]
Weed, L.L. (1970) Medical Records, Medical Education and Patient Care - The Problem-Oriented Records as a Basic Tool. The Press of Case Western Reserve University, Chicago.

[Weed, 1978]
Weed, L.L. (1978) Das problemorientierte Krankenblatt. Schattauer, Stuttgart.

[Winter and Haux, 1995]
Winter, A. and Haux, R. (1995) A three level graph-based model for the management of hospital information systems. Meth. Inform. Med. **34,** pp. 378-396.

[Woolf et al., 1999]
Woolf, S.H., Grol, R., Hutchinson, A., Eccles, M., Grimshaw, J. (1999) Potential benefits, limitations, and harms of clinical guidelines. Brit. Medical Journal **318**, pp. 527-530.

[Wunsch, 1986]
Wunsch, G. (1986) Handbuch der Systemtheorie. Akademie-Verlag, Berlin.

[XBRL_WWW]
XBRL International. eXtensible Business Reporting Language Organization
http://www.xbrl.org

17 Annex A: Normative References

[ASTM E1986-98]
ASTM E1986-98, Data User Role Name

[CEN ENV 12924]
CEN ENV 12924, Medical Informatics – Security Categorisation and Protection for Healthcare Information Systems.

[CEN ENV 13606]
CEN ENV 13606, Health informatics – Electronic healthcare record communication, 1999
Part 1: Extended Architecture
Part 2: Domain Term List
Part 3: Distribution Rules
Part 4: Messages for the Exchange of Information.

[CEN ENV 13608]
CEN ENV 13608, Health informatics - Security for healthcare communication -
Part 1: Concepts and terminology
Part 2: Secure data objects
Part 3: Secure data channels

[CEN ENV 13729]
CEN ENV 13729, Health informatics – Secure user identification — Strong authentication using microprocessor cards (SEC-ID/CARDS), 1999.

[ISO/IEC PDTR 14516]
International Standards Organization: Information technology - Security techniques - Guidelines on the use and management of TTP services

[ISO/IEC 2382]
International Standards Organization: Information technology – Vocabulary
Part 8 – Security

[ISO 7498-2]
International Standards Organization: Information processing systems, Open Systems Interconnection, Basic Reference Model
Part 2: Security Architecture.
Note: ISO 7498-2 is superseded by ISO/IEC 10745 (ITU-T X.803), ISO/IEC 13594 - IT-Lower layers security (ITU-T X.802) and ISO/IEC 10181-1 (ITU-T X.810).

[ISO/IEC 7816]
International Standards Organization: Information technology - Identification cards, Integrated circuit(s) cards with contacts –
Part 1 – Physical characteristics
Part 2 – Dimensions and location of the contacts
Part 3 – Electronic signals and transmission protocols
Part 4 – Interindustry commands for interchange
Part 5 – Numbering system and registration procedure for application identifiers
Part 6 – Interindustry data elements
Part 7 – Interindustry commands for Structured Card Query Language (SCQL)
Part 8 – Security related interindustry commands
Part 9 – Further interindustry commands (working draft)
Part 10 – Electronic signals and answer to reset for synchronous cards
Part 11 – Card structure and enhanced functions for multi-application use (working draft)

[ISO 9564]
International Standards Organization: Banking – Personal Identification Number management and security –
Part 1– PIN protection principles and techniques
Part 2 – Approved algorithm(s) for PIN encipherment

[ISO/IEC 9594-8]
International Standards Organization: Information technology - Open Systems Interconnection - The Directory –
Part 8 – Authentication framework
[Note: equiv. to ITU-T/X.509]

[ISO/IEC 9735]
International Standards Organization: Electronic data interchange for administration, commerce and transport (EDIFACT), Application level syntax rules, multiple Parts (1-10).

[ISO/IEC 9796]
International Standards Organization: Information technology, Security techniques, Digital signature scheme giving message recovery, multiple Parts (1-2).

[ISO/IEC 9797]
International Standards Organization: Information technology, Security techniques, Message authentication codes.

[ISO/IEC 9798]
International Standards Organization: Information technology – Security techniques – Entity authentication
Part 1: General
Part 2: Mechanisms using symmetric encipherment algorithms
Part 3: Mechanisms using digital signature techniques
Part 4: Mechanisms using a cryptographic check function
Part 5: Mechanisms using zero knowledge techniques

[ISO/IEC 9979]
International Standards Organization: Information technology, Security techniques, Procedures for the registration of cryptographic algorithms.

[ISO/IEC 10118]
International Standards Organization: Information technology - Security techniques - Hash-functions –
Part 1: General
Part 2: Hash-functions using an n-bit block cipher algorithm
Part 3: Dedicated hash-functions
Part 4: Hash-functions using modular arithmetic

[ISO/IEC 10164-16]
International Standards Organisation: Information technology, Open Systems Interconnection, Extension for General Relationship Model.

[ISO/IEC 10165-7]
International Standards Organisation: Information technology, OpenSystems Interconnection, General Relationship Model (see also ISO/IEC 10164-16).

[ISO/IEC 10181]
International Standards Organization: Information technology - Open Systems Interconnection - Security frameworks for open systems –
Part 1: Overview [equivalent to ITU-T Rec. X.810]
Part 2: Authentication framework [X.811]
Part 3: Access control framework [X.812]
Part 4: Non-repudiation framework [X.813]
Part 5: Confidentiality framework [X.814]
Part 6: Integrity framework [X.815]
Part 7: Security audit and alarms framework [X.816]

[ISO 10202]
International Standards Organization: Financial transaction cards -Security architecture of financial transaction systems using integrated circuit cards -
Part 6. Cardholder verification

[ISO/IEC 10736]
International Standards Organization: Information technology, Telecommunications and information exchange between systems, Transport layer security protocol.

[ISO/IEC 10745]
International Standards Organization: Information technology, Open Systems Interconnection, Upper layers security model.

ISO/IEC 10746-2]
International Standards Organization: Information Technology – Open Distributed Processing – Reference Model: Part 2: Foundations.

[ISO/IEC 11577]
International Standards Organization: Information technology, Open Systems Interconnection, Network layer security protocol.

[ISO/IEC 11586]
International Standards Organization: Information technology, Open Systems Interconnection, Generic upper layers security, multiple Parts (1-6).

[ISO/IEC 13594]I
International Standardization Organization: Information technology, Lower layers security.

[ISO/IEC 13888]
International Standards Organization: Information technology – Security techniques – Non-repudiation
Part 1: General
Part 2: Mechanisms using symmetric techniques
Part 3: Mechanisms using asymmetric techniques

[ISO/IEC 14888]
International Standards Organization: Information technology, Security techniques, Digital signature with appendix, multiple Parts (1-3).

[ISO/IEC 15408]
International Standards Organisation: IS Information Technology –Evaluation Criteria for IT Security

[ISO 17090]
International Standards Organisation: DTS Health Informatics – Public Key Infrastructure,
Part 1: Framework and overview
Part 2: Certificate profile
Part 3: Policy management of certification authority

[ISO/IEC 17799]
International Standards Organisation: Information technology — Code of practice for information security management.

[ITU-T/X.509]
see ISO/IEC 9594-8

[PKCS]
PKCS: RSA Labs, Public key Cryptography Standard
PKCS#1: RSA Encryption Standard. Version 2.0, October 1998]
(equiv. to RFC2427)
PKCS#7: Cryptographic Message Syntax Standard. Version 1.6, May 1997
PKCS#11: Cryptographic Token Interface Standard, Version 2.01,
PKCS#15: Cryptographic Token Information Format Standard, Version 1.0, April 1999

[SS 62 43 30]
Identification Cards - Electronic ID Application. Swedish Standard 62 43 30: 1998

18 Annex B: List of Abbreviations

ADT	German Data Exchange Specification for administrative data
ANSI	American National Standards Institute
ASN.1	Abstract Syntax Notation 1
AuthCert	Authentication Certificate
BDT	German Data Exchange Specification for care-related data
CA	Certification Authority
CAD	Card Accepting Device
CORBA	Common Object Request Broker Architecture
CRL	Certificate Revocation List
DES	Data Encryption Standard
DN	Distinguished Name
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
EDI	Electronic Data Interchange
EDIFACT	EDI for Administration, Commerce and Transport
EDI-MS	EDI Messaging System (ITU-T X.435, part of MHS)
EHCR	Electronic Healthcare Record
EHR	Electronic Health Record
EIC	Electronic Identity Card
EPR	Electronic Patient Record
ESS	Enhanced Security Services (Part of S/MIME Version 3)
FIPS PUB	Federal Information Processing Standards Publication (by NIST)
FTAM	File Transfer, Access and Management
FTP	File Transfer Protocol
GRM	General Relationship Model
HCE	Healthcare Establishment
HCP	Health Care Professional
HL7	Health Industry Level 7 Interface Standard
HLLP	Hybrid Lower Layer Protocol
HP	Health Professional
HPC	Health Professional Card
HR-XML	Human Recources XML Consortium
HTML	Hypertext Markup Language
IBAG	Infosec Business Advisory Group
IC	Integrated Circuit
ICC	Integrated Circuit Card
ICT	Information and Communications Technology
IDEA	International Data Encryption Algorithm
IESG	Internet Engineering Steering Group

IESG	Internet Engineering Steering Group
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPSEC	IP Security
ISO/IEC	International Standard Organisation
ISO/IEC	International Standard Organisation/International Electrotechnical Commission
IT	Information Technology
ITU	International Telecommunication Union
ITU-T	International Telecommunication Union (formerly the CCITT)
L2F	Layer 2 Forwarding
L2F	Layer 2 Forwarding
L2TP	Layer 2 Tunnelling Protocol
LDT	German Data Exchange Specification for laboratory data
LLP	Lower Layer Protocol
MAC	Message Authentication Code
MD5	Message Digest 5
MHS	Message Handling system
MIME	Multipurpose Internet Mail Extension
ML	Markup Language
MLLP	Minimal Lower Layer Protocol
MOSS	MIME Object Security Services
MSP	Message Security Protocol (NIST SDNS)
NIST	National Institute of Science and Technology
NLSP	Network Layer Security Protocol
NRD	Non-Repudiation of Delivery
NRO	Non-Repudiation of Origin
NRR	Non-Repudiation of Receipt
NRS	Non-Repudiation of Submission
NRT	Non-Repudiation of Transport
OASIS	Organization for the Advancement of Structured Information Standards
ODP	Open Distributed Processing
OID	Object Identifier
OMG	Object Management Group
OMT-2	Object modelling Technique 2
OSI	Open Systems Interconnection
PC	Personal Computer
PCT	Private Communications Technology
PEM	Privacy Enhanced Mail
PGP	Pretty Good Privacy
PIN	Personal Identity Number
PKCS	Public key Cryptography Standard
PKI	Public key Infrastructure

PKIX	Public key Infrastructure X (Internet PKI Standard)
PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunnelling Protocol
RFC	Request For Comments
RIM	Reference Information Model
RM	Reference Model
RND	Random Number
RSA	Rivest, Shamir & Adelman (originators of RSA algorithm)
SAX	Simple API for XML
S/MIME	Secure/MIME
SDE	Secure Data Exchange
SDNS	Secure Data Network System
SFTP	Secure File Transfer Protocol
SGML	Standard Generalized Markup Language
SHA-1	Secure Hash Algorithm 1
SHTTP	Secure HyperText Transfer Protocol
SILS	Standard for Interoperable LAN Security
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SOCKS	Sockets Secure Protocol
SPKM	Simple Public key GSS-API Mechanism
SPRI	Swedish Institute for Health Services Development
SSH	Secure Shell
SSL	Secure Sockets Layer
SSPI	Microsoft Security Support Provider Interface
TCP/IP	Transport Communication Protocol / Internet Protocol
TLS	Transport Layer Security
TLSP	Transport Layer Security Protocol
TTP	Trusted Third Party
TVP	Time Variant Parameter
UML	Unified Modelling Language
XBRL	eXtensible Business Reporting Language
xDT	Set of German Data Exchange Specifications (e.g. ADT, BDT, LDT)
XKMS	XML Key Management Specification
XMI	XML Metadata Interchange
XML	Extensible Markup Language
XPath	XML Path Language
XSD	XML Schema Definition Language
XSL	Extensible Stylesheet Language
XSL-FO	XSL Formatting Objects
XSLT	XSL Transformation
XTM	Topic Map

19 Annex C: TrustHealth-2 Pilot - Requirements and Solutions for the Secure ONCONET Magdeburg/Saxony-Anhalt

The next two chapter concern technical details of practically implemented solutions a technician might be interested in. Although the presented components are running within the German ONCONET created by the author's department, neither warranties will be given nor responsibilities will be taken for implementations and problems occuring elsewhere.

19.1 Cancer Centre Magdeburg

The Cancer Centre Magdeburg / Saxony-Anhalt (TZM) is an entity of all cancer-care-related partners in the entire region. The centre as a members' organisation is responsible for the education and training processes for physicians and non-physicians involved in the care process. As a legal entity, the TZM has prepared rules and regulations for the work. Within TH-2, the TZM acted as an RA for non-physicians and was responsible for request forms' handling, card distribution and card handling, etc.

19.2 Health Professionals of other clinics

The users of the TrustHealth-2 results of both the pilots and the other WPs were medical as well as non-medical staff belonging to different clinics and hospitals inside and outside the UHM. They work as physicians, as documentary staff, as researchers, or as technicians.

The hospitals that have been invited to participate actively in the process of implementing, demonstrating, and validating the TrustHealth-2 security infrastructure including the related security services (TTP, certificates, directories, cards, terminals, etc.) have been mentioned and described in a more detailed manner in TrustHealth-2 deliverables. At the beginning, clinics already having an on-line access to the register were the favourites. But the project was of course not limited to them.

At the end of the realisation period there will be involved users from several hospitals and clinics of the federal state of Saxony-Anhalt using the medical record system based application GTDS.

19.3 TTP (CA) providers

Within the German legislation on communication services and the fundamental law on Digital Signature the framework of security infrastructure needed in the Information Society of multi-purposes and multi-modal communicating and co-operating systems for citizens has been specified. In that context, detailed requirements and recommendations have been mentioned in implementing regulations (SigG, SigV) regarding authorities needed, protocols and forms used, and services defined for a common security infrastructure in e-commerce, healthcare, and any other domain.

The first step was the definition, implementation and accreditation of the German root CA, established in the governmental "Regulierungsbehörde für Telekommunikation und Post" observing the scene after privatising the former governmental post and telecommunication provider. This root CA is fulfilling the strong requirements for security including the physical security according the German data protection and data security legislation. Providing the basis for accreditation further CA but not users, the market for CAs is currently under development. Candidates for the first CAs to be set-up are, e.g., debis, German Telekom, TÜViT, etc.

19.4 Directory software and solutions providers

Supported by the legislation and the security infrastructure framework mentioned in the Chapter above, but also driven by the general development of a global market including telematics, communications, e-commerce etc., German and European providers are increasingly offering security infrastructure services and solutions. This includes facilities for centralised and decentralised key generation, key issuing services, directory services and related solutions like certificate revocation handling.

Providers dealing with such services are, e.g., ControlData and SNI from Germany, iD2 from Sweden, and Baltimore from Ireland. In preparation of the German TH-2 validation site scenario, the Magdeburg Medical Informatics Department was negotiating with the providers mentioned to set up a local directory service.

