Abstract

An observation chart contains a collection of information from several different health information systems used at a hospital. Today, health personnel often has to access these health information systems during patient care and manually register information from them into the observation chart. Integration of the health information systems which constitute an observation chart is therefore needed. Integration means that systems used by a large amount of users are put together in such a way that all users gain access to the information they need. An integration will increase the efficiency of information flow by automatically retrieving information from relevant health information systems into an electronic observation chart. These improvements in turn will hopefully result in better quality of patient care, reduced time spent on treating each patient and therefore also reduced costs.

This thesis describes a security focused integration architecture for an electronic observation chart system (EOC-system). This thesis also explores standards, strategies, laws and regulations relevant for the architectural description of the EOC-system. The EOC-system is going to be developed by CARDIAC, a company focusing on technology within health care, and the architectural description will be a support in this development process.

The architectural description for CARDIAC's EOC-system is based on the Model-based Architecture description Framework for Information Integration Abstraction (MAFIIA), which is an architectural description framework for software intensive systems with a specialization towards Information Integration Systems (IIS). The architectural description has also followed MAFIIA's two extensions, MAFIIA/H and MAFIIA/RBAC, which respectively relate to the health care domain and to role-based access control (RBAC).

The work with this thesis, following the MAFIIA architectural description framework, has resulted in a detailed and structured architectural description which sees the architecture from several viewpoints and describes different aspects of it. Security and integration are emphasized in the architectural description; a combination of a service-oriented and portal-oriented integration architecture is chosen and the security mechanisms digital signing, secure communication, auditing and access control are ensured.

Preface

This report is a result of the work with the master's thesis in *TDT4900 Computer and Information Science* during spring 2005. The work is performed at the Department of Computer and Information Science at the Norwegian University of Science and Technology (NTNU) in cooperation with CARDIAC AS.

The objective of this master's thesis project is to create a security focused integration architecture for an electronic observation chart.

We would like to thank everybody who has motivated and supported us and contributed to this work, specially our supervisor at NTNU, Lillian Røstad, and our supervisor in CARDIAC AS, Marius Kvitnes.

Trondheim, July 7, 2005

Mirela Divic

Ida Hveding Huse

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Introduction

This chapter contains a motivation for this thesis, a presentation of the goals and scope, together with an overview of the document organization.

1.1 Motivation

Development of information systems began little by little. In the beginning systems were developed to solve only very simple work tasks, but gradually information systems have become more and more complex. Still, there are several detached information systems supporting different work processes within an enterprise. Users therefore often have to access several different information systems when performing complex work tasks. But, users expect instant access to all business functions an enterprise can offer, regardless of which system the functionality may reside in. This requires information systems to be connected into a larger, integrated solution.

At present, integration of information systems is of special interest in the health sector. A lot of ongoing work is concerned with integration of health information systems. Larger hospitals in Norway today have hundreds of health information systems. Most of these systems are information systems which only perform simple tasks, such as the processing and storing of laboratory information. An integrated solution supports more complex tasks, which imply access to several different health information systems. Such a solution will gather information which is scattered around in several health information systems. An integrated solution will hopefully give a optimal information base which is necessary for proper patient treatment. This in turn will probably contribute to improved continuity of patient care.

This project is carried out in cooperation with CARDIAC AS. CARDIAC is an abbreviation for Computer Aided Research, Development, Instrumentation and Control, and the name reflects activities in instrumentation, research, monitoring and control in the health sector. CARDIAC has developed a middleware platform specially for integration of health informa-

tion systems, IMATIS[1] Platform.

One of CARDIAC's goals is to develop a health information system for electronic observation charts (EOCs) based on IMATIS Platform. The EOCs will look much like the paper-based observation charts which contain a chronological overview of the most essential information about a patient, such as vital signs (i.e. pulse, respiration, blood pressure, etc.), selected patient information, medication, treatment, tests and examinations, ordinations and distributions. Information within a paper-based or electronic observation chart is a subset of the information contained in a traditional patient record or electronic patient record (EPR).

As they are used in hospitals today, paper-based observation charts are actually a collection of information from several different health information systems. Because the observation chart collects information that is necessary in order to make a diagnosis and provide proper patient care, it is probably the most central and mostly used document in a patient record. The paper-based observation chart is also an important collaboration document between health personnel during patient care.

One of the weaknesses with paper-based observation charts today is manual registration of patient information, which may involve manual typing errors. For the improvement of today's routines, automatic information retrieval from medical technical equipment and health information systems, which are important for the patient care, is required. Therefore, medical technical equipment, such as sensors and monitoring systems, and health information systems, such as administrative systems and laboratory systems, should be integrated with the EOC-system. An integration will also support health personnel in performing their work tasks, thereby improving quality of documentation and patient care.

Information security is important because the health information systems contain sensitive information which has to be secured. It is also stated by law that the patient's right to privacy has to be protected when processing personal data. In addition, some regulations about information security are given.

When health information systems are integrated, information security becomes even more important. In addition to securing each system, also the communication lines between them have to be secured. The access control becomes even more important since the access control mechanisms within each of the integrated systems should be enforced in order to allow health personnel to access the information they need for proper patient care, regardless of which system the information is retrieved from. An integration of several health information systems often increases the number of users. The more users, the more important it is to know which actions are performed by whom.