19.5 Validation Site Hardware and Software Description

19.5.1 The architecture

The description of the architecture used for the Magdeburg and the Halle pilot has to be completed in the future.

Here, a general model of the communication processes and the related functions within the application has been developed and described below.

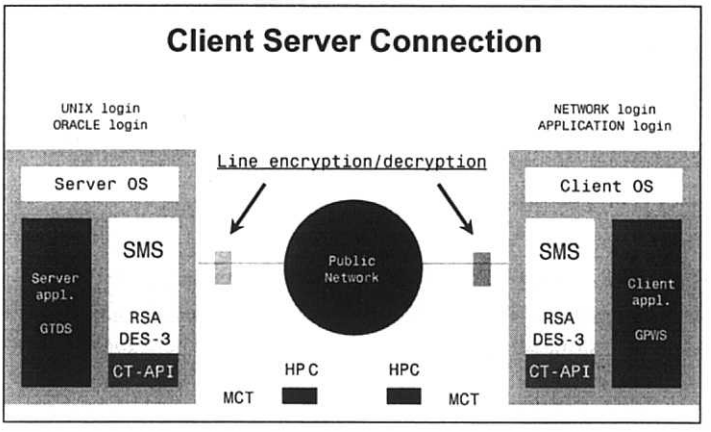


Figure 19.1: Client-Server-Connection

19.5.2 Card operating system STARCOS

The STARCOS® system constitutes a complete operating system for smart cards and is offered by Giesecke&Devrient as a standard product. Millions of STARCOS® cards are used for many smartcard applications. STARCOS® offers excellent security features and is compatible to ISO/IEC 7816 and the EMV 96.

19.5.2.1 The STARCOS operating system

STARCOS® (Smart Card Chip Operating System) was initially developed as a joint initiative of GMD-Forschungszentrum Informationstechnik GmbH, GAO (Gesellschaft für Automation und Organisation) and Giesecke&Devrient (G&D). Nowadays STARCOS is a

standard product of G&D and constitutes a complete multi-application operating system for smart cards.

International standards organisations and private industries have worked to define acceptable standards and conventions regarding smart cards physical and logical characteristics. G&D places great emphasis on ensuring STARCOS' compliance with both existing and forthcoming ISO/IEC and the EMV 96 standards.

Smart card operating systems provide for control of and ensure the integrity of communications between the chip and a card accepting device, access to the memory storage areas (EEPROM), and information processing. They administer the chip's resources and supply all necessary functions for operation and administration of multiple applications. Under STARCOS®, the number of loadable applications is only limited by the amount of EEPROM memory available. Additionally, the registration, creation and loading of data for an application can be done independently with defined security levels.

The main features of the current standard version STARCOS® S 2.1 include:

- The support for multiple applications in the card, which may be installed independently of each other (multi-functionality)
- The implementation of hierarchical file structures (file organisation)
- Multi-level security mechanisms during communication (Secure Messaging)
- The implementation of various access controls (authentication) via the state machine concept
- DES and triple-DES for authentication and encryption

Additionally the standard public key version STARCOS SPK 2.1 includes:

- Command, data and protocol compatibility with STARCOS S 2.1
- The generation and verification of digital signatures according to ISO/IEC 7816-8
- 1024 bit with RSA algorithm
- 512 bit with DSA algorithm
- Secure Hash Algorithm One (SHA-1)
- Internal authentication with RSA, including session key exchange

The public key version STARCOS® SPK 2.2 is an enhanced version of STARCOS SPK 2.1, which includes above all RSA key generation. Meanswhile, the next version is available.

STARCOS is used world-wide for ID, healthcare, access control and loyalty projects. Due to the flexibility of the STARCOS platform and because of its excellent security features several solutions for payment systems are based on STARCOS:

- EC-Austria with the electronic purse Quick
- Europay International: Debit/Credit
- VISA Cash
- VISA Easy Entry
- STARCOIN, the payment system of Giesecke&Devrient

STARCOS® is implemented on hardware platforms of different semiconductor suppliers and is available with 2 Kbytes, 4 Kbytes, 8 Kbytes, and 16 Kbytes EEPROM which provides the application or system provider the ability to select and deploy smart cards to a population as dictated by the application mix necessary for individuals in the cardholder population. No customisation of the card or terminal applications is required when moving from one EEPROM size to another.

Applications under STARCOS® are represented as a part of the freely definable file system in which the user data, access conditions, keys and so on are stored. Applications can be developed with G&D's Smart Card Management and Application Generator (STARMAG) design tool and use the multi-purpose command set offered by the STARCOS® operating system.

19.5.2.2 The STARMAG Toolkit

The STARMAG Toolkit (Smart Card Management and Application Generator) is a complete environment for smart card application development. STARMAG contains the following components:

- STARMAG
- STARCOS Library
- STARTEST
- PC/CTI
- STARMAG System Card
- STARCOS test cards
- G&D smart card reader

STARMAG provides a universal design and personalisation tool for smart card applications and personalisation projects on MS Windows (Win 3.x, Win 95 and Win NT) operating systems. It supports most of the current G&D STARCOS operating system versions as, e.g., the STARCOS® S 2.1, the STARCOS® SPK 2.2, or the STARCOS SV 1.0 (Visa Cash implementation of G&D).

The main features of STARMAG include:

- graphical user interface for the design of the file and security structure for smart card
- applications based on STARCOS
- initialisation of smart cards
- electrical personalisation of smart cards
- optical personalisation of cards with a thermotransfer printer
- interface for integration of personalisation data via external files (i.e. dBASE files, binary, Doc files, ...) or manual data entry
- printing of PIN mailers and generation of PINs
- batch processing for initialisation and personalisation procedure

19.5.2.3 The STARCOS Libraries

The program package STARCOS® Library is a high level driver library (CS-API, Card Service Application Programming Interface) for developing smart card applications on PC operating systems as MS DOS (from version 3.1 onwards) or MS Windows (Win 3.x, Win 95 and Win NT). The STARCOS® Library is subject to continuous further developments and will incorporate new future G&D smart card operating systems and terminals. The STARCOS® Library provides all communication components for G&D smart card operating systems and terminals to the developer and is based on our PC/CTI library. Since the STARCOS® Library automatically adapts to the different types of smart card operating systems and terminals, the developer can program applications, which in turn may integrate different smart card operating systems. For testing of smart cards the program STARTEST (included) (see Chapter 4.2.4) may be used to execute the most important functions of the STARCOS Library manually.

The STARCOS 2.1 Library supports well-known compilers as, e.g., the Microsoft Visual C++ from version 1.5 onwards. The STARCOS® 2.1 Library running under MS Windows supports all development systems that provide for incorporating DLLs. STARCOS® Library is distributed as a program package in the following variants.

19.5.2.4 The STARCOS Library BASIC

This library which is ready to be downloaded from the G&D WWW pages (see [WWWGDM]) includes all basic functions for communication with smart cards and/or card terminals without cryptographic functions as:

- simple, uniform and direct portrayal of the commands for all standard smart card operating systems of Giesecke&Devrient supported on the API level featuring high level programming
- language functionality
- encapsulation of all transmission protocols involved, other transmission interface details and the smart card hardware (buffer size, T=0, T=1)
- heavy data transmission from the smart card and vice versa using simple functions
- detection of smart card terminal and operating system currently used and hence automatic
- adaptation of all library commands to the respective differences
- automatic administration of specific library details of the smart card operating systems
- read/write and manipulate all standard and special data structures on the smart card
- authentication using PINs

19.5.2.5 The STARCOS® Library CRYPT

This library which is not prepared to be downloaded from the G&D WWW pages due to the current export restrictions includes all functions of the basic variant and additional DES and RSA functions (RSA with key lengths up to 1024 Bits) as:

- authentication using single and triple DES methods
- secure messaging with single and triple DES encipherment
- access to card encipherment functions
- single and triple DES encipherment/decipherment on the library level
- manipulation of smart card data structures secured by DES
- key generation on the library level
- hash functions (SHA-1, MD5)
- compute and verify signatures on smart card or library level
- encipherment/decipherment for RSA on the library level

19.5.2.6 The STARTEST tool

STARTEST is a universal test tool for manual test of smart cards applications on MS Windows (Win 3.x, Win 95 and Win NT) operating systems. The program provides entry dialogues for defining command parameters for all smart card commands. After command execution, the status and response data will be displayed in the main program window. All functions available are contained in the Functions menu and divided into functional groups. The most important functions can also be accessed from the tool bar.

STARTEST supports the commands at a more abstract level than the basic smart card commands. Complete authentication sequences (where several commands are processed one after the other) are supported automatically.

19.5.2.7 The PC/CTI interface

The PC/CTI (PC Card Terminal Interface) program package is a low level driver library for the development of smart card applications for many common operating systems. PC/CTI supports the G&D smart card terminals CCR2, ICT800, KCT800 and the MIFARE board.

The drivers incorporate all functions for communication between a PC and the smart card or card terminal, including complete handling of the transport protocol for the transmission of application layer data (layer 7 of the OSI reference model).

19.5.3 Security toolkit SECUDE™

SECUDE™ (Security Development Environment) is a security toolkit developed by GMD Darmstadt and now provided, updated, and distributed by the SECUDE GmbH Darmstadt.

19.5.3.1 SECUDE™ Software Development Kit

The guarantee of authenticity and the protection of the private sphere in electronic communications become of more vital concern the more electronic data processing permeates all areas of our lives. Examples of this are the protection of the private sphere in sensitive electronic mail, forgery-proof digitally signed electronic forms and contracts, the encryption of local files, network authentication, Electronic Data Interchange (EDI) and the distribution of software. The use of asymmetric and symmetric cryptography makes authenticity and confidentiality in a world-wide open electronic communications society available.

The SECUDE™ development kit is a library written in ANSI-C, offering well known and established symmetric and asymmetric cryptography for popular hardware and operating system platforms. The development kit consists of a library of functions allowing the incorporation of security efficiency in practically any application (e.g. client/server, e-mail, office applications), and a documentation in Hypertext Mark-up Language (HTML) describing in detail the C programming interface. There are also various tools to ensure an immediate deployment of security.

It offers a library of security functions and a well documented C-API which allows to incorporate security into virtually any application. In addition there is a number of ready-to-use utilities with the following features (e.g.):

- asymmetric and symmetric cryptographic functions and various hash-functions
- security functions for origin authentication, data integrity, non-repudiation of origin and data confidentiality purposes based on symmetric and asymmetric algorithms mentioned above;
- security functions on the basis of digital signatures mechanisms mentioned above;
- Diffie-Hellman key agreement;
- key certification functionalities, handling of certification paths, cross-certification, certificate revocation;
- utilities and library functions for the operation of certification authorities (CA) and interaction between certifying CAs and certified users;
- optional: secure access to public X.500 directories for the storage and retrieval of certificates, cross-certificates and revocation lists (LDAP V3).

SECUDE™ in the current version 5.2 contains the following APIs (for more information see [WWWGMD]).

- **AF** - Authentication Framework and Certification:

This module adds X.509 certification functionality to SECUDE™. Both local (i.e. PSE-located) certificates and directory-located certificates can be addressed. Therefore SECUDE™ offers an integrated X.500 Directory User Agent or alternatively an AF-Database, which is an emulation of an X.500 Directory running on a file system. Additionally ASN.1 encoding and decoding routines and a lot more auxiliary functions are available.

- **CRYPT** - cryptographic algorithms;
- **GSS** - Generic Security Services;
- **PKCS** - Public key Cryptography Standard.

SECUDE can be used as PKCS #11, SECUDE™ cannot yet deal with underlying PKCS #11.

- **PEM** - Privacy Enhanced Mail Support:

This module converts functions, which realise the Internet Specifications RFC 1421 - 1424. The basic idea of PEM is to define document-oriented message encipherment and authentication procedures for the protection of messages through the use of end-to-end cryptography between originator and recipient with no special processing requirements imposed on the message transfer system. This makes them transparent to the mail transfer systems and either applicable for local security services.

- S/**MIME** - Secure MIME:

To implement the **LDAP interface** between SECUDE and the directory a shareware version from the University of Michigan is used. The description and the software is currently available from the following location: ftp://terminator.rs.itd.umich.edu/ldap/ldap-3.3.tar.Z.

19.5.3.2 Secure Log-On with R/3 and Encrypted Communications

To authenticate the user to the R/3 server, digital signatures are exchanged. Before starting an R/3 session the user enters once only his password, thus providing **local** authentication with regard to his **Personal Security Environment** (PSE). This single log-on procedure is comparable to entering a secret number when withdrawing cash from an automatic teller.

Every time an R/3 client is started, client and server use public key procedure to exchange digitally signed acknowledgements which they verify locally. For user authentication SECUDE™ applies an asymmetric encryption procedure. During this procedure a symmetric session key is established which is used to efficiently encrypt data for the following client/server communications. For the asymmetric encryption the RSA algorithm is used; for the symmetric encryption DES, Triple DES or IDEA can be configured.

19.5.3.3 SECUDE™ authentemail

As the use of open communications networks increases, e-mail services are more and more replacing conventional telephone and fax services. Whereas till lately only printed documents could be exchanged, now the underlying files can be mailed. But e-mail services are vulnerable: On the one hand, it is easy to forge data about the sender, which makes digital signatures desirable to ensure the message origin. On the other hand, tampering with the texts themselves cannot be proved. Here, too, digital signatures help to ensure the integrity of data. Considering the current state of technology, e-mails are more comparable to postcards than to enveloped letters. Thus companies cannot dare to transmit confidential data this way. Here the use of encryption technology is required to ensure confidentiality.

E-mail Standards PEM and MailTrusT

Up to this specification, two standards for secure e-mail transmission in the Internet are defined: Privacy Enhanced Mail (PEM), and Secure MIME (S/MIME). In Germany, the MailTrusT project group of the TeleTrusT association has specified a third standard (MTT specification) that follows the Internet standards. **SECUDE authen**temail provides the user with these standards.

SECUDE authentemail is smoothly integrated into the e-mail systems Microsoft Exchange and Microsoft Outlook. The user is provided with only a few additional buttons. Thus a message can easily be **signed** and **encrypted**.

To comply with the highest security requirements, **SECUDE authen**temail – Security Grade High additionally includes a smartcard and a smartcard terminal. The smartcard is used to store the participant's digital identification.

19.5.4 Secure file transfer protocol SFTP

At the University Hospital of Magdeburg, a validated implementation using a comprehensive security enhanced file transfer protocol (SFTP) is already available that is based solely on standards (ISO, NIST FIPS-PUB, ANSI and IETF/IESG RFCs). SFTP was developed during TrustHealth 1 and has been successfully presented to the HL7 community at the HL7 Spring Working Group Meeting in Baltimore at April 1998.

The software is written in C/C++ and is based on Windows95/Windows NT. For TrustHealth 2, SFTP may be re-written using Java gaining portability for other platforms (HP-UX and other). A graphical user interface is available for presentation. Several software packages and some hardware devices are needed for realisation. Security mechanisms are supplied by the security engine **SECUDE** offered through various application programming interfaces (APIs). However, any other security engine could be used that has similar capabilities. SECUDE™ has been adjusted by the GMD Darmstadt for usage of the HPC. The smart card and the card terminal is accessed CT-API included by SECUDE™.

SFTP is based upon the TCP/IP protocol suite using the FTP client/server model as defined in RFC 0959 regarding the additional requirements. The protocol interpreter (PI) and the data transfer process (DTP) involved realise FTP processing by analysing and evaluating commands and replies (the part of the PI) as well as performing data transfer if needed (the part of the DTP). Thus, the PI is managing the control connection and the DTP is responsible for the data connection. All transfers (control and data connection) performed by the original RFC 0959-FTP protocol are insecure having no security services like strong authentication, confidentiality, integrity or accountability (in the sense of non-repudiation of origin and receipt). Only simple authentication is carried out transmitting the password in plain text. Looking at the process model of FTP, SFTP enhances security by **securing the control connection *and* the data connection** applying strong cryptographic algorithms (like hybrid encryption using 1024 bit RSA and triple-DES or IDEA session keys). The security mechanisms are based on public key cryptography establishing a public key infrastructure (PKI, trusted public keys). Furthermore, before the client could perform any command (except the command to request authentication) or data transfer on the server, a **strong mutual 3-way system authentication** is carried out. User authentication is realised using HPCs with PIN protection. Biometrical identification may be available within TrustHealth 2. The HPC is carrying all private keys for each user. In the specification context, the next figures have been discussed already in Chapter 10. To ease the reading of this chapter, the figures have been presented once more. Figure 19.2 demonstrates the schema of the strong mutual three way authentication procedure.

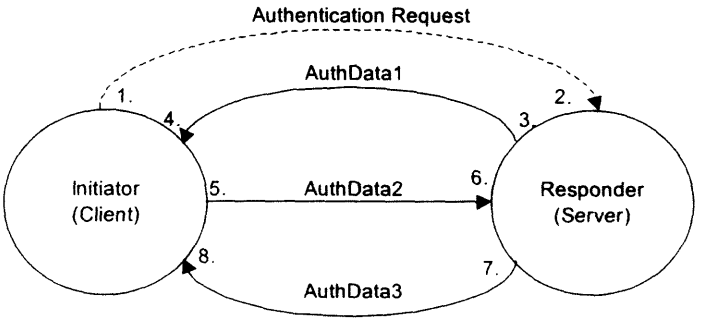
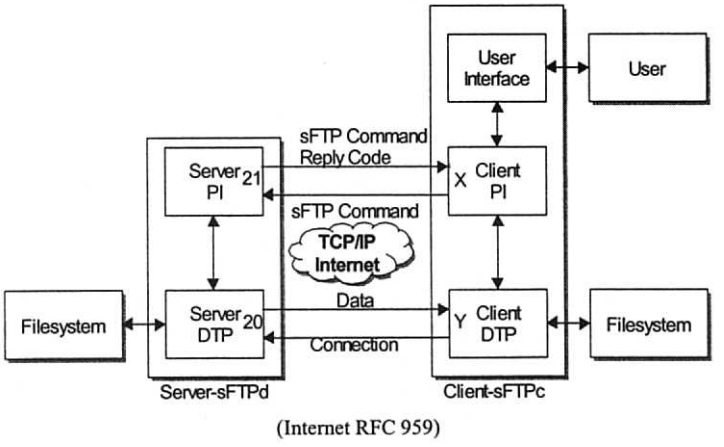


Figure 19.2: Schema of the Strong Mutual Three Way Authentication Procedure

Figure 19.3 and Figure 19.4 present the FTP control data and message data handling and the non-repudiation services respectively within the Magdeburg SFTP implementation. This solution has been developed in the context of another European project called MEDSEC (DG III).

The file transfer of HL7 messages (batch processing) is carried out by transmitting one or more messages grouped in a file and encoded according to the encoding rules of HL7. Responses are grouped and transported similarly. Proving communication security, SFTP wraps HL7 messages applying various selectable cryptographic message syntaxes as PKCS#7, security multiparts for MIME, or S/MIME. Security based on MIME takes advantage of the object-based features of MIME and allows secure messages. In general, SFTP is independent of the cryptographic syntax used, thus any other syntax can be implemented without much effort. Moreover, SFTP is able to process any desired type of file data as EDI messages (EDIFACT, HL7, X12, xDT and other) or arbitrary binary data. This openness is achieved by messages wrapping realising communication security and protocol negotiation using tag-length-value encoded data.



(Internet RFC 959)

Figure 19.3: FTP Control Data and Message Data Handling

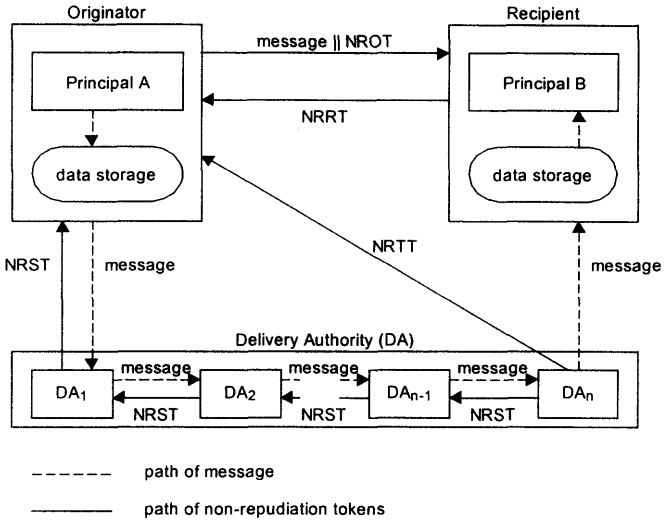


Figure 19.4: Non-repudiation Services

19.5.5 Secure file formats HL7/XML and xDT

The format of the messages exchanged between client and server should be based upon the most important EDI message syntax such as HL7 or XML. Since the HL7 Standard is moving towards XML, this new message syntax may be preferred. In Germany, xDT is very important set of specifications for the message exchange and communication between medical offices and between them and other healthcare providers including transfer of billings (ADT), medical data (BDT), laboratory requests and results (LDT), etc.

As mentioned in the section about SFTP above, this application is able to transport any kind of message data securely by wrapping the entire file. Since this transportation mechanism is independent of the message syntax delivered, the capabilities of the end-systems (client and server) exporting and importing different message syntax must be analysed.

The GTDS-connected export and import interfaces for ASCII, HL7 and BDT are still under development and will hopefully be available soon within the TH-2 time schedule. For now, XML has not been considered for the near future but is under discussion e.g. for HL7.

Regarding the client there a three different alternatives for the application environment. All are based upon the operating system WINDOWS 95/98. First of all, a Microsoft Access based application for data entry and documentation is being developed at the University Hospital of Magdeburg for the local surgery. Access offers import and export of MS Access, ASCII, MS Excel, HTML, dBASE, MS Fox Pro and ODBC data sets. Meanwhile, the University of Giessen has developed a graphical version of the terminal based client application for the GTDS. This application is based upon the ORACLE Developer tools as forms and report. At the moment there is neither an export nor an import interface available. Finally, it is always an option to develop a completely new documentation and data entry application within the TrustHealth 2 project. For interoperability and portability reasons this should be a WWW-based Java application using JDBC for database connectivity.

If some data should be entered into the ORACLE database (GTDS), the client's export interface must be able to generate HL7 or BDT messages that are in turn imported by the corresponding interface of the GTDS. And if there are queries for the GTDS, the HL7 or BDT

encoded answer has to be imported into client's application. The queries send by the client may be simple SQL statements.

Concerning the capabilities of client and server to import or export different kinds of message syntax as stated above, the interfaces needed for the client are currently missing completely and those for the server are under heavy development. Furthermore, the HL7 interface for the server seems to be available more likely than the BDT interface. So, for further working purposes, messages encoded by HL7 should be exchanged first, followed by BDT and probably XML messages at last when the interface become available.

19.5.6 Server Application GTDS

As mentioned above, the GTDS is a clinical documentation system for all professionals and organisations involved in patients' cancer care realising an early version of an electronic health record. It is based upon the popular and high performance relational database management system ORACLE. The application is developed using the well-known ORACLE development tools as SQL*Plus, SQL*Forms, SQL*Report etc. The patient- as well as case-related application facilitates the record, the storage and the processing of the essential cancer data (basic documentation, extended basic documentation, specialised or organ-specific documentation) beginning with the cancer diagnosis and finishing with the conclusion of the case. The application is organised in accordance with the general documentation structure as diagnosis, treatment, consile, aftercare, conclusion. The functional and the data access rights are professional-related, considering organisational relationships and care-related roles. The application realises both the mandatory and the discretionary access model. In the future, the access control management will be further improved. Enhanced data security measures are introduced.

19.5.7 Client Applications GTDS

The current PC client emulating the terminal mode provides the user interface to the server GTDS. Local security measures are used. Currently, enhanced clients and in the future independent doctor's working places are under development to

- extend and to improve the functionality and therefore the usability and acceptability of the client, and
- reduce the traffic between client and server.

Depending on the availability of security and client application functionalities, different service levels will be provided, beginning with simple software PSE or HPC-based authentication, confidential mail services and secure FTP.

19.5.8 The hardware components

19.5.8.1 Server

The server is a HP9000 / G30 machine running HPUX v9 (further updates to 10.x are planed). The hard disk capacity is about 9.3 GB consisting of three volumes, the server runs with 256 MB RAM. For archiving procedures a special DAT drive is used. A battery based power support system ensures a 24 hours availability. For external communication processes an interface multiplexing system with 8 serial ports has been installed, actually one port is connected to a modem with 14.4 Kbit/s. For this line the MACS (Modem Access Control System) is used.

For communication purposes with other hospitals and clinics an ISDN based system is installed. The server is connected to an ISDN-Router (manufacturer AVM) with a so-called LAN (Local Area Network) box ensuring line encryption and device authentication. An-

other network connection is used only for internal purposes, for connecting terminals as well as other servers of the campus network.

For test purposes another server is available. It is an HP9000 / 730 HPUX v10.2 with a hard disk capacity of about 3.4 GB and a 64 MB RAM, DAT drive, CD drive, and internal network connection.

For TTP functions, a new HP system with updated facilities was introduced. Close to the GTDS server, it will host the local directory and other functions related to TTP services.

19.5.8.2 Clients

The client is a WINDOWS95 based PC currently running a terminal emulation software or a first version of a local database system including ftp services. The PC has a Pentium 120 (or higher speed) processor with a hard disk capacity of about 2 GB and 32 MB RAM as well as a network connection. The system is connected to the University Hospital LAN.

19.5.8.3 Card Terminal

The chipcard terminal which is used within the pilot is called ICT 800. It is manufactured by G&D and follows the Multifunctional Card Terminal-Specification [MCT]. In addition to a normal chipcard terminal which is used for ID cards, also two plug-in cards can be inserted at the bottom side of the terminal. The terminal is equipped with a keypad according to ISO/IEC 9564, and an LC display.

This card terminal type has a feature which allows the users to update the internal card terminal software. Therefore, a simple terminal software program (e.g. Telnet, Telix etc.) is requested to upload the new version. So within the process of TH-2 it is possible to use the same hardware with new software versions in order to avoid the waste of budgets for devices.

19.5.8.4 Cards

The smartcard STARCOS® SPK 2.2 is manufactured by Giesecke&Devrient Munich [WWWG&D]. The STARCOS® SPK version which will be used within the pilot operation has the characteristics which are needed for signing documents according to German Digital Signature Law. Besides keys (RSA 1024) are generated in the chip, so that the whole life cycle of the private key is on the card alone.

19.6 Example for PKCS#7-Based Security

For application of PKCS#7-only security, the plain data file is available on the file system. Next, this file is signed applying the `signedData` object of PKCS#7 (with the `contentInfo` field carrying the message data). At last, this oject is encrypted using the `envelopedData` object of PKCS#7. After transportation, this file is processed conversely (decryption following verification).

19.6.1 Example for Security Multiparts for MIME

In this subsection an example is presented, at which an HL7 message is secured by hybrid encryption using Security Multiparts for MIME as specified in [RFC1847] and PKCS#7 for signing and encryption. For hybrid data encryption, a nesting of content-types is performed as explained in [MIME-SECURE] and [SMIME2] using a triple DES session key (112 bits significant, DES3-EDE2-CBC).

First of all, the HL7 sample message as shown in Figure 19.5 is available on the file system in plain text.

```
MSH|^~\&|DPS||CLOVERLEAF||19970922075909||ADT^A08|0165648|P|2.2|||AL|NE
EVN|A08|19970922075857
PID||123456|SSW23084913|97045331|Sorglos^Susi||19490823|W|||Milchstrasse
99^^Magdeburg^^39999^D|15303000|6211123||deutsch||0|||||||||D
NK1|1|Sorglos^Harry|EHEMANN|Milchstrasse 99^^Magdeburg^^39999^D|6211123
PV1||S|MKG01^^^MKG|R|||||||||||||S|97999999|||||||||||||||||||95901||||
|19970917084100
DG1|1|ICD9|2398||19970917085134|AUF
IN1|1|001441346|0969999|HaMü Lg Ost/Gst Magdeburg|Keplerstraße
6^^Magdeburg^^39104^D|||1|HAM||||1200||M|Sorglos^Susi||19490823|Milchstra
sse^^Magdeburg^^39999^D|||||||||||||||0969999^99604^1200^1000^9
```

Figure 19.5: HL7 Sample Message

This message is Base64-encoded and inserted into a MIME entity using the content-type "`application/x-EDI-HL7`" as proposed in [HL7SEC] (regarding [RFC1767] and [MIME-SECURE]). For readability of the HL7 messages, the quoted-printable encoding could be implemented. Next, this entity is canonicalised (that means 7-bit ASCII representation with lines terminated by carriage return <CR> and line feed <LF> as specified in [RFC1848] Chapter 2.1.1.) as presented in Figure 19.6.

```
Content-Type: application/x-EDI-HL7; charset=us-ascii<CR><LF>
Content-Transfer-Encoding: base64<CR><LF>
Content-Description: HL7 V2.2 message<CR><LF>
<CR><LF>
TVNIfF5+XCZ8RFBTfHxDTE9WRVJMRUFGfHwxOTk3MDkyMjA3NTkwOXx8QURUXkEw<CR><LF>
OHwwMTY1NjQ4fFB8Mi4yfHx8QUx8TkUKRVZOfEEwOHwxOTk3MDkyMjA3NTg1NwpQ<CR><LF>
SUR8fDEyMzQ1NnxTU1cyMzA4NDkxM3w5NzA0NTMzMXxTb3JnbG9zXlN1c2l8fDE5<CR><LF>
NDkwODIzfFd8fHxNaWxjaHN0cmFzc2UgOTleXk1hZ2RlYnVyZ15eMzk5OTleRHwx<CR><LF>
NTMwMzAwMHw2MjExMTIzfHxkZXV0c2NofHwwfHx8fHx8fHx8RApOSzF8MXxTb3Jn<CR><LF>
bG9zXkhhcnJ5fEVIRU1BTk58TWlsY2hzdHJhc3NlIDk5Xl5NYWdkZWJ1cmdeXjM5<CR><LF>
OTk5XkR8NjIxMTEyMwpQVjF8fFN8TUtHMDFeXl5NS0d8Unx8fHx8fHx8fHx8<CR><LF>
U3w5Nzk5OTk5OXx8fHx8fHx8fHx8fHx8fHx8fHx8OTU5MDF8fHx8fDE5OTcwOTE3<CR><LF>
MDg0MTAwCkRHMXwxfElDRDl8MjM5OHx8MTk5NzA5MTcwODUxMzR8QVVGCklOMXwx<CR><LF>
fDAwMTQ0MTM0NnwwOTY5OTk5fEhhTfwgTGcgT3N0L0dzdCBNYWdkZWJ1cmd8S2Vw<CR><LF>
bGVyc3RyYd9lIDZeXk1hZ2RlYnVyZ15eMzkxMDReRHx8fDF8SEFNfHx8fDEyMDB8<CR><LF>
fE18U29yZ2xvc15TdXNpfHwxOTQ5MDgyM3xNaWxjaHN0cmFzc2VeXk1hZ2RlYnVy<CR><LF>
Z15eMzk5OTleRHx8fHx8fHx8fHx8fHwwOTY5OTk5Xjk5NjA0XjEyMDBeMTAw<CR><LF>
MF45Cg==<CR><LF>
```