[1] Integrated Module-based Administrative Technology Information System

1.2 Goal

The goal of this thesis is to create a description of a security focused integration architecture for CARDIAC's EOC-system. In order to create this architectural description several subgoals are identified in this thesis.

First of all, it is necessary to gain knowledge of the EOC-system and its domain.

Another subgoal is to study prevailing standards, strategies, laws and regulations which are relevant for CARDIAC's EOC-system. Based on the identified laws and regulations, relevant security concerns should be explored.

An architectural description for integration of health information systems needs to be based on an integration architecture, so a subgoal is to explore different integration architectures which may be used in the implementation of CARDIAC's EOC-system.

The architectural description presented in this thesis will be based on the architectural description frameworks: generic MAFIIA, MAFIIA for IIS, MAFIIA/H and MAFIIA/RBAC. Of these architectural description frameworks, MAFIIA/RBAC is the newest one, and it has never been tested in practice. Therefore, a subgoal of this thesis is also to try out MAFIIA/RBAC on a specific case.

1.3 Limitation of scope

CARDIAC's intension is to implement the EOC-system within the National Health Service, primarily within the Health Region for Central Norway. Since this project is done in cooperation with CARDIAC AS, this thesis will only focus on the above mentioned health region. The Health Region for Central Norway has developed some strategies for integration and architecture of health information systems, which have to be followed by CARDIAC. These strategies will also set some limitations on the work done in this master's thesis project.

Since the EOC-system is in the centre of the integration described in this thesis, and since EOCs are mostly used within somatic hospitals, the scope is limited to only cover somatic hospitals.

The operation of CARDIAC's EOC-system is limited to only running inside internal networks of a hospital, which means that the system cannot be accessed through Internet or other LAN networks.

1.4 Document organization

This report is divided into seven parts.

Part I is an introduction which motivates and presents the goals and scope for this thesis.

Part II is a prestudy which serves as background information for the reader. It contains a description of the National Health Service, CARDIAC's EOC-system, information security within the health domain, laws and regulations related to information security and possible integration architectures.

Part III gives a brief introduction to the methodology used in creating the architectural description of the EOC-system. The MAFIIA architectural description framework and its extensions are the frameworks which are used.

Part IV presents the security focused integration architecture of CARDIAC's EOC-system.

Part V contains discussions of the work done during this thesis, conclusions on the results achieved and suggestions for further work.

Part VI is the bibliography.

Part VII is the appendix.

Part II

Prestudy

Chapter 2

Introduction

When creating an architectural description for an integrated health information system, it is necessary to have knowledge of the domain which the system will operate in. Additionally, it is important to have proper knowledge of the target system itself and the other systems which are supposed to be integrated with the target system. It is also important to be aware of the prevailing laws and regulations which are relevant for the architectural description.

Security is important in all software systems, specially in the health domain. Since this thesis has a large focus on security, there is a need for an identification of relevant security issues. Finally, relevant integration architectures need to be examined.

With all this information in place, a proper basis for the creation of the architectural description is laid.

Chapter 3

Norwegian health service

CARDIAC aims to implement the EOC-system within somatic hospitals in the Norwegian health service. The first implementation will be within the Health Region for Central Norway, and the EOC-system therefore has to satisfy certain standards and strategies introduced by this health region and the Norwegian Government.

This chapter first presents the organization of the National Health Service and the Health Region for Central Norway, before presenting relevant standards and strategies.

3.1 The National Health Service

In 2002 there was a major change in the National Health Service when the Norwegian Hospital Reform [41] was carried out. Before 2002, all hospitals were administrated by the county authorities, but after the reform, the hospitals became health enterprises owned by the Norwegian Government [67]. As shown in Figure 3.1, Norway was divided into five health regions:

- Health Region for Northern Norway
- Health Region for Central Norway
- Health Region for Western Norway
- Health Region for Eastern Norway
- Health Region for Southern Norway

The organization in the National Health Service is shown in Figure 3.2. For each of the five health regions presented above, there is established a regional health enterprise. The five regional health enterprises have responsibility for the specialist health services provided within their geographical

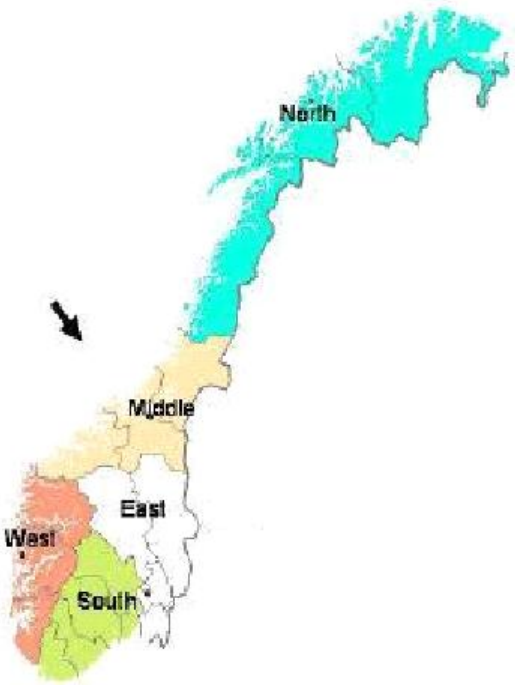


Figure 3.1: Norway's five health regions.

area, and they own all health enterprises within the health region they belong to. Health enterprises run hospitals and other specialist health services, and they may consist of several hospitals or institutions [69].