Figure 19.6: MIME Entity of the HL7 Sample Message

Then, the MIME entity given in Figure 19.6 is signed using PKCS#7. Following [RFC1847], the MIME entity is inserted into a `multipart/signed` MIME message as the first body part depicted in Figure 19.7. The digital signature (PKCS#7-`signedData` object with an empty `contentInfo` field as explained in [SMIME2]) is Base64-encoded and inserted into the second body part. According to [RFC1847], the signature covers the MIME header of the first body and the first body itself as marked bold in Figure 19.7.

```
Content-Type: multipart/signed;<CR><LF>
   protocol="application/pkcs7-signature";<CR><LF>
   micalg=md5; boundary=sig_bound<CR><LF>
<CR><LF>
--sig_bound<CR><LF>
Content-Type: application/x-EDI-HL7; charset=us-ascii<CR><LF>
```

```
Content-Transfer-Encoding: base64<CR><LF>
Content-Description: HL7 V2.2 message<CR><LF>
<CR><LF>
TVNIfF5+XCZ8RFBTfHxDTE9WRVJMRUFGfHwxOTk3MDkyMjA3NTkwOXx8QURUXkEw<CR><LF>
OHwwMTY1NjQ4fFB8Mi4yfHx8QUx8TkUKRVZOfEEwOHwxOTk3MDkyMjA3NTg1NwpQ<CR><LF>
SUR8fDEyMzQlNnxTU1cyMzA4NDkxM3w5NzA0NTMzMXxTb3JnbG9zXlN1c218fDE5<CR><LF>
NDkwODIzfFd8fHxNaWxjaHN0cmFzc2UgOTleXk1hZ2RlYnVyZ15eMzk5OTleRHwx<CR><LF>
NTMwMzAwMHw2MjExMTIzfHxkZXV0c2NofHwwfHx8fHx8fHx8RApOSzF8MXxTb3Jn<CR><LF>
bG9zXkhhcnJ5fEVIRU1BTk58TWlsY2hzdHJhc3N1IDk5X15NYWdkZWJ1cmdeXjM5<CR><LF>
OTk5XkR8NjIxMTEyMwpQVjF8fFN8TUtHMDFeX15NS0d8Unx8fHx8fHx8fHx8fHx8<CR><LF>
U3w5Nzk5OTk5OXx8fHx8fHx8fHx8fHx8fHx8fHx8OTU5MDF8fHx8fDE5OTcwOTE3<CR><LF>
MDg0MTAwCkRHMXwxfElDRD18MjM5OHx8MTk5NzA5MTcwODUxMzR8QVVGCklOMXwx<CR><LF>
fDAwMTQ0MTM0NnwwOTY5OTk5fEhhTfwgTGcgT3N0L0dzdCBNYWdkZWJ1cmd8S2Vw<CR><LF>
bGVyc3RyYd9lIDZeXk1hZ2RlYnVyZ15eMzkxMDReRHx8fDF8SEFNfHx8fDEyMDB8<CR><LF>
fE18U29yZ2xvc15TdXNpfHwxOTQ5MDgyM3xNaWxjaHN0cmFzc2VeXk1hZ2RlYnVy<CR><LF>
Z15eMzk5OTleRHx8fHx8fHx8fHx8fHx8fHwwOTY5OTk5Xjk5NjA0XjEyMDBeMTAw<CR><LF>
MF45Cg==<CR><LF>
<CR><LF>
--sig_bound<CR><LF>
Content-Type: application/pkcs7-signature; name=smime.p7s<CR><LF>
Content-Transfer-Encoding: base64<CR><LF>
Content-Disposition: attachment; filename=smime.p7s<CR><LF>
Content-Description: secure MIME (RFC1847, RFC2311) cryptographic signa-
ture<CR><LF>
<CR><LF>
MIAGCSqGSIb3DQEHAqCAMIACAQExDjAMBggqhkiG9w0CBQUAMIAGCSqGSIb3DQEH<CR><LF>
AQAAMYGHMIGEAgEBMCAwGzELMAkGA1UEBhMCREUxDDAKBgNVBAoTA2dtZAIBBDAM<CR><LF>
BggqhkiG9w0CBQUAMA0GCSqGSIb3DQEBAQUABEAUDT2sCVZ0AHnieifA3J7cdW/u<CR><LF>
KwR7HbUeQKdb8QQXf1l0VBMUYPR4eKN0vmQqImpfqWhSkdBpT4eLdCFrZyfvAAAA<CR><LF>
AAAA<CR><LF>
<CR><LF>
--sig_bound--<CR><LF>
```

Figure 19.7: Signed HL7 Sample Message Using Secure MIME Multiparts

Next, the whole `multipart/signed` message (including MIME headers and trailers) as presented in Figure 19.7 is encrypted using the PKCS#7-`envelopedData` object. The result is shown in Figure 19.8. According to [RFC1847], the first body parts contains control information (control value and protocol parameter) to decrypt the data in the second body part. Analogously, after transportation the HL7 message is decrypted and verified. For validation of the digital signature, the part covered by the signature must be canonicalised first.

```
Content-Type: multipart/encrypted; <CR><LF>
   protocol="application/pkcs7-mime";<CR><LF>
   boundary=enc_bound<CR><LF>
<CR><LF>
--enc_bound<CR><LF>
Content-Type: application/pkcs7-mime; charset=us-ascii<CR><LF>
Content-Transfer-Encoding: 7bit<CR><LF>
Content-Description: control information for decryption<CR><LF>
<CR><LF>
Version: 1.5<CR><LF>
<CR><LF>
--enc_bound<CR><LF>
Content-Type: application/octet-stream; name=smime.p7m<CR><LF>
Content-Transfer-Encoding: base64<CR><LF>
Content-Disposition: attachment; filename=smime.p7m<CR><LF>
Content-Description: secure MIME (RFC1847) cryptographic message<CR><LF>
<CR><LF>
```

```
MIAGCSqGSIb3DQEHA6CAMIACAQAxeDB2AgEAMCAwGzELMAkGA1UEBhMCREUxDDAK<CR><LF>
BgNVBAoTA2dtZAIBAzANBgkqhkiG9w0BAQEFAARAgDEpj+/hxNHduq4iFd3M0pcn<CR><LF>
0ywweRBx14Crr0Zt+qeVbMNYbo2r9LXLc0l4B9vYuVFoQTKG9AwnUhZOOsITATCA<CR><LF>
BgkqhkiG9w0BBwEwCQYFKyQDAQ0EAKCABIIGcFDv/uBgSkKbz9fg+E1X44ZdI237<CR><LF>
AyB5uPCXDyc9V35KMbZ/C99f7QU0rNUn6jxLW88/hl9dRIMsdAy4XcWGhWqywfPn<CR><LF>
NfaQwMn4whalRvtlxEvU97Ti35ksPCicZYNdDYT79K/ZFvw5s2+vjP02JiAXj3oA<CR><LF>
ms2YPG8VjkDXCyuPTLd7gd7y5ztkAVzCC4sYHL+tf9s9t7iajrQZuFBFpgqHtFFh<CR><LF>
jREiXnVRvnymeEQidVCmCshufJ7c4+BPykwPM7Qlo7h5QM/TeuY6ZsgmuAXCS/zA<CR><LF>
FMHlRrRHdyIY/rM9wGUQR/VvHs2RtOg8zMQ2e89APOc+NC7dilgQRY0+F4GojtYo<CR><LF>
TLs7cJ1HZmWtONdQG0rWi8h4nLrcdWAkzHAaxOOI0YhNdNLGyfbfZDHqBfrrek9K<CR><LF>
G1xDh1I5gNBjJ8rSlSAgzRAZZa+vFSNtE4WEdsLroMUOPCzmIpb7PJoeix+tL+E1<CR><LF>
A7jTZ93oOMGhSoitDYH76kb/40XqluiVg19Fx9+yXvNBAUHBYBe6B+l9fzr3TUro<CR><LF>
1u6Hutlt2iHohs0Pn3wdToJfcquRpR7MWrfmijfUbfSbsVdc6FAcX4u6gPUirLBl<CR><LF>
gV4enklPy/7qucR8jA85JDmHDPfRv+lM3qIqoH9ckfHqwGg/pT0rU4vYj2hcwTPL<CR><LF>
UY8HZc3bOuMreDJ3FC5Au3qKqtMFCK+5Xmh6TLY5jU/qtnHStYBgNT/6fFwte/nL<CR><LF>
fyjYG/qc9UDAX/MlJKEHNORp0Gx8KOP0aw6YpwvJN0ug8+xE2XMPoHKoe9ea6QZu<CR><LF>
ZxrzlQi4KZd1Z1azR05X5nqsCFuzFDr1Ez+LKbEpw+oqS70lNRjVvzJcAY9G3iuy<CR><LF>
VGNrRLCptJJKiFwd2DxLYXJQBlh5KJojNFyKA8Rggz7yh8/bE3x21ZvMaAFO/lk/<CR><LF>
Ky1ai8kUw9LULvSMnNgA32ECRX4EFuK3IO1V5l0hjW6WA9scl6Q51qn2d/gzEWmq<CR><LF>
+oBjQp4kL80XLBgOgcvm4/jflWHhryDENLdXMhRctGuYaR/1YP58FqHeNCayZf4z<CR><LF>
cE4GudAc6i8A186oLtZDNUlNbHiROZ+wIkbMTlAymRV1QJt/BjlCPBwdv1XEDcQ2<CR><LF>
IpLqb9hUXdbnrSHuOpizvNNj2DK+7CS5F/fKIdlDKtM4W8nFnQnzWujtsGWmvlJo<CR><LF>
ILaYpxVHkNFuZtlQpeY+w3bqMKEm5Wk0PeLIPAG6YtcxmAyeIPW65aJyhQBefm2a<CR><LF>
hhgOPsjCpnRHHdyZdXxObDCDg2qV2sqoEP3eTOXIbPRV9Ozer6uTiN6+ZgNRTkAU<CR><LF>
Ui2bpb0nVRMpelQqldIx0q5yO89FLG0uxywa4A3VypPBQWlhxwK7rb4DBKNl0PdG<CR><LF>
C7a0uuQNPmjOOC2hg/dbB86iK+N7dkf0uhr3BiCX8fDhRFGzyhJDeAcORrnh0Aoy<CR><LF>
mjaaOZFKnacZud0bmoSW9kUsPv/NlB4UTQOsPSFBV+dOyJuBlY38IAuqtBAy/mmd<CR><LF>
xDSk0hHTVpk8PGRS0f2P2PrX7nu4nfWuWUCsr/KCuYRqNdfwvVN5BqciazXE2w+R<CR><LF>
0jS4PMfVaNIyT2hW58G9T/A/iW+KRYDh3wrvVZZs5DV+P7WFsTzcXyv8lP+wfevy<CR><LF>
dAMpjOoTtFmYN6mbpbX9yL/wGRM0lcNuZQfuyQ5xxjSfraYHKaJ7zBx4YfQEbv2L<CR><LF>
6KB87gxqBSerjeQSuxHu6TJzEyGolBEnj4onujsYH2WcKyAEn4j6TgbeJQtDo+Eq<CR><LF>
j1c1nI07YK982ZS9nIh1/sDxOBCK6PhF5zptP/emm+2XOXIdKW6jIpWvat94XV5C<CR><LF>
3DZqPftSCrqqUFaaTfl0qxL3fFPC2bqntB0WDaquachZzF0DaWy4APOB/SAfp+Wt<CR><LF>
L4yoUYt2g7rphHrfj4Rn9rqSxzVYnayq/6+H2SHYWMazsJ48NKZ9ct8NmuehwkUR<CR><LF>
8AOcZYuinlLzoJrcpnLLb7GCwTL8MncNhilvruujvFsr1IMsfwn5D7VGTyVnvYPG<CR><LF>
vD+8aOWTSRV3Kt/hauvNG4/kPbDTD6i/hoxuA77pOWSkiqStOzvt778Pkd6Biril<CR><LF>
GM+KgEVCC20jN+qLcB1y4ELEQjIF0Pmx8p/vjoIW6ojDwltRpzkDywYNZTA58VvY<CR><LF>
N82PYa1RRMieax7OzV3wYb/OysZxyGkcweIV72q354bk3cQ3/MyH/jsXDoEECEDp<CR><LF>
vu56B2RSAAAAAAAAAAAAAA==<CR><LF>
<CR><LF>
--enc_bound--<CR><LF>
```

Figure 19.8: Encrypted Message Using Nesting of Secure MIME Multiparts

For signed-only transportation, the process ends up in the `multipart/signed`-structure as presented in Figure 19.7. In the case of encrypted-only delivery, the MIME entity in Figure 19.6 is converted to the `multipart/encrypted`-structure directly without any intermediate step.

19.7 Example for S/MIME Version 2

In this subsection an example is presented, at which an HL7 message is secured by hybrid encryption (applying DES3-EDE2-CBC) using S/MIME version 2 as specified in [SMIME2] and PKCS#7 for signing and encryption. First of all, the HL7 sample message as shown in Figure 19.8 is available on the file system in plain text.
This message is Base64-encoded and inserted into a MIME entity using the content-type "`application/x-EDI-HL7`" that is canonicalised afterwards as presented in Figure 19.8. Next, this entity is signed using the PKCS#7-`signedData` object (with the `contentInfo` field carrying the MIME entity) as explained in [SMIME2]. Alternatively, `multipart/signed` can be used for signing. At last, the PKCS#7 object is inserted into an `application/pkcs7-mime` MIME entity as shown in Figure 19.9.

```
<CR><LF>
     name=smime.p7m<CR><LF>
Content-Transfer-Encoding: base64<CR><LF>
Content-Disposition: attachment; filename=smime.p7m<CR><LF>
Content-Description: s/mime v2 (RFC2311) signed data<CR><LF>
<CR><LF>
MIAGCSqGSIb3DQEHAqCAMIACAQExDjAMBggqhkiG9w0CBQUAMIAGCSqGSIb3DQEH<CR><LF>
AaCAJIAEgYNDb250ZW50LVR5cGU6IGFwcGxpY2F0aW9uL3gtRURJLUhMNzsgY2hh<CR><LF>
cnNldD11cy1hc2NpaQ0KQ29udGVudC1UcmFuc2Zlci1FbmNvZGluZzogYmFzZTY0<CR><LF>
DQpDb250ZW50LURlc2NyaXB0aW9uOiBITDcgVjIuMiBtZXNzYWdlDQoNCgRAVFZO<CR><LF>
SWZGNStYQ1o4UkZCVGZIeERURTlXUlZKTVJVRkdmSHd4T1RrM01Ea3lNakEzTlRr<CR><LF>
d09YeDhRVVJVWGtFdwQCDQoEQE9Id3dNVFkxTmpRNGZGQjhNaTR5Zkh4OFFVeDhU<CR><LF>
a1VLUlZaT2ZFRXdPSHd4T1RrM01Ea3lNakEzTlRnMU53cFEEAg0KBEBTVVI4ZkRF<CR><LF>
eU16UTFObnhUVTFjeU16QTRORGt4TTN3NU56QTBOVE16TVh4VGIzSm5iRzl6WGxO<CR><LF>
MWMybDhmREU1BAINCgRATkRrd09ESXpmRmQ4Zkh4TmFXeGphSE4wY21GemMyVWdP<CR><LF>
VGxlWGsxaFoyUmxZblZ5WjE1ZU16azVPVGxlUkh3eAQCDQoEQE5UTXdNekF3TUh3<CR><LF>
Mk1qRXhNVEl6Zkh4a1pYVjBjMk5vZkh3d2ZIeDhmSHg4Zkh4OFJBcE9TekY4TVh4<CR><LF>
VGIzSm4EAg0KBEBiRzl6WGtoaGNuSjVmRVZJUlUxQlRrNThUV2xzWTJoemRISmhj<CR><LF>
M05sSURrNVhsNU5ZV2RrWldKMWNtZGVYak01BAINCgRAT1RrNVhrUjhOakl4TVRF<CR><LF>
eU13cFFWakY4ZkZOOFRVdEhNREZlWGw1TlMwZDhVbng4Zkh4OGZIeDhmSHg4Zkh4<CR><LF>
OAQCDQoEQFUzdzVOems1T1RrNU9YeDhmSHg4Zkh4OGZIeDhmSHg4Zkh4OGZIeDhP<CR><LF>
VFU1TURGOGZIeDhmREU1T1Rjd09URTMEAg0KBEBNRGcwTVRBd0NrUkhNWHd4ZkVs<CR><LF>
RFJEbDhNak01T0h4OE1UazVOekE1TVRjd09EVXhNelI4UVZWR0NrbE9NWHd4BAIN<CR><LF>
CgRAZkRBd01UUTBNVE0wTm53d09UWTVPVGs1ZkVoaFRmd2dUR2NnVDNOMEwwZHpk<CR><LF>
Q0JOWVdka1pXSjFjbWQ4UzJWdwQCDQoEQGJHVnljM1J5WWQ5bElEWmVYazFoWjJS<CR><LF>
bFluVnlaMTVlTXpreE1EUmVSSHg4ZkRGOFNFRk5mSHg4ZkRFeU1EQjgEAg0KBEBm<CR><LF>
RTE4VTI5eVoyeHZjMTVUZFhOcGZId3hPVFE1TURneU0zeE5hV3hqYUhOMGNtRnpj<CR><LF>
MlZlWGsxaFoyUmxZblZ5BAINCgRAWjE1ZU16azVPVGxlUkh4OGZIeDhmSHg4Zkh4<CR><LF>
OGZIeDhmSHd3T1RZNU9UazVYams1TmpBMFhqRXlNREJlTVRBdwQCDQoECE1GNDVD<CR><LF>
Zz09BAINCgAAAAAAADGBhzCBhAIBATAgMBsxCzAJBgNVBAYTAkRFMQwwCgYDVQQK<CR><LF>
EwNnbWQCAQQwDAYIKoZIhvcNAgUFADANBgkqhkiG9w0BAQEFAARAFA09rAlWdAB5<CR><LF>
4nonwNye3HVv7isEex21HkCnW/EEF39ZdFQTFGD0eHijdL5kKiJqX6loUpHQaU+H<CR><LF>
i3Qha2cn7wAAAAAAAA==<CR><LF>
```

Figure 19.9: Signed HL7 Sample Message Using S/MIME Version 2

For encryption, the signed message given in Figure 19.9 is enveloped using the PKCS#7-envelopedData object and inserted into an application/pkcs7-mime entity as described in [SMIME2] and shown in Figure 19.10.

Analogously, after transportation the HL7 message is decrypted and verified. For validation of the digital signature, the part covered by the signature must be canonicalised first.

```
Content-Type: application/pkcs7-mime; smime-type=enveloped-data; <CR><LF>
     name=smime.p7m<CR><LF>
Content-Transfer-Encoding: base64<CR><LF>
Content-Disposition: attachment; filename=smime.p7m<CR><LF>
Content-Description: s/mime v2 (RFC2311) enveloped data<CR><LF>
<CR><LF>
MIAGCSqGSIb3DQEHA6CAMIACAQAxeDB2AgEAMCAwGzELMAkGA1UEBhMCREUxDDAK<CR><LF>
BgNVBAoTA2dtZAIBAzANBgkqhkiG9w0BAQEFAARAcj8ScrceifR6f6bWKikPdply<CR><LF>
KhuNHXu20zPOBlJmh1NT4eUu4it5lMzw0TLCkQXeMN/sb5eB7cMHJ77Zlm2hbDCA<CR><LF>
BgkqhkiG9w0BBwEwCQYFKyQDAQ0EAKCABIIHqFGAzQzunoFgPsRBR3YL/oKJjBq6<CR><LF>
41e9JQfEuBIUfSylKAojLfFUeB76sIGkEwe/Ckc10huEo7LybX//D6SILPjHiOmL<CR><LF>
l1P+mydMvDMeiUVB36D3c4V8vh1isqThumYVT+jbHcRMwjLwul5Ue6Dj05sjUNcz<CR><LF>
n+AOazTOTkC2KumW1bvm9J3FYCJrFBwn2lSZa/OYSEkRKUR2vIVaaufDdZUQ2s9n<CR><LF>
1qHN/nblNoRNmUz2v2KJy2gSC6JhXfzmkXD2OgiRtzKEuNmLhk0JGN+1Xn0aQlk9<CR><LF>
vlzIVw9ycMfN3zhctwA8P5OMFleQXqQvLMkiFrp+QqeT38+zitpHqv2x1jdkNdd/<CR><LF>
JzTIcwRQGSOz7FuCLRC1L5CTqGGfyFyv6IB5m8/uwxAidVlx1G/wB2vXgmfK80qU<CR><LF>
pjVG3IoE5bJZDAynAWbPWnwUE0FdM3YibfKVrC7SNW1FNZn7cAcvJ5er9YUyxOEO<CR><LF>
9q8aC7D4u727shuYqVzDLbrqKVQjUQwU9xNU/1P+CR0MWT+2RV8R8l/QZzYc8+NZ<CR><LF>
AqItSmbyGeN1zqjoo+mBDD79UDKhprisTduI7VqqT9N5tAz2b8ZD1gS3nq8V0dws<CR><LF>
meFmyChKUIL8eaZe+VAJ+ofNJlPGBbPxUa0SbnG9ELF/h12A1YXrKw+WzAGhlFAZ<CR><LF>
myfQtSr3MbvV2IDQNZ7pVfpSz1Ul5gY+CybEAyWUh7+Bg/Ob/fN0wsv5KniE1gHW<CR><LF>
upAsRiz2++LIp3wnniFg8jKpz1Dh4x03p4FDO+1s3qGH8ESvmutTCVzXOj439m/h<CR><LF>
7D9iLtxSMJZu06ARNvGM2Cz3ISvSNV7oJekYgEQ/L6E1pXBsXvCuULfmb6pYedkU<CR><LF>
s3BMwWoMPzd6h+kmQRHlykSJgXx8PwJ4ilEf2uRCo7M2CmG1uvzXYlT+YFwWBgP9<CR><LF>
KZaYbHnX36ATTRpjdtwe0LrvMuZ3h1P6f6DtHPKjaV6UG8FW/q702x3E5Wjq036x<CR><LF>
7iBP95A8tnlMsmvoHfz6esgrd4RRYvN00i6vnCALYkdhjSqledKiYvatxfMylLnF<CR><LF>
WQf8sa41Xb0jeQWsc2iZz3yM0jg2mnuuYpafrstKARvQAQrKcgIh8Qb3AxUgnB9g<CR><LF>
YM/jks+iMgNLFvwU04JKXchKmc9KvwHclqc5FUfAUGqw9OP3x1JWESq6BsjGFKTJ<CR><LF>
XBCP1/MktKPUdyLOXaggNgP8YMJPkHkbc55WQGelshf+by75PJmjdTipVL+0OAlO<CR><LF>
lXAxMMu7P2f/5+eEA4Yv5Dkd3v6v6IaTudKo9m+ZnlUteS4H+Jq9Fq+U6evIpjxY<CR><LF>
q5Zzz/6TM9h7SXnkWJK4WVjJpnJkqnmPFeA0r2yBfmIjhKullXwbGFcx+Mi9tutM<CR><LF>
Di9uze41qzxFxth8EDFxcF9h0Z0ept66lReN2zC5WNG7bkrnWfGn3gDv2FtABzw9<CR><LF>
rNTIUD8YlkAHaIM22ms106LYBSSxTvejFkraxb0a/Cm9Sj1g7Hh/um9rfQfNY+IR<CR><LF>
Q9KTGedE+hC7VcGlegndY3XwTe0GFoIPDVfSzSDFtlI2H6XcRv3i6C+9OhEsxIFk<CR><LF>
hqz30WzdrZUYemnKAPUwmWK8DTwJF6XkP2+lRVXtkvjOt/hutFVL6wbU9C5Mulu5<CR><LF>
FflqEOE1BYQF+6XFcMfHOKso3fzh47F/bVi/aYQm/uS0LZlaf8UxACOEO5oZYr2w<CR><LF>
2JwGlJi6vI4MGkivw6Riih3Ar4osFJEwllxjFivXISwAvQL2D4eY+yx29WtxXNpC<CR><LF>
SoZtfhNObYCPcGphIBh9DQFMm0+1CKKr50Ndwl4Rc7Zx77DLMc3dLeVQUni6u7gN<CR><LF>
yAlseIL4Nj5fPdSTFbOEHbyiZX3dMaEh32FHMtFbYg9qzgw4yfZI+0G5m4QOM/BM<CR><LF>
i4SnGwik/LPU6stCHxPfRRY42MkS2fTnNYkbBFD+qL8GKbFGTH8+CiuBsGSSsMt4<CR><LF>
eqJyC3La4FfcjZCJKVe3dCAZS7kEJ+XaJgc0uUrjmsO7otP0lGchmWfQY/XLpo0K<CR><LF>
XhRmZR3j+8pFyJqKkSMODY9W553WM6PLUqSrtmHV+ti/dM/7aXxdGSRbFxS6NruQ<CR><LF>
3wP/l3cOi/I8M0Mw02ZFMruX/Y+RRFKqsk1FESnGE56V6BfACk/hqc04+cHKLWvk<CR><LF>
```

```
gDEhgFqwqhPVkz+8wF9VlEWJBMBepO3OQM9Q+xFQPIYQSnkX4H2a/7fWb4ayp/Pa<CR><LF>
4IA9vcW5GyrV58x/zPyvGReIWf/bdS366ZOOd8Lkvs+2kWF3J0XK6/YMYmAMNGyS<CR><LF>
o9/J4KHbEHxCseHZVmaJxD6IX43tGvifLq7TCQLfV2lDhsS9KxREdRz9ojW3wqum<CR><LF>
cto/+lsW/hjW9cjyL3zjSbu0Mssvy17oG58uUCJvUwB1iSqD/k3IYYPG909KoG2m<CR><LF>
OYmzSbKpBeoH8s6BW5JpvYFCa0ymbBRJ6KZ98jPTCs2lpQSLfIq/IxizBxutT2KS<CR><LF>
Rxq33Na/SD3wUdzhO2Ajy+PklKvUOTWGf3K6gRC8ZWj0QVNtd+xqUOIFojZv4u/f<CR><LF>
UJ+cmdgdsNPLV25TRB0cE29a7geLg0pYZ+5cG6z0ZEodRgeIWLM7dvFHyrPctJSV<CR><LF>
Pse8YCX8+ZbCyGyH6yhnxc3bn0EECHPi1OuZCZjKAAAAAAAAAAAAAAA==<CR><LF>
```

Figure 19.10:Encrypted HL7 Message Using S/MIME Version 2

For signed-only transportation, the process ends up in the `application/pkcs7-mime`-structure as presented in the "Standard Guide for Implementing Secure EDI (HL7) Communication Security", A-4. In the case of encrypted-only delivery, the MIME entity in Figure 19.6 of the same document is converted to the structure in figure A-6 directly without any intermediate step.

19.8 References

[WWWG&D] WorldWideWeb pages of the Giesecke & Devrient company Munich, in particular the STARCOS-related pages under http://www.gdm.de/starcos

[WWWGMD] WorldWideWeb pages of the GMD Darmstadt, in particular the TKT-related pages under http://www.gmd.darmstadt.de/TKT

[BB&PP97] B.Blobel, P.Pharow: Security Infrastructure of an Oncological Network Using Health Professional Cards. In: L. van den Broek, A.J.Sikkel (Edrs.): Health Cards '97, pp 323-334. Series in Health Technology and Informatics Vol. 49. IOS Press, Amsterdam 1997.

[BB&PP98] B.Blobel, P.Pharow: Results of European Projects Improving Security of Distributed Health Information Systems. In: B.Cesnik, A.T.McCray, J.-R.Scherrer (Edrs.): MEDINFO '98, pp. 1119-1123. IOS Press Amsterdam, Berlin, Oxford, Tokyo, Washington DC 1998.

[BBPPVS] B.Blobel, P.Pharow, V.Spiegel: Shared Care Information Systems Based on Secure EDI. In: P.W.Moorman, J.van der Lei, M.A.Musen (Edrs.): EPRiMP – The International Working Conference on Electronic Patient Records in Medical Practice, pp. 164-171. IMIA Working Group 17, Rotterdam 1998.

[MCT] GMD Darmstadt (Ed.): „Multi-functional Card Terminal", Version 1.01, 1998-06-30; Draft for "Arbeitsgemeinschaft Karten im Gesundheitswesen"

20 Annex D: Implementation of an DIABCARD Security Environment

20.1 Application Security for the DIABCARD Client System (Phase I)

In this chapter, the integration of application security in the DIABCARD client system is described in detail. First, the basic agreements regarding the integration of security services are given resulting in an implementation scheme for the different layers as required in the previous chapter. Afterwards, the key objects on the smartcard are presented and the functionality of the DIABCARD Security DLL is introduced. Finally, the realisation of the application security services is given for each layer.