This report will mainly focus on somatic hospitals because these hospitals cover the observation chart's range of use. A somatic hospital investigates and treats patients with physical diseases or injuries, and it provides surgical and/or medical treatment of patients [49]. More precisely, the main focus of this thesis will be on somatic hospitals within the Health Region for Central Norway because CARDIAC is supposed to deliver their first installation of the EOC-system within this health region. Thus, an arrow points on the Health Region for Central Norway in Figure 3.1. The Health Region for Central Norway consists of six health enterprises and one regional health enterprise. The Central Norway Regional Health Enterprise has a superior responsibility for the specialist health service in Nord-Trøndelag, Sør-Trøndelag and Møre og Romsdal.

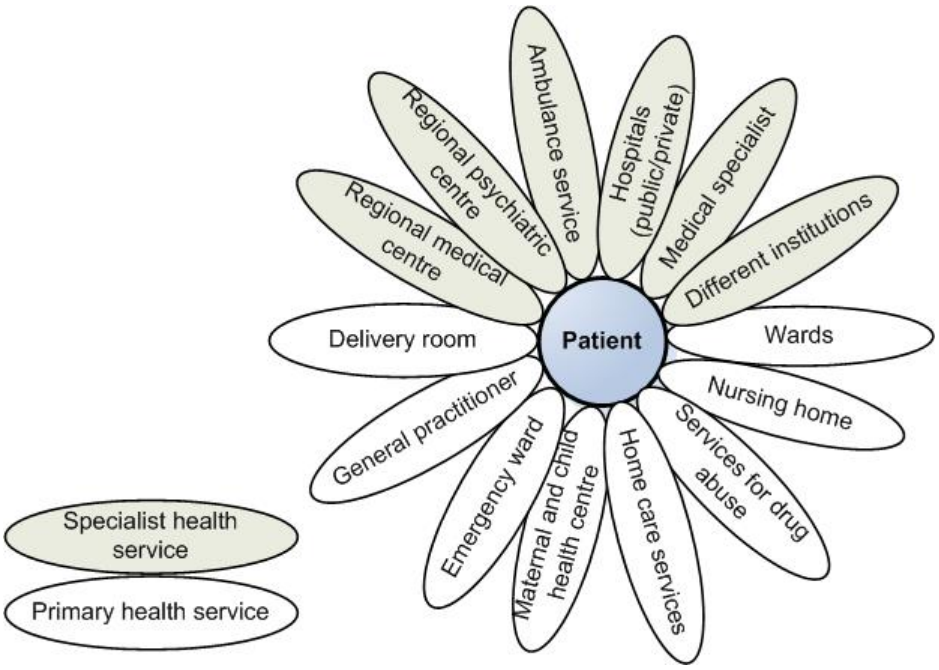


Figure 3.2: Organization in the National Health Service [69].

3.2 Standards and strategies

In the steering document 2004 [35] from the Ministry of Health and Care Services [1] to the regional health enterprises, information and communication technology (ICT) is one of the five national priority areas. ICT is of great importance for the development of the National Health Service because ICT supports work tasks within strategic areas, such as interaction and cooperation within the health enterprises.

When introducing ICT into the health sector, there are certain strategies and standards that have to be followed. The most important ones are described in the following sections.

3.2.1 Te@mwork 2007 Electronic Cooperation in the Health and Social Sector

In March 2004, the national strategy for ICT development in the health and social sector for the period 2004-2007, Te@mwork 2007 [2], was introduced. The strategy provides coherence and direction for national priority areas for electronic interaction in the sector for a three-year period.

The vision of Te@mwork 2007 is that *"patients and clients shall experi-*

[1] Former Ministry of Health
[2] Norwegian: S@mspill 2007

ence continuity of care when using the services" [34]. This vision demands for improving the flow of information within the National Health Service. Improved information flow presupposes working with the following areas: infrastructure, information structure, information security, EPRs, exchange of electronic messages and professional support. Working with these areas, motivates for an integration between health services, information resources and health information systems.

3.2.2 Common ICT strategy

The common ICT strategy is a strategy for the regional health enterprises with proposals for common priority areas and measures. The strategy defines roles and responsibilities of the regional health enterprises in achieving the goals of Te@mwork 2007 [34]. The common ICT strategy is at the core of the coordination of strategies and action plans for the development of ICT in the health sector. The common ICT strategy is a document gathering all contractual obligations the regional health enterprises have according to the national action plans. The document also contains goals set by the regional health enterprises.

The common ICT strategy has six priority areas, some of them are relevant for this thesis:

- Electronic cooperation: Consolidate more extensive use of electronic message exchange.
- Comprehensive and well-defined information base: A common definition of concepts is a basis for all electronic interaction in the health sector.
- Information Security: The establishment of a trade standard for information security and a set of minimum requirements for information security in the sector.