20.1.1 Basic Agreements regarding the Integration of Security Services

The DIABCARD security solution has been developed as a modular program system with clearly defined DLL interfaces. The integration of application security in complex software such as the DIABCARD client system must be considered as a difficult task. This is caused by the different interoperating components (see Figure 11.2) which has been created by various software developers of different institutions not always caring for modern software engineering, deploying different software development environments and not regarding security services. The availability and adaptability of the source codes have to be changed and equipment have been essential hurdles.

For source code management of the DCC (written in Borland Delphi V3), the developer from the Aristotle University of Thessaloniki (Greece) and the developer from the Medical Informatics Department in Magdeburg agreed that Thessaloniki is continuing the improvement of the DCC sending Magdeburg the newest source codes including detailed information where changes have been made to the prior version. Then, Magdeburg has integrated application security writing a security dynamic link library (DLL, C programming language) providing the functions to realise the security services needed (see Chapter 20.1.4). For cryptographic operations, this library uses the Security Development Environment for Open Systems (SECUDE™, written in C) from the GMD Darmstadt (see the annex for more information). SECUDE™ is a cryptographic toolkit offering many application programming interfaces including smartcard access. A more detailed description of SECUDE™ can be found in Chapter 19.5.3.

For testing purposes, the GSF Munich provided one IBM PC/SC card reader including serial cable for accessing the DIAB.PDC. A second PC/SC card reader is used for interoperability testing between two DIABCARD client applications.

The source code of the DCS was not available, because it had been developed by a German commercial software company (ACG SmartGate). The DIABCARD Data Access API (Java programming language) was written by IBM Germany and has been made available to the GSF Munich. However, the code could not be re-compiled due to problems concerning the Java IDE and matching towards the DCS program.

Therefore, all security services had been implemented by changing the source code of the DCC. Regarding the notion of *architectural placement*, the services are all placed on the DCC-layer, but their *impact*, i.e. the "location" *where they take effect* primary is spread over several layers as the DCC-layer, the PDD-layer, and the DCS-layer having individual targets for each. These layers of effect and the intended aim of the security service for each layer is given in Table 20.1 as an overview. In the following, when talking about layers and services, always the impact location is considered.

Table 20.1: Impact of Application Security Services and their intended Usage

DIABCARD Client Component / Application Security Service	DCC	PDD	DCS
Access Control	• User identification and authentication.	• Database locking. • Prohibit to interpret stored table data (applying confidentiality).	• Prevent bypassing the program start (applying confidentiality).
Authorisation	• Role management based upon the identification / authentication process. • Restrict functional rights and medical data access rights for authenticated users.	–	–
Accountability	• Non-repudiation of origin for medical data (responsibility).	–	–
Integrity	• Detect medical data manipulation. • Data origin authentication for medical data. • Detect program changes or replacements.	• Detect table file changes or replacements.	• Detect program changes or replacements.
Confidentiality	–	• Prohibit to interpret stored table data.	• Prevent bypassing the program start.

In the following paragraphs, the solution for integrating application security is presented. First, the key objects of the SC-PSE and the DIABCARD Security DLL are introduced. Then, the implementation of the security services for each layer as given in Table 20.1 is described in detail.

20.1.2 Security Objects in the Smartcard Personal Security Environment (SC-PSE)

As introduced in Chapters 11.7 and 11.9 , the TH.HPC stores all key objects and attribute information necessary to realise the services needed for application security. Because attribute certificates have not been available until the time of writing, authorisation and access control (management of roles, functional rights and data access rights) is based on the proven identity of the HP only. This proven identity is gained from the owner name of the SC-PSE that is represented as distinguished name (DN) after authentication. In this pilot, the test scenario includes two users (each owning an SC-PSE) and one Certification Au-

thority (CA, owning an SW-PSE) acting as TTP establishing a simple trusted certification path for a PKI as shown in Figure 20.1.

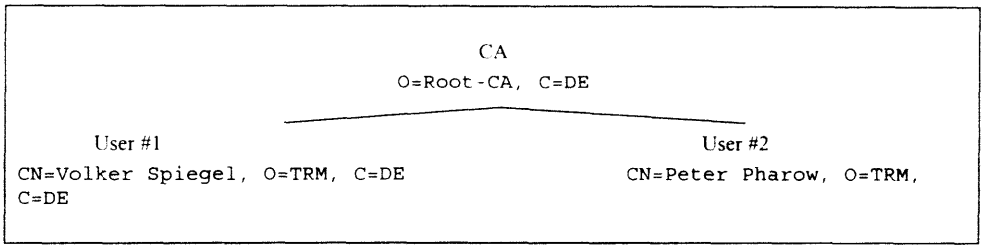


Figure 20.1: The Simple Trusted Certification Path for the SC-PSE PKI

The table of contents (TOC) of one of those user SC-PSEs is given in Figure 20.2 (generated by SECUDE™). The TOC contains information about the creator of the PSE ("created by", which is the login name of the person acting as CA), the date and time of the creation process, whether the PSE has one or two key pairs and some other data about storage capacity and the software extension. Most important is the listing of objects. After each object name, the object length (in Bytes, 1 Octet is equal to 1 Byte) or key size for private keys and a shortcut indicating the storage area (SC for SmartCard) is given. In the following paragraphs, each object of the TOC is introduced explaining the aim and use.

```
Table of Contents of PSE E:\SECUDE\tcos_ext.pse:
Created by: MedInf      C: May 20 17:19:39 1999 / TWO keypairs
SmartCard with 1258 bytes PSEFile (3130 bytes available), unused SW ex-
tension with default path 'E:\SECUDE\tcos_ext.pse'
Objects:
 1. SignSKey       (32 octets) SC
 2. CrlSet         (92 octets) SC
 3. PKList         (324 octets) SC
 4. SCertCA        (406 octets) SC
 5. EncSKey        (43 octets) SC
 6. PKRoot         (361 octets) SC
 7. EncCert        (492 octets) SC
 8. DecSKnew       (key file 3, size 1024) SC
 9. SignCert       (492 octets) SC
10. SignSK         (key file 2, size 1024) SC
```

Figure 20.2: Contents of a user SC-PSE

For user-related services, the SC-PSE contains two different RSA key pairs (objects "SignCert"/"SignSK" for signing and "EncCert"/"DecSKnew" for encryption which is not used in this pilot yet) each having 1024 Bit key size for enhanced security. The public keys (objects "EncCert" and "SignCert") are embedded in X.509v3 certificates. They are stored in a local directory as explained in 20.1.3 whereas the private keys (object "DecSKnew" and "SignSK") are kept securely on the smartcard non-readable for everyone (they are generated in the chip and never leave their secure environment for the whole life cycle). Having a directory service ready, local management of certificates (object "PKList") or revocation lists (object "CrlSet") is not applied in this pilot. However, these objects have minimal content and are required by the security software SECUDE™ for operating (generated by

default). An example for a public key printout is presented in Figure 20.3 showing the object "SignCert" (generated by SECUDE™). The object "EncCert" is very similar and for that reason not included here.

```
     Version:                2 (X.509v3-1996)
 SubjectName:                CN=Volker Spiegel, O=TRM, C=DE
  IssuerName:                O=Root-CA, C=DE
SerialNumber:                7 (decimal)
    Validity  -  NotBefore: Thu May 20 17:21:41 1999 (990520152141Z)
                  NotAfter: Sat May 20 17:21:41 2000 (000520152141Z)
    Public key Fingerprint: 706F 64CE 7A77 2CAC FB64 1799 F39E B30E
    SubjectKey:             Algorithm RSA (OID 1.2.840.113549.1.1.1),
NULL

Certificate extensions:
Authority Key Identifier: CF22 DE8B 5E76 AFD0 295C A0D1 D756 13F0 0741
3BAD
Subject Key Identifier:   39F2 E6FD 12D6 B4DB AF79 0A33 F091 4A97 7A6D
A31F
Key Usage:                    (CRITICAL) digitalSignature nonRepudiation
 Basic Constraints:            NOT allowed to act as a CA !
 Signature: Algorithm md5WithRsaEncryption (OID 1.2.840.113549.1.1.4),
NULL
 Certificate Fingerprint:
52:8E:00:51:63:1D:C6:FB:8B:52:24:AB:BF:E2:C3:3E
```

Figure 20.3: Public key for Signing (SignCert)

A default link to the CA is generated by SECUDE™ and provided by the object "PKRoot" containing the signature verification certificate of the CA (including the DN, but without the signature). Because of the security policy defined, requiring to check each certificate itself *before* usage, another object called "SCertCA" is held additionally representing the complete CA verification certificate including the signature. Since this certificate is very similar to the certificate displayed above, it is not depicted here.

As explained in Chapter 11.9, different symmetric keys concerning the services confidentiality and integrity are set up for the treatment of files. Therefore, the object "SignSKey" is installed for providing integrity using a keyed hash function as message authentication code (MAC) based on block ciphers. Here, the algorithm MD5-DES3-EDE2-CBC is used. Currently, this is the strongest MAC-algorithm SECUDE™ has to offer. For confidentiality, a strong symmetric key (algorithm DES3-EDE3-CBC) is stored on the smartcard in the object "EncSKey". In this scenario, these keys cannot be changed by the user bearing the security weakness that all users have the same keys on their smartcards allowing them to decrypt or verify files that had been encrypted or signed by other users before.

For enhanced security measures, these two symmetric keys may be seen as session keys changeable by the current user of the TH.HPC. Since the keys can only be changed on the smartcard that is currently inserted, only one user owns the proper keys and will be capable of decrypting or verifying files. It is not possible to store these keys in a so-called SC-extension only available at the local workstation due to the fact that many systems may be involved and that one user cannot open the SC-extension of another (the SC-extensions are secured by random generated DES-keys).

To conclude, the first scenario is preferred and has to be used if more than one user is working with a local workstation having a high level of security whereas the other scenario may be applicable if maximum security is needed having only one user per local workstation. Both scenarios are implemented in the DIABCARD Security DLL as described in Chapter 20.1.4.

An alternative to the second scenario is the use of a ticket server providing a transaction-related key to each authorised user for this specific transaction (sequential mode). This procedure, well-know from the Kerberos protocol, is currently introduced for securing multi-user information systems.

20.1.3 Directory Services

Following Chapter 11.12, a Public key Infrastructure (PKI) has been established (see Chapter 20.1.2). Additionally, directory services for public key certificates and revoked certificates had been realised. Due to the limited amount of participating users, local services were preferred and implemented using the AF-Database (authentication framework) interface of SECUDE™ storing the certificates in a directory hierarchy tree on the hard disk of each DIABCARD workstation. The interface allows to enter (CA only) and to retrieve certificates from the directory and the CRL. Following this approach, network traffic is limited and performance is sped up.

20.1.4 The DIABCARD Security DLL: Functions for Application Security

For the integration of application security in the DIABCARD client system, a software component had been developed by the Magdeburg Medical Informatics Department encapsulating all the functionalities needed for realisation of the security services. This DIABCARD Security DLL can be bound to any other application which needs application security services as well (re-usability). For cryptographic operations, this library uses the AF-API and SECURE-API from SECUDE™ (see Chapter 19.5.3). In the following subsections, the functions for each security service are given and described shortly using the C-style program syntax. As mentioned in the common security model, the first service (20.1.4.1) deals with communication security (authentication) and application security (access control) as well.

20.1.4.1 Authentication and Access Control

For identification and authentication when starting the DCC, the SC-PSE has to be opened by inserting the TH.HPC in the card reader and typing the correct PIN on the pad. After leaving the application, the SC-PSE must be closed as well. This functionality is provided by:

- `BOOL **USR_OPEN_PSE**(char *pszPSEName);`

Parameters: [IN]: `char *pszPSEName`: name of the SC-PSE
Return: TRUE, if PSE was opened successfully (user authenticated)
FALSE, if PSE was NOT opened successfully (authentication failed)

- `void **USR_CLOSE_PSE**();`

20.1.4.2 Authorisation

As mentioned in Chapter 20.1.2, authorisation including management of roles as well as restriction of functional rights and data access rights is based on the proven identity gained from the owner name of the SC-PSE after authentication. For managing this requirement, the following operations are included:

- `char* USR_GET_DName();`

Return: distinguished name (DN) of the SC-PSE owner as character string, if successful
NULL, if function failed

- `void USR_FREE_DName(char **pszString);`

Parameters: [IN/OUT]: `char **pszString`: DN of the SC-PSE owner to be freed

20.1.4.3 Accountability

The accountability service concerns the provision of responsibility for medical data assuring the data origin is provable without repudiation. This is realised by user-related digital signing of data using the own private key and verification applying the public key:

- `BOOL USR_SIGN_Data(char *pszData2Sign, int nLenData2Sign, char **pszSignature, int &nLenSignature, void **pvoidAlgId);`

Parameters: [IN]: `char *pszData2Sign`: medical data to be signed
`int nLenData2Sign`: length of data in bytes
[OUT]:`char **pszSignature`: digital signature
`int &nLenSignature`: length of digital signature in bits
`void **pvoidAlgId`: pointer to algorithm identifier

Return: TRUE, if signing was successful
FALSE, if signing was NOT successful

- `BOOL USR_VERIFY_Data(char *pszData2Verify, int nLenData2Verify, char *pszSignature, int nLenSignature, void **pvoidAlgId, char *pszDName);`

Parameters: [IN]: `char *pszData2Verify`: medical data to be verified
`int nLenData2Verify`: length of data in bytes
`char *pszSignature`: digital signature
`int nLenSignature`: length of digital signature in bits
`void **pvoidAlgId`: pointer to algorithm identifier
`char *pszDName`: DN of signer

Return: TRUE, if verification was successful
FALSE, if verification was NOT successful

- `Void USR_FREE_Signature(char **pszSignature, void **pvoidAlgId);`

Parameters: [IN/OUT]: `char **pszSignature`: signature bits to be freed
`void **pvoidAlgId`: algorithm identifier to be freed

20.1.4.4 Integrity

Generally, this service has two different aims according to the kind of data treated (see Chapter 20.1.2). For (medical) data, integrity is needed in the sense of detection of data manipulation and offering means of authenticating the data origin. These needs are realised by digital signing the medical data. Therefore, the functions presented in Chapter 20.1.4.3 are used. Secondly, if program files are concerned, the objective is to detect file manipulation or replacements (integrity service). For realisation, MACs are calculated over the file data:

- `BOOL USR_SIGN_File(char *pszInFileName, char *pszOutSigFileName, BOOL bChangeSessionKey, int nSessionKeyAlg);`

Parameters: [IN]: `char *pszInFileName`: name of file to be signed

char *pszOutSigFileName: name of signature file
BOOL bChangeSessionKey: if TRUE, the symmetric key
on the SC-PSE will be changed; FALSE means not to
change the key
int nSessionKeyAlg: if symmetric key is changed, use
this algorithm for the new key

Return: TRUE, if signing was successful
FALSE, if signing was NOT successful

- `BOOL USR_VERIFY_File(char *pszInFileName, char *pszInSigFileName);`

Parameters: [IN]: `char *pszInFileName`: name of file to be verified
`char *pszInSigFileName`: name of signature file

Return: TRUE, if verification was successful
FALSE, if verification was NOT successful

Both scenarios of permanent or changeable symmetric keys (as explained in Chapter 20.1.2) can be realised by the signing function. In this pilot, a permanent symmetric key is applied using the MD5-DES3-EDE2-CBC MAC algorithm. For changing keys, the boolean parameter `bChangeSessionKey` has to be set to TRUE specifying the MAC algorithm identifier in the parameter `nSessionKeyAlg` for the newly generated key. Then, this key is used for the signing operation. The following MAC algorithms can be applied (restricted by SECUDE™): MD5-IDEA, MD5-DES-CBC and MD5-DEC3-EDE2-CBC.