Electronic cooperation and comprehensive and well-defined information base, emphasize the need for integration, while information security itself is an important area within the health sector. To achieve these three priority areas certain goals are set:

- Continuity of patient care attained by information following the patient care and work processes.
- The health enterprises make their work processes more efficient by avoiding that the same work is done twice.
- The information shall flow effectively between different health information systems without mistakes or misunderstandings.

To reach these goals, the many different systems which are involved in patient care have to be integrated.

3.2.3 HEMIT's strategies

HEMIT [3] was established in 2003 as a special IT-unit which gathers all IT-departments in the Health Region for Central Norway. HEMIT now runs all the central servers and all software and infrastructure within this health region.

HEMIT has formulated several IT-strategies. The most relevant for this work are:

- The integration strategy
- The IT architecture strategy

The purpose of the integration strategy is to integrate health services independently of the technological platform they are implemented on. The integration strategy is not yet implemented, and it is meant to be used as a requirement specification for IT-vendors when purchasing or developing new systems or IT-equipment [51]. The integration strategy should be seen in combination with the IT architecture strategy.

The IT architecture strategy is supposed to give a better overview over all information systems and their services. The architecture is based on a service-oriented architecture model for the purpose of Web services.

3.2.4 KITH's EPR-standard

KITH [4] is a centre of competence for the establishment of extensive, efficient and secure implementation and use of ICT in health care.

KITH has developed an EPR-standard [49] as part of the program on *"Standardization and coordination of information- and communication systems in the national health care service"* initialized by the Ministry of Health and Care Services [5]. KITH's EPR-standard aims to form a common platform for all EPRs in the National Health Service according to prevailing laws. The EPR-standard focuses on architecture of patient records, archiving of EPRs, access control, and integration of EPRs with code standards and classification systems.

KITH's standard does not cover all aspects of an EPR, e.g. a limitation is placed on the requirements for the contents in an EPR. Although KITH is working on other standards and requirement specifications within the health sector, this report will only focus on the EPR-standard because it identifies relevant laws and regulations regarding patient records.

[3] Norwegian: Helse Midt-Norge IT
[4] Norwegian: Kompetansesenter for IT i helse- og sosialsektoren AS
[5] Former Ministry of Health

3.3 Integration of health information systems

The above mentioned ICT strategies and standards constitute the foundation for a move towards integration of health information systems. An integration of health information systems provides for:

- Better quality of care, which implies better information quality. For example, a user does not have to type in the same information more than once because it is easy to make mistakes when the same information has to be typed in several times.
- Continuity of care, which implies that the needed patient information is available for the right person at the right time.

In addition to better quality of care and continuity of care, integration of health information systems hopefully results in reduced costs.

3.4 Summary

This chapter has presented relevant standards and strategies. The recommendations and suggestions in these standards and strategies will be ensured in the architectural description created during this project. The architectural description is for an EOC-system, and EOCs will therefore be described in the next chapter.

Chapter 4

Patient observation chart

A patient record includes necessary and relevant information for the patient care. This information was originally divided into ten different groups. Because this project is done in collaboration with CARDIAC, and CARDIAC aims to develop its own EOC-system, this chapter will only focus on group F, which contains all kinds of observation charts.

A patient observation chart is an important tool for organizing and recording patient care activities in the hospitals. The paper-based observation chart contains essential information about the patient, and it is used to quickly get an overview of the patient treatment.

An introduction to patient records is given in this chapter, before patient observation charts and CARDIAC's EOC-system are presented in detail.

4.1 Patient records

Patient records have been used in Norway since the 19th century for the purpose of documentation of patient care. Since patient records were introduced, they have gone through dramatic changes. From being hand-written and organized in paper archives, they have now become electronic with the possibility to automatically receive and retrieve patient information necessary for the patient care. A patient record is a working tool for health personnel and should contribute to a proper medical treatment. All medical treatment for a patient should be documented in a patient record to ensure the quality and continuity of patient care.

In Norway, the information in a paper-based patient record is classified in groups from 'A' to 'J' as seen in Table 4.1. The table only shows a simplified overview of the information in a paper-based patient record. The contents of the groups are detailed in Appendix A [11].

Nowadays, EPRs are dominating the Norwegian health service. An EPR is a patient record in which information is stored electronically and can be retrieved and reused by means of suitable software [36]. An EPR should

Group	Name	Content
A	Summaries	Contains patient's biographical data and details about when, where and why the patient was hospitalized. It also contains a nursing summary and a document written in cooperation with the patient when he is discharged from the hospital.
B	Doctors' record	Contains all internal doctors' notes which are not written in the observation chart. It also contains answers to internal referrals.
C	Laboratory results	Contains all laboratory and test results from examinations of blood and other fluids, pus and tissue.
D	Organ function	Contains all examinations in which the patient has to be present, except picture diagnostics from group E.
E	Picture diagnostics	X-ray, Computed Tomography (CT), Magnetic Resonance (MR) and ultrasound examinations.
F	Observations and treatment	All kinds of observation charts are contained in this group.
G	Nursing documentation	All documentation about the nursing of the patient, together with observations, evaluations, decisions and actions.
H	Documentation from other professionals	Contains replies and reports from other professional groups who have done separate therapies or treatments on the patient, e.g. a childbirth is leaded by a midwife and she is a part of some "other professional group".
I	External correspondence	Contains external referrals, evaluations of external referrals, hospitalization documents and other correspondence.
J	Attestation/messages/ statements	Contains copies of ALL messages which are send/received regarding the patient.

Table 4.1: Overview of the contents of a paper-based patient record.

at least include information corresponding to the information archived in a paper-based patient record.

Group F, observations and treatment, is in Table 4.1 listed as a part of the paper-based patient record and it should therefore also be a part of the EPR. Group F usually consists of patient observation charts which are produced during a patient's hospital stay.

This master's thesis project is done in cooperation with CARDIAC, and CARDIAC aims to develop an EOC-system. Group F in the paper-based patient record is therefore of particular interest for this work. Both paper-based and electronic observation charts are described next.

4.2 Paper-based observation chart

The paper-based observation chart is a binder that contains a chronological overview of the most essential information about a patient, such as vital signs (i.e. pulse, respiration, blood pressure, etc.), selected patient information, medication, treatment, tests and examinations, ordinations and distributions [21]. In short, the observation chart is used to quickly get an overview of the patient treatment, and a good overview helps in improving the quality of care.

There are many different types of paper-based observation charts and each hospital is free to use its own types. Within each hospital, observation charts are specially designed for the ward they are used in, which means that there can be several different types of observation charts within one single hospital [11].

A standard observation chart is for example quite different from the observation chart which is used in an intensive care unit. The chart is, of course, much more detailed in an intensive care unit. In an intensive care unit for new-born three different observation charts are used: main observation chart, medication chart and a special observation chart for observation of nutrition consumption and a few other variables.

Figure 4.1 shows a detailed picture of a typical paper-based observation chart used in an intensive care unit for new-born.

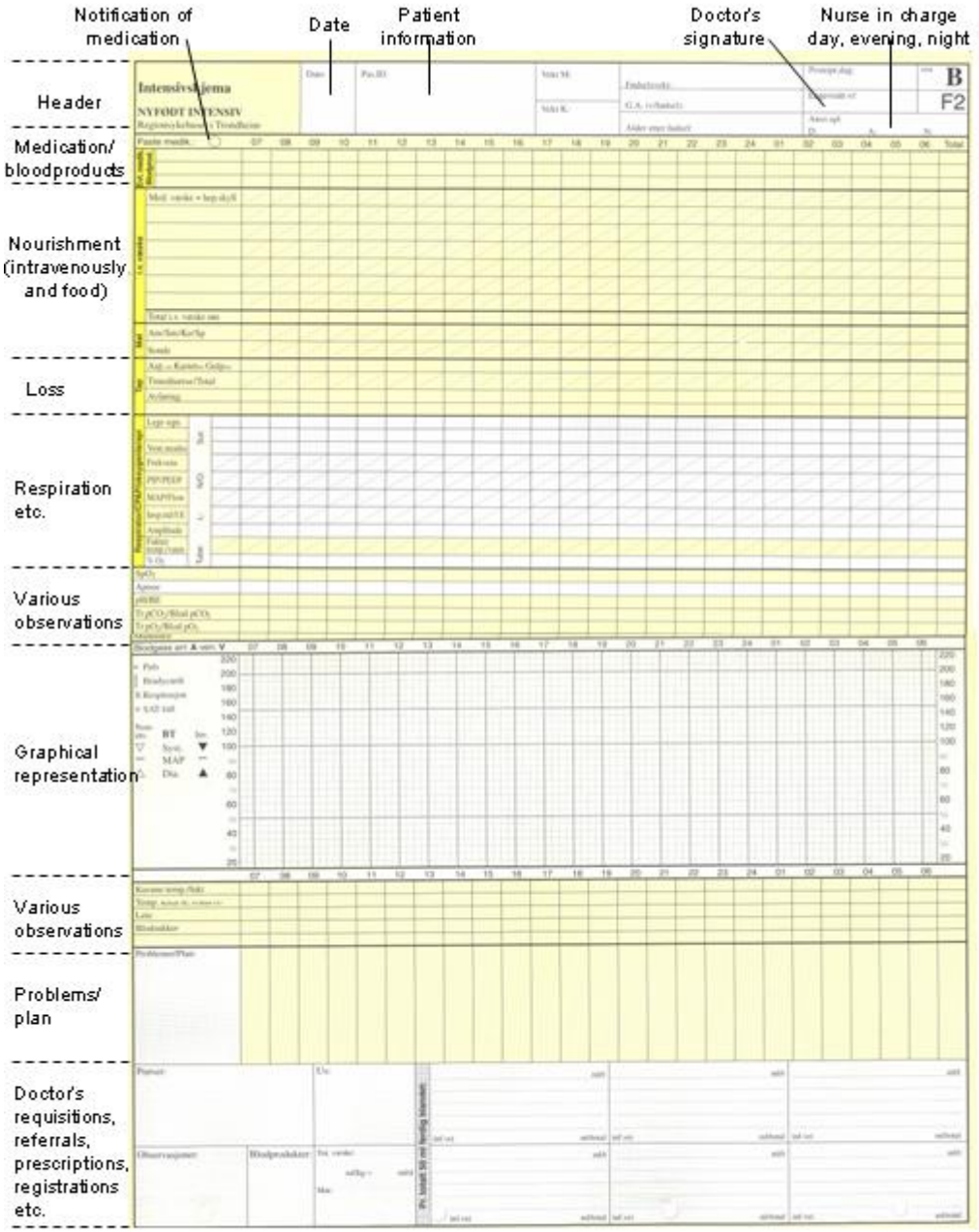


Figure 4.1: Paper-based observation chart.