20.1.4.5 Confidentiality

The confidentiality service deals with preventing interpretation of program data or unauthorised program start-ups and is treating program files only. As discussed in Chapter 20.1.2, symmetric techniques are applied. Like the signing operation given in Chapter 20.1.4.4, this function allows both scenarios of permanent or changeable keys:

- `BOOL USR_ENCRYPT_File(char *pszInFileName, char *pszOutFileName, BOOL bChangeSessionKey, int nSessionKeyAlg);`

Parameters: [IN]: `char *pszInFileName`: name of file to be encrypted
`char *pszOutFileName`: output file name
`BOOL bChangeSessionKey`: if TRUE, the symmetric key
on the SC-PSE will be changed; FALSE means not to
change the key
`int nSessionKeyAlg`: if symmetric key is changed, use
this algorithm for the new key

Return: TRUE, if encryption was successful
FALSE, if encryption was NOT successful

- `BOOL USR_DECRYPT_File(char *pszInFileName, char *pszOutFileName);`

Parameters: [IN]: `char *pszInFileName`: name of file to be decrypted
`char *pszOutFileName`: name of output file

Return: TRUE, if decryption was successful
FALSE, if decryption was NOT successful

20.1.5 Security Services for the DIABCARD Core Application (DCC)

As explained in the security requirements above (see Chapter 11.8), the application security services access control, authorisation, accountability, integrity and confidentiality have

been implemented. Moreover, all services on the DCC-layer – except the integrity services for the DCC program files – are user related dealing with medical data only. In the following subparagraphs, the implementation issues for each service concerning aim (see Table 20.1) and realisation are described in detail.

Following the demand for interoperability between the different DIABCARD test sites in Germany, medical data on the DIAB.PDC is not encrypted.

20.1.5.1 Access Control

User identification and authentication are the objectives of the access control service. The existing DCC has only very limited access control using a name and password dialog box (`procedure FormShow` in `main.pas`, `login.pas`) that is presented after the start-up. As discussed in Chapter 5, this is not adequate by far for user identification and authentication in health information systems. Therefore, this dialog has been replaced by the request to insert the HP's TH.HPC typing the correct PIN using the functions presented in Chapter 20.1.4.1. If the HP passes the verification of ownership (possession of the TH.HPC) and knowledge (PIN), the procedure of strong authentication using cryptographic algorithms starts, the identity is proven and the DCC gives access to the DIABCARD workstation. Otherwise, the DCC ends its operation here.

If an HP has been authenticated this way, there is no further authentication dialog for accessing the PDD or the DCS. Bypasses, namely accessing the PDD (that is storing all medical data) or starting the DCS directly without authorisation (unauthorised application usage) are prevented by appropriate means of confidentiality applied in these two layers as explained in Chapter 20.1.6 and 20.1.7. After passing the authentication dialog, the integrity of the DCS and PDD files is checked. Then, the DCS files are decrypted and the server is started automatically. Closing the DCC entails encrypting the DCS files again.

20.1.5.2 Authorisation

Restrictions are necessary for authenticated users concerning the acquisition and handling of medical data. Therefore, a detailed access control management has been implemented processing the *functional rights* (program functions) as well as the *data access rights* within a function like selection, creation, deletion, reading, writing, alteration of data and *right management*. This prevents unauthorised disclosure and manipulation of data, respectively.

Minimal authorisation was already available in the DCC only featuring and not strongly controlling two roles: user and administrator. The HPC provides roles, which have been implemented in the DCC, i.e. different, access rights for the various groups. After authentication, each HP is only permitted to process certain functionality on medical data it is allowed to access. Authorisation inside the DCC is based on the proven identity of the HP obtained from the opened SC-PSE by retrieving and evaluating the distinguished name (owner name of the SC-PSE). For realisation, the operations given in Chapter 20.1.4.2 are applied. The existing "SecurityLevel" (see function `CheckSecurityLevel` in `global.pas` and table `users.db`) in the DCC is used for grouping purpose. Moreover, the professional identifier connected with the data items has been adjusted to realise personal right management.

20.1.5.3 Accountability

Next, the responsibility of the Health Professional for data items has been realised enabling to determine the originator of the data without repudiation. For that purpose, user related digital signatures has been integrated in the DCC using the interface shown in Chapter 20.1.4.3.

The DCC accesses the DIAB.PDC for reading and writing operations using the TCP/IP DCS-Access Service[49] [DCS_1998]. This service is passing the commands and data to the DCS and returns the results by the same way. The data elements on the DIAB.PDC are represented by items consisting of the field description, data value, performer identifier, and timestamp. All commands and data transferred between the DCC and the DCS are TLV-encoded as described in the DIABCARD Server Interface description [DCS_1998]. The PDD is accessed from the DCC via the Borland Database Engine (BDE).

Concerning the source code of the DCC, all DIAB.PDC and PDD operations are performed item by item[50]. Therefore, signing and verification inside the DCC are performed on *data item level*. In general, the digital signature is generated and attached to the data items before writing to the PDD or the DIAB.PDC. Verification of the signatures is performed after reading the items from the PDD or the DIAB.PDC. For signing, the digital signature and the distinguished name of the originator are attached to the data value separated by special characters allowing to identify each component.

Because this method extends the data field, the length fields in the TLV-encoding of each data item has to be adjusted accordingly when accessing the DIAB.PDC. Furthermore, each item has a special type (number, string, enumerated, date, year) and the DCS performs type checking that is taken into account.

Signing on *group level* may be preferred to item-signing because of performance and memory storage reasons[51], but had not been considered due the lack of source code for the DCS and compiling problems of the DIABCARD Data Access API (see Chapter 20.1.1). Since the PDC operations inside the DCC are looping over each item to process the whole group, the signing of groups inside the DCC calculating the signature over all items of a group might be imaginable. Then, the signature has to be bound to the group. A possible solution would be to define an additional item like "digitalSignature" for each group. However, this imposes far-reaching and serious changes with huge impact of all programs and database tables and is therefore not feasible. In addition, the DCS source code is not available for adjustment and there is the problem to represent the connection between the signature and the group on the DIAB.PDC as the data elements on the card are items. Finally, signature generation and verification inside the DIABCARD Data Access Java API[52] [DDATA_1998] is not applicable, because personal digital signatures are not be available on this level since there is no personal reference to the HP (the TH.HPC is opened in the DCC).

Regarding the source code of the **DCC**, the TCP/IP DCS-Access Service is implemented in `dcsconn.pas` by low-level and high level-operations. The data items are TLV-decoded/encoded (as defined in [DCS_1998]) and parsed/constructed by the following high-level functions and procedures that are also retrieving the answer from the DCS via the socket interface:

- `function ReadData: boolean;`
- `function WriteData: boolean;`
- `procedure GetCurrPatientDetails;`

---

[49] This service is only needed by applications developed in other programming languages than Java (like the DCC).

[50] The DCC reads and writes the groups item by item, but the Group Module and the Cache Module of the DCS provide grouping for writing and reading, respectively. Two database tables are lying behind these modules (table itemgroups and dsviews in the Microsoft Access database itemgroups.mdb).

[51] The maximum storage capacity of the DIAB.PDC currently is 16 Kbytes only (approx. 12 Kbytes for medical data) which may become problematic concerning the additional bytes needed for the signature. There is no storage restriction regarding the PDD.

[52] The C API description is not considered here, cause the DCS is written in Java and uses Java API calls.

- `procedure PutCurrPatientDetails;`
- `procedure CheckViews;`

Returned items consist of a field description, data value, performer identifier, and time-stamp. The latter two fields are not used in the `PutCurrPatientDetails` or `GetCurrPatientDetails` procedures (see below). The listed low-level procedures are called from the higher-level and send the encoded data or command through the socket interface to the DCS:

- `procedure AskItemRead(tName: string);`
- `procedure AskItemAvailable(tName: string);`
- `procedure AskItemChange(tName, tValue: string);`
- `procedure AskItemWrite(tName: string);`
- `procedure AskViewAvailable(tName: string);`

The high-level procedures and functions of the TCP/IP DCS-Access Service itself are invoked by operations of other modules (`.pas`-files) of the DCC. A dependency graph of these calling hierarchy is given in Figure 20.4.

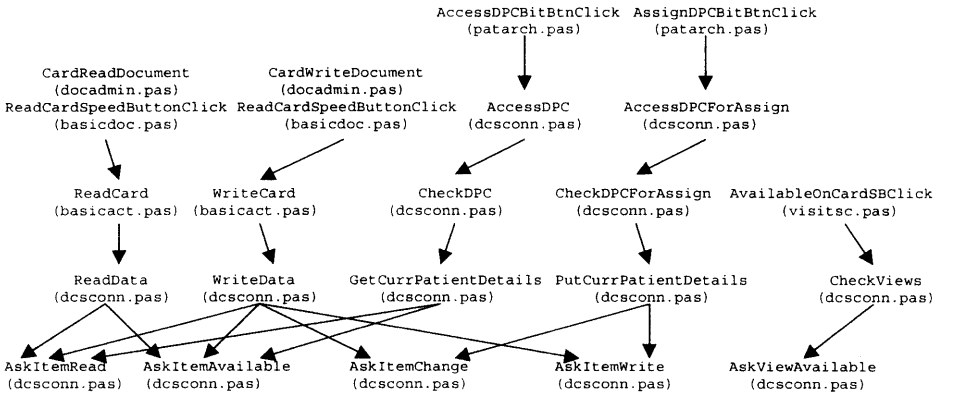


Figure 20.4: Calling Hierarchy for Item Operations on the DIAB.PDC

In the next paragraphs, the background of the high-level procedures and functions of the TCP/IP DCS-Access Service concerning their purpose and workflow in the hierarchy are described showing why some calls have been adjusted to integrate signing or verification and some not.

After the TH.HPC has been presented successfully (invoked by button "Professional"), the buttons for the DIAB.PDC access are available (see procedures `AccessDAC` and `CheckDAC` of `dcsconn.pas`). If pressing the button "Patient", the procedure `AccessDPC` is called. Having presented the DIAB.PDC successfully, the procedure **`GetCurrPatientDetails`** is invoked automatically (see procedure `AccessDPC` and `CheckDPC` of `dcsconn.pas`). This procedure retrieves the basic information for the patient as surname, forename, date of birth, and sex by first testing the availability of the item calling `AskItemAvailable(ItemName)` following the data transfer using `AskItemRead(ItemName)`. If the patient is not present in the DCC, a new entry can be created.

Because in this workflow the medical data is read from the DCS by socket operations inside the procedure `GetCurrPatientDetails`, the verification process takes place here.

If a DIAB.PDC is assigned to a new patient, four items of the basic information (see last paragraph) are written on the card. This is performed in the procedure **`PutCurrPatientDetails`** after pressing the button "Assign Card" (see procedures `AccessDPCForAssign` and `CheckDPCForAssign` of `dcsconn.pas`). Each item is written by first invoking `AskItemChange(ItemName, NewValue)` and then `AskItemWrite(ItemName)`. Because in this workflow the medical data is passed to the DCS inside the procedure `AskItemChange`, the signing process is performed here.

The procedure **`CheckViews`** is called when the button "Available on Card" in the visits menu is pressed. For each entry (group name) of the table `DSVIEW.DB` that has a panel view, the procedure tests if this view is stored on the DIAB.PDC by calling `AskViewAvailable(ViewName)`. In case of availability, this view is added to a tree view as top element (if level is 1) or child element (if level is 2). The tree view is shown to the user for information purposes. No digital signature (neither signing nor verification) is needed for this workflow, because there is no medical data transfer.

Any documents (custom or basic) that have to be read from or written to the DIAB.PDC need the invocation of the procedures `ReadData` or `WriteData`, respectively (buttons "Read Card" or "Write Card" of the DCC GUI).

According to the source code of **`ReadData`**, the read operation loops (using `WHILE`) *item-wise* by first calling `AskItemAvailable(ItemName)` and then `AskItemRead(ItemName)`. All items from each group of the selected document are read if available on the PDC and stored in the tables (`.DB`-Files). For looping the list `ItemsList` is processed that contains all these items taken from the related TBL-file[53] (except those entries that are `SEPARATORs`, `LABELs` or have the type `GROUP`, see procedure `CreateItemsList` in `basicact.pas`). Because in this workflow the medical data is read from the DCS by socket operations inside the procedure `ReadData`, the verification process takes place here. Since data items are written into the PDD as well, the digital signature and encryption of the PDD files are refreshed (see Chapter 20.1.6).

The write operation of **`WriteData`** is performed *item-wise* as well. First, all existing items for that document are read from the PDC using `AskItemAvailable(ItemName)` and `AskItemRead(ItemName)` looping (using `WHILE`) over all items of `ItemsFullList` (this is created from `ItemsList` containing all items of `ItemsList` plus all the items of the group that each `ItemsList[i]` belongs to, see procedure `CreateItemsFullList` in `basicact.pas`). Items not available on the DIAB.PDC are read from the data fields of the table (these items have been changed recently and are therefore written to the `.DB`-tables after entry). They have new values with no prior old data. Items available on the PDC represent old values that are already stored in tables. They are read from the tables and merged with the new values. This set of items (representing the current state of the `.DB`-tables) containing the new changed items and the old unchanged items is written to the PDC in the second step. In this part, for each item contained in `ItemsList`, the procedures `AskItemChange(ItemName, NewValue)` and `AskItem-`

[53] Each document has a corresponding TBL file that defines which items will be included in the document and what groups they belong to. The information in the TBL files is used to automatically generate the screen forms of the documents and can be used e.g. for customisation of the language and local terms.

`Write(ItemName)` are called looping (using `FOR`) over all items to write the whole group to the PDC.

Following this concept, the physician is not responsible only to the items she or he changed, but also to the items that are adapted from prior entries of possibly different physicians (all items in a group take the same performer identifier). A confirmation dialog assures the awareness of this process. Moreover, the data on the PDC is overwritten (deleted first writing a sequence value of zero) keeping always the newest data, whereas all versions are kept in the PDD.

For example, the document "foot" (see `foot.tbl` in `\tbl`) consists of the groups symptoms, examination, amputation, bones, and outcome which respectively have many items. All items and their values are stored in the PDD namely in the file `foot.db` in `\data`. The `ItemsList` consists of 111 entries (F3200, F3250, ...) and the `ItemsFullList` of 115 entries.

Please note that the documents "BasicInformationSheet" and "DiabetesPassport" use a special procedure `CreateItemsList` in `extraact.pas`.

Because in this workflow the medical data is read from the DCS by socket operations inside the procedure `WriteData`, the verification process takes place here. As already mentioned above, a signing process has to be performed inside the procedure `AskItemChange`. Since data items are read from the PDD as well, signature verification and decryption of the PDD files are carried out too (see Chapter 20.1.5.3).