4.3 Electronic Observation Charts (EOCs)

As mentioned previously, an EPR has to include all information that is archived in a paper-based patient record, meaning that it also has to contain EOCs.

EOCs replace the traditional paper-based observation charts used within Norwegian hospitals. Since the EOCs are stored electronically, information in them can be used for other purposes, in addition to documenting the patient care. For example, the information may be used for statistical and research purposes or for legal matters.

An EOC can also give a bird's-eye view of the patient care, and it should be able to tell [33]:

- *How much information there is about the patient.* An EOC may provide functionality of a zoom in function which makes it easier to see the total amount of information before going into details of it.
- *Which periods of the patient's life he has been medically treated.* An EOC may use the patient's medical history to emphasize certain periods of a patient's life on a time scale.
- *Who has treated the patient and what tests have been done.* Since all medical actions must be signed by someone who has made the decision, it is easy for an EOC to present an overview of who has treated the patient and what tests have been done.
- *What plans are made for future treatment of the patient.* It should be possible to get an overview of ordered laboratory tests, medical appointments, etc. in the EOC.

4.4 CARDIAC's EOC-system

CARDIAC aims to develop an EOC-system that will be integrated with other health information systems. Since the paper-based observation chart has been such an important and useful part of the work processes for the health personnel, CARDIAC has chosen to base their EOC-system on the structure of the paper-based observation chart. As shown in Figure 4.2, CARDIAC wants the EOC-system to be divided into two parts:

- Navigation caremap
- Patient chart forms

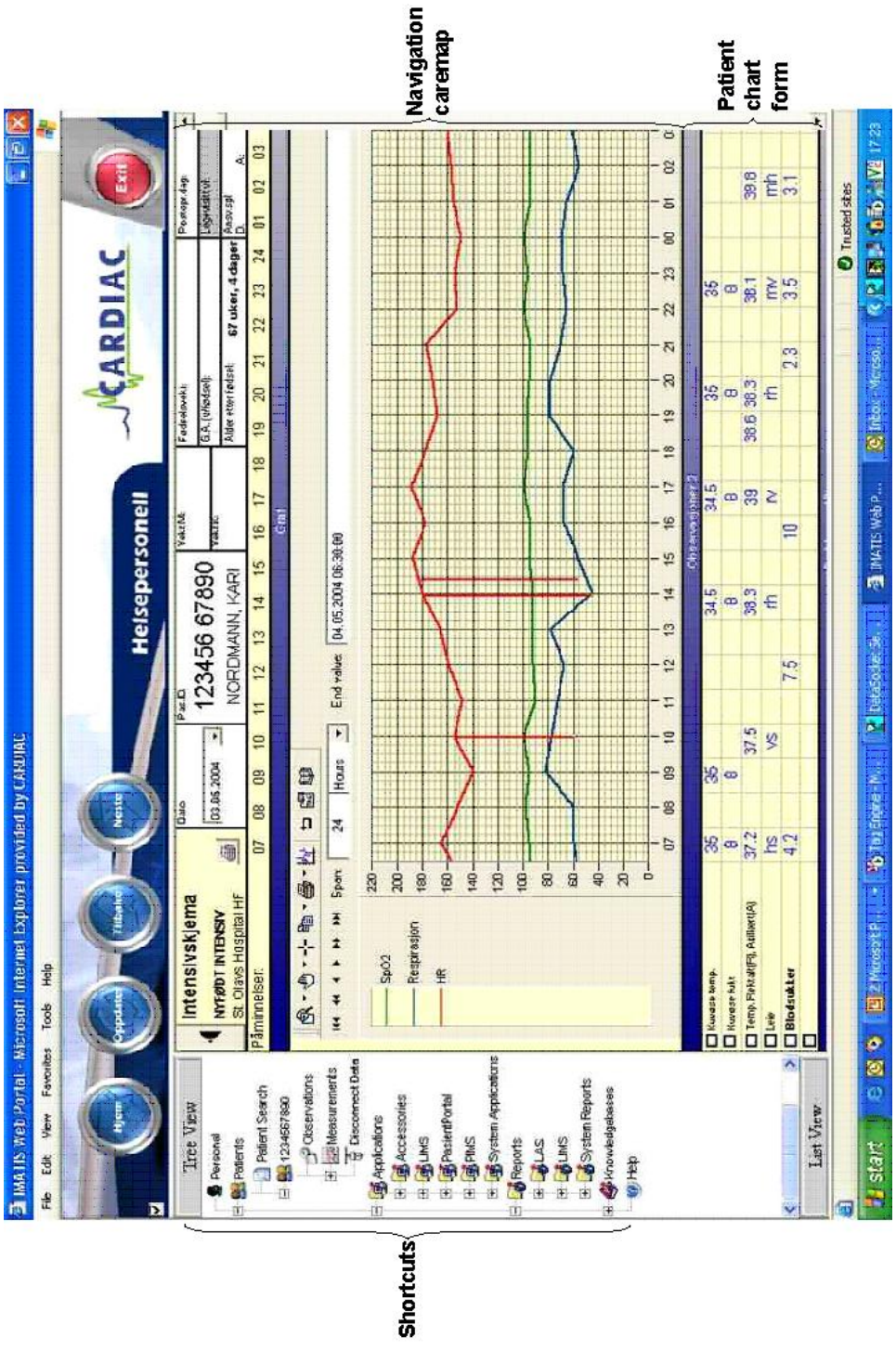


Figure 4.2: CARDIAC's EOC-system consists of a navigation caremap and a patient chart form, in addition to shortcuts to other health information systems.

The navigation caremap is intended to be the main navigation window in the EOC-system. It should be a "read-only" version of the patient observation chart in which the time scale can be changed to show different amounts of information; either just a few days or several years of treatment. The navigation caremap will be divided into different categories which indicate vital signs, medication/prescription, test/result, picture diagnostics, etc. The time scale will give a categorical overview of historical documentations and occurrences, the most important, present parameters that describe the patient's condition and planned actions for the patient care. The patient information presented in the time scale will be retrieved from other health information systems, such as administrative systems, picture archiving systems and laboratory systems. This means that these health information systems should be integrated with the EOC-system. With that, users will easily get an overview of the patient's condition and its development, including observations, occurrences, treatment, documentation and test results.

In the every day work with patients, EOCs will be used. These are electronic versions of the paper-based observation charts; they look alike and have the same structure as the paper-based ones, and they allow input. The EOCs will be used to get an overview of the patient condition during one day or a week. They may be configured so that the information contained and presented in the forms is adapted to the work situation. In other words, EOCs may be specifically adapted to the wards in the hospital.

4.5 Integration of CARDIAC's EOC-system

As shown in Figure 4.2, CARDIAC wants to base their EOC-system on the paper-based observation chart. In addition to allowing health personnel to register observations continuously, the observation chart usually needs information from several other health information systems. This is shown in Figure 4.3. An EOC-system should provide the opportunity for the health personnel to register some observations manually, while others should be retrieved automatically from other health information systems or medical technical equipment.

CARDIAC's idea is that the navigation caremap shall give direct access to other health information systems. For example, it should be possible to click on certain icons for requisition or booking of blood tests, picture diagnostics, electrocardiogram (ECG) or other examinations. Response functionality for these areas should also be available in the navigation caremap. For example, the EOC-system may contain a note which says that a blood test result is finished. By clicking the note or the icon beside the note, the user will be able to see the results of the test immediately.