Digital signatures for the data stored in the **PDD** can be applied by item or group, basically. *Signed by item* is problematic e.g. if creating indexes, because each item is extended by distinguished name and signature including component separators. *Signing by group* requires an extra entry for each group to store the signature losing the bound between the signature and the data over that it has been calculated. Furthermore, the signature may become invalid if the sequence of items over those the signature has been calculated is not the same as in the verification process (problem of group creation, ordering and resolution).

Therefore, the most applicable way of providing signatures in the PDD has been implemented, performing signature operations by item having the problem that the indexes are influenced by the overhead of distinguished name, signature and separator characters.

Concerning the source code of the DCC, the database core operations are performed in `datamod.pas` (e.g. `PostTables` is looping over all data fields). These operations are called from procedures in `basicact.pas` (e.g. `Save`) which are invoked in `basicdoc.pas` (e.g. `SaveSBClick`) or `docadmin.pas`. At least, all procedures in these files have been changed.

20.1.5.4 Integrity

This service has two different aims according to the kind of data treated. For medical data, integrity is needed in the sense of detection of data manipulation and offering means of authenticating the data origin. These needs are realised by digitally signing the medical data, which already have been discussed in the section concerning the accountability service (see Chapter 20.1.5.3).

If program files are concerned, the objective is to detect file manipulation or replacements. For realisation, symmetric keys are used for the calculation of MACs over the file data (see Chapter 20.1.2) using the functions given in Chapter 20.1.4.4. Integrity check of the DCC program files is performed after the HP has successfully passed the authentication dialog (and before starting the DCS, for this issue see Chapter 20.1.7), because the keys can only

be accessed if the SC-PSE has been opened. Re-calculation of signature is performed if a change of a program file is necessary.

20.1.6 Security Services for the Paradox Database (PDD)

Regarding the security requirements given in Chapter 11.8, the application security services access control, integrity and confidentiality are implemented. All services on the PDD-layer are not user related dealing with PDD files only. In the following subparagraphs, the implementation issues for each service concerning aim (see Table 20.1) and realisation are described in detail.

20.1.6.1 Access Control

For restricting the access of the PDD, database table locking as well as cryptography-based mechanisms like database file encryption are applied. The latter is described in Chapter 20.1.6.3. Due to the lack of stored procedures or similar means in the PDD, it is not possible to integrate TH.HPC authentication on the database level.

The locking mechanism is applied for each table using the Borland Delphi IDE changing the table properties (exclusive access). A drawback with this concept is the limited range of effect. Only the tables that are currently open under the DCC are locked for third-party applications trying to get access to the database tables whereas the other tables are not locked. It is definitely not practicable to open all tables on start-up of the DCC exclusively.

Furthermore, table passwords can be applied. A password (up to 15 characters) can be defined for each database table using the Borland Delphi Database Desktop (Tools → Utilities → Reconstructure → Table Properties → Password Security). All files belonging to the password protected table (e.g. `.DB` for the Paradox table, `.PX` for the primary index of the Paradox table, `.XGn/.YGn` for the composite secondary index of the Paradox table) are scrambled in a certain (proprietary) way using low cryptographic techniques if any at all. If the table is opened, the password is required to access the table data. Password caching makes sure that the user is not bothered that much. Taking all this into consideration, the password mechanism is not used due the given security weaknesses.

20.1.6.2 Integrity

Moreover, all the PDD database table files are protected by integrity detecting file changes or replacements. These files are `.DB` for the Paradox table, `.PX` for the primary index of the Paradox table, `.XGn/.YGn` for the composite secondary index of the Paradox table. This is achieved by the usage of symmetric keys for the calculation of MACs over the file data (see Chapter 20.1.2) applying the functions given in Chapter 20.1.4.4. Integrity checking of certain PDD files is performed after reading table data contained in those files. After closing the table data, the corresponding PDD files are signed again.

20.1.6.3 Confidentiality

For restricting the access of the PDD tables, confidentiality is applied to all database table files (including `.DB`, `.PX`, `.XGn` and `.YGn`) prohibiting the interpretation of stored table data e.g. by external database viewers or tools that have no appropriate means of authentication. This is accomplished by the usage of symmetric keys for encryption of the file data (see Chapter 20.1.2) applying the functions given in Chapter 20.1.4.5. Decryption and encryption is performed for each table on demand, i.e. each single table is decrypted if accessed and encrypted if closed. This gives a very high level of security protection, especially for those tables that are not opened (and therefore not locked) during the runtime of the DCS. A minor drawback is a minimal loss of performance.

20.1.7 Security Services for the DIABCARD Server (DCS)

Following the security requirements in Chapter 11.8, the application security services access control, integrity and confidentiality are being implemented. All services on the DCS-layer are not user related dealing with DCS program files only. In the following subparagraphs, the implementation issues for each service concerning aim (see Table 20.1) and realisation are described in detail.

20.1.7.1 Access Control

Because the source code of the DCS is not available, only limited security measures are implemented on this level. Since there should be no further authentication dialogue for accessing the DCS when authenticated to the DCC already, the DCC establishes a security context to the DCS by applying confidentiality (see Chapter 20.1.7.3).

20.1.7.2 Integrity

The DCS files are protected by the integrity service detecting program changes or replacements. This is accomplished by applying symmetric keys for the calculation of MACs over the file data (see Chapter 20.1.2) applying the functions given in Chapter 20.1.4.4. Integrity checking is done after passing the authentication dialog successfully and before the server is starting automatically. Re-calculation of the signatures will be performed if a change of the DCS files is necessary.

20.1.7.3 Confidentiality

For restricting the access to the DCS, i.e. starting the server without permission, all files for the DCS are encrypted applying symmetric keys (see Chapter 20.1.2) using the functions given in Chapter 20.1.4.5. After passing the authentication dialog successfully, all files are decrypted and the server is started automatically. After exiting the DCC, the server is shut down and all DCS files are encrypted again.

20.2 Communication Security for the DIABCARD Client System (Phase II)

For the exchange of sensitive personal medical data between the DIABCARD workstation and the departmental information system, appropriate security measures have to be provided as required in Chapter 11.10 for a trustworthy communication. The security solution presented in this chapter is based on the MEDSEC project results using the Secure File Transfer Protocol (SFTP) for communication security. Therefore, only the main aspects and results are given. For further details on communication security services and SFTP, the MEDSEC documents [MEDSEC_D30, MEDSEC_D31] have to be consulted.

20.2.1 User-Related Communication Security

As mentioned in Chapter 11.10, secure communication can be provided either user-related or not user-related reflecting the accountability of the different parties involved. According to the general communication security solution developed in the framework of the European MEDSEC project [MEDSEC_D30, MEDSEC_D31], the services and mechanisms defined, specified and implemented are the same for both scenarios. Only the authentication procedure and the authenticated principals are different.

In the user-related scenario, the user authenticates himself/herself using his/her HPC. After identification and verification as the cardholder using the PIN, the user PSE at the HPC is opened afterwards performing the strong authentication procedure based on cryptographic algorithms.

In the not user-related scenario, the third party opens its PSE using an appropriate authentication token (key, password). Then, the strong mutual authentication can proceed. Because the PSE needed must be implemented at the local site, this PSE is a software one, which is described in detail in Chapter 20.2.2.

20.2.2 Security Objects in the Software Personal Security Environment (SW-PSE)

As introduced in Chapter 11.10, the SW-PSEs store all key objects and information necessary to realise the services needed for not user-related communication security. In the test scenario, two systems (a workstation and a server) are involved each having an SW-PSE. Moreover, a Certification Authority (CA) is acting as TTP establishing a simple trusted certification path (see Figure 20.5). For this PKI, another CA is installed as the one presented in Chapter 20.1.2, because a different kind of principals is involved. This PKI is dealing with systems only and not with users, and therefore is requiring a slightly different security policy offering the same level of security (e.g. same key size).

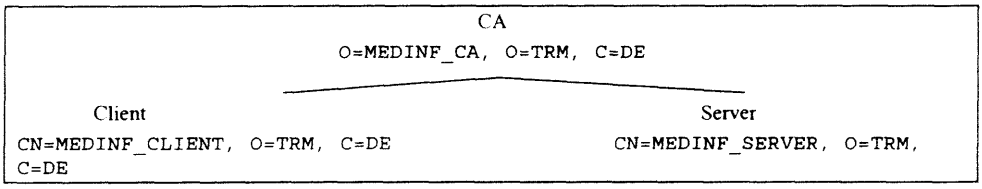


Figure 20.5: The Simple Trusted Certification Path for the SW-PSE PKI

The table of contents (TOC) of one of those system SW-PSEs is given in Figure 20.6 (generated by SECUDE™). The TOC almost contains the same objects as shown in Chapter 20.1.2. Therefore, this section has to be consulted for further details.

```
Table of Contents of PSE C:\SECUDE\server.pse:
Created by: MedInf      C: Apr 23 12:31:56 1999 / TWO keypairs
PSEFile (v2) is DES encrypted / not compressed
Objects:
 1. CrlSet     C: Apr 23 12:33:39 1999 (108 octets)
 2. PKList     C: Apr 23 12:33:39 1999 (424 octets)
 3. PKRoot     C: Apr 23 12:32:30 1999 U: Apr 23 12:33:39 1999 (511 oc-
tets)
 4. EncCert    C: Apr 23 12:32:30 1999 U: Apr 23 12:33:39 1999 (569 oc-
tets)
 5. DecSKnew   C: Apr 23 12:32:30 1999 (156 octets)
 6. SignCert   C: Apr 23 12:32:16 1999 U: Apr 23 12:33:38 1999 (569 oc-
tets)
 7. SignSK     C: Apr 23 12:32:15 1999 (156 octets)
```

Figure 20.6: Contents of a system SW-PSE

However, no symmetric keys are stored in the SW-PSEs. As explained in Chapter 10, hybrid encryption is performed for the data connection of SFTP. A strong symmetric session key (key size 128 Bits and above) is applied for bulk data encryption that is renewed for each transfer. The objects "EncCert"/"DecSKnew" are needed for encryption/decryption, and the objects "SignCert"/"SignSK" are required for verification/signing. The public keys are embedded in X.509v3 certificates (objects "EncCert" and "SignCert") and the key size amounts to 1024 Bit for all asymmetric keys. These keys are stored in a local directory as

explained in 20.1.3. All objects are combined to one file stored on the hard disk (PSEFile). For security reason, this file is DES encrypted.

20.3 References

[DCS_1998] T. Demmer, E. Neufeld, A. Steyer: Diabcard Server Description, Version 1.01, 30.10.1998, ACG-SmartGate Software GmbH, Köln.

[DDATA_1998] R. Sulzmann: DiabCard Interface Description, Version 1.6, 19.05.1998, IBM Deutschland Entwicklung GmbH, Böblingen.

[MEDSEC_D30] Blobel B, Spiegel V, Krohn R, Pharow P, Engel K (1998) Standard Guide for HL7 Communication Security. ISIS MEDSEC Project, Deliverable 30, August 1998. http://www.math.aegean.gr/medsec/d1.html

[MEDSEC_D31] Blobel B, Spiegel V, Krohn R, Pharow P, Engel K (1998) Standard Guide for Implementing EDI Communication Security. ISIS MEDSEC Project, Deliverable 31, August 1998, http://www.math.aegean.gr/medsec/d1.html

[Varv_1997] Dr. A. Varvitsiotis: Euromed-ETS Test Coverage, Institute of Communication and Computer Systems (ICCS), University of Athens (NTUA), Version 3.0, 28. August, 1997.

21 Annex E: European Legal Framework for Healthcare Security

The following European legal instruments address the healthcare sector or have a direct impact on it [Blobel and van Eecke, 1999].

Table 21.1: Impact Regarding Confidentiality of Communication

95/46/EC	Directive on the protection of individuals with regard to the processing of personal data and the free movement of such data • good data, fair use, privacy, fair access, responsibility, access to information
97/66/EC	Directive on the processing of personal data and the protection of privacy in the telecommunication sector • prohibit listening, tapping or storing, erase traffic data, consent to marketing use
R(97)5	Council of Europe Recommendation on the protection of medical data

Table 21.2: Impact Regarding Electronic Documents

99/93/EC	Directive on a Community framework for electronic signatures • define essential requirements for electronic signature certificates and certification services, • minimum liability rules for services providers • define several levels of electronic signatures • stipulate that an electronic signature could not be legally discriminated against solely on the grounds that it is in electronic form • legal recognition of electronic signatures irrespective of the technology used
COM(1998)585	Green paper on Access to Public Sector Information - improve transparency, access, and fair pricing

Table 21.3: Impact Regarding Consumer Protection / Liability

92/59/EEC	Directive on the General Product Safety • no-fault liability system on the producer • the onus of proof is on the producer to show that the harm not to arise from the use of his good
93/42/EEC	Directive on Medical Devices - A medical device is any instrument, apparatus, appliance, material or article whether used alone or in combination, including software necessary for its proper application intended to be used on human beings for the purposes of diagnosis, prevention, monitoring, treatment or the alleviation of disease, injury or handicap or control of conception
2000/31/EC	Directive on certain legal aspects of E-commerce • liability of intermediaries, exemption for 'mere conduits' • dispute settlement schemes

Table 21.4: Impact Regarding Service Provision / Citizen Access

95/62/EC	Directive on the application of open network provision to voice telephony • EU Universal Service Obligation
98/10/EC	Directive on the application of open network provision to voice telephony and on universal service for telecommunications in a competitive environment • Guaranteed provision of service
97/7/EC	Directive on the protection of consumers in respect of distance contracts • right to written information about contracts, • right to withdraw

Table 21.5: Impact Regarding Internet Content

21/12/1998	Action Plan Promoting Safer Use of Internet • non-regulatory initiatives • hotlines • self-regulation • filtering • special emphasis on content harmful to children

Table 21.6: Impact Regarding Cryptography

OECD 1997	Recommendation of the Council concerning guidelines for cryptography policy and Guidelines for Cryptography Policy, 27 March 1997 • remove, or avoid creating in the name of cryptography policy, unjustified obstacles to international trade and the development of information and communications networks • promotion of international co-operation between governments
Wassenaar 1996	Arrangement on Export Controls for Conventional Arms and Dual-Use Goods and Technologies, revised in 1998. • export restrictions on cryptography signed by 33 countries • all cryptography products of up to 56 bits key length are free for export, • mass-market cryptography software and hardware of up to 64 bits key length are free for export, • the export of products that use encryption to protect intellectual property is relaxed • export of all other cryptography still requires a license
3381/94 EC	EU Dual-Use of goods regulation (amended by Regulation (EC) 837/95 of April 1995) and EU Council Decision No. 94/942/CFSP (last amended by Council Decision 98/232/CFSP) • license is needed for the export of cryptography hardware and software outside of the EU. An exception exists for the export of mass-market and public-domain software. • for a transitional period, the Regulation also requires a licence procedure for intra-Community trade of cryptography products

Table 21.7: Impact Regarding Computer Criminality

R (89) 9	Council of Europe Recommendation on computer-related crime of 1990 - stimulates the member states to legislate on computer criminality - lists the possible computer crimes (hacking, data manipulation, ...)
R (95) 13	Council of Europe Recommendation concerning problems of criminal procedure law connected with information technology of 1995 - stimulates the member states to adapt current procedural law to new technologies - lists the necessary legal instruments (seizure of data, network search...)

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X