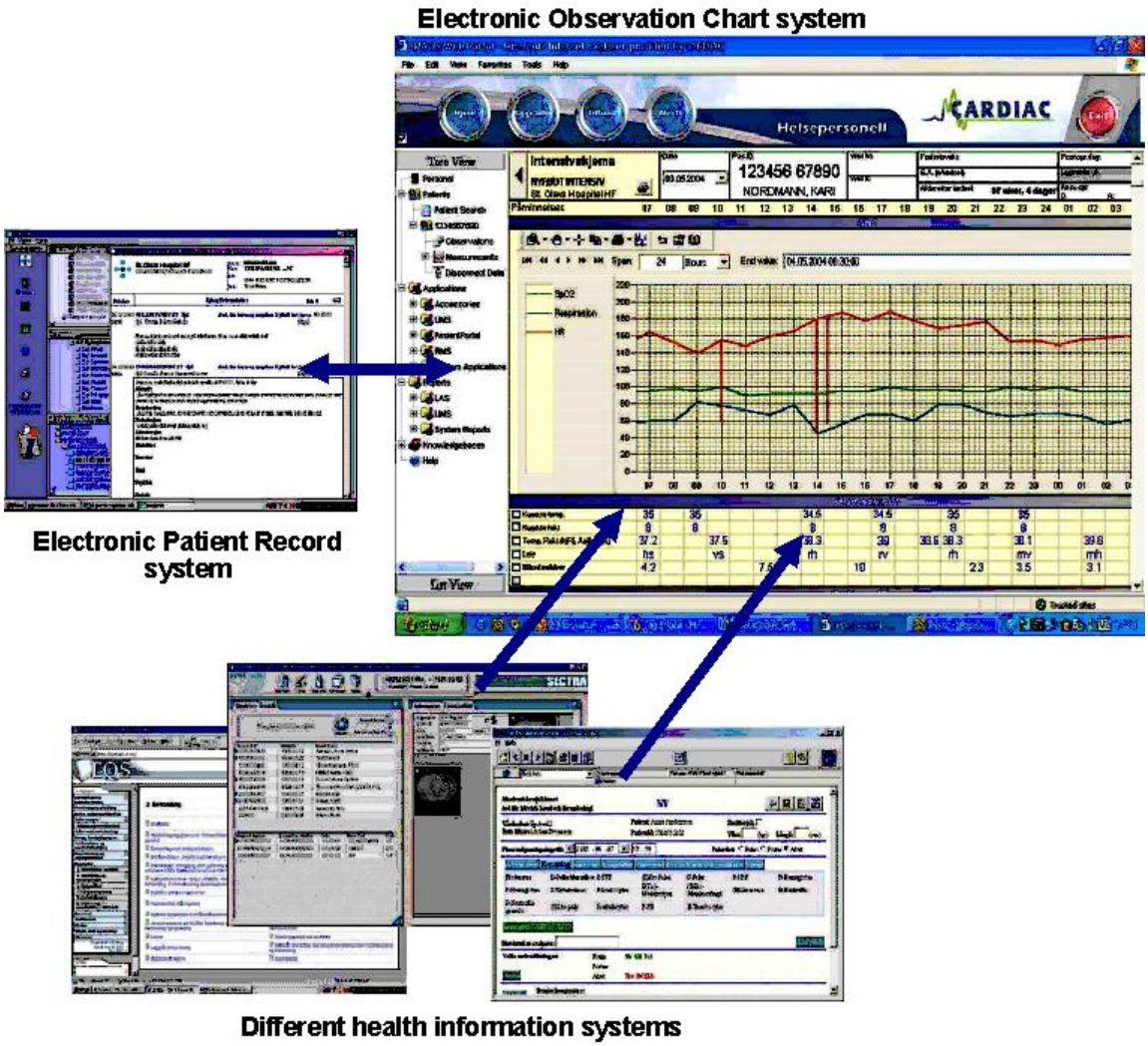


Figure 4.3: EOC in integration with other health information systems.

CARDIAC has identified some health information systems which are particularly relevant for the EOC-system. These systems all need to be integrated in order to improve the quality of the information in the EOC-system and improve the efficiency of patient care.

Relevant health information systems for integration with the EOC-system, are presented in the subsequent sections.

4.5.1 EPR-system

The information in EPRs resides in several different health information systems. Throughout this thesis, the EPR-system is referred to as one of these health information systems. It is referred to as a system only containing

information about CAVE [1], doctor's documentation and nursing documentation.

An integration of the EPR-system with the EOC-system is necessary for proper patient care.

4.5.2 PAS

Patient Administrative System (PAS) handles administrative information about a patient and the patient's hospital stay. An integration with PAS is needed because PAS provides information about patient's biographical data, diagnosis and patient care incidences [2].

4.5.3 LIS

Laboratory Information System (LIS) handles laboratory tests and stores test results. CARDIAC wants to increase the efficiency of patient treatment by providing direct access to test results in the EOC-system. In order to achieve this, integration with LIS is also necessary.

4.5.4 PACS/RIS

Picture Archive and Communication System/Roentgen Information System (PACS/RIS) is a system for storing pictures such as X-rays, Magnetic Resonance (MR), Computed Radiography (CR) and other. In the same way as with LIS, an integration with PACS/RIS is necessary in order to improve the efficiency of patient treatment.

4.5.5 RoS

An integration with the Requisition and Response system (RoS [3]) is necessary in order to allow users of the EOC-system to write requisitions and to view the response which is given when the requisition has been processed.

Throughout this thesis, it is assumed that LIS and PACS/RIS are integrated with RoS. Hence, the EOC-system only needs to communicate with RoS in order to get pictures and information about laboratory tests.

4.5.6 Medication

When integrated with the Medication system, the EOC-system will have the possibility of allowing users to prescribe medications in the system. This may result in faster and more efficient processing of medication.

[1] Latin: avoid. Information about allergies or hypersensitivity towards certain types of medication that a patient may have.

[2] patient care incidences: points in time when the patient has been hospitalized

[3] Norwegian: Rekvisisjon og Svar

At this point, no medication system has been implemented yet, but medication systems are still taken into account when creating the architectural description of the EOC-system.

4.5.7 EQS

Extend Quality System (EQS) contains quality procedures for performing different tasks in the patient care. An integration with the EOC-system would improve the efficiency and quality of patient care because the procedures may be linked to work processes and work tasks. For example, every time a nurse is supposed to feed the patient intravenously, there will be procedures for how she is supposed to attach the needle to the patient.

4.5.8 MTU

Medical technical equipment (MTU [4]) embraces all equipment which is used in monitoring a patient's condition, i.e. instruments such as electrocardiogram (ECG) and incubators. An integration between these instruments and the EOC-system is necessary in order to avoid typing mistakes when health personnel reads information from the instruments and registers it in the EOC. CARDIAC has already developed a system for automatic data acquisition from these instruments, IMATIS Medical Data Acquisition System. IMATIS Medical Data Acquisition System is a system specially designed for automation, standardization and quality assurance of data from medical technical equipment. The system is able to present data in real-time or historically and even for transmission to other systems, such as the EOC-system.

4.6 Summary

This chapter has presented CARDIAC's EOC-system and health information systems relevant for integration with the EOC-system. The most relevant of the identified systems are the EPR-system, PAS, RoS, Medication system, EQS and MTU. These will be chosen for integration with the EOC-system in the architectural description.

The following chapter will focus on legal requirements for information security which are relevant for the National Health Service in general and for CARDIAC's EOC-system specifically. Several important security mechanisms are also described in the following chapter.

[4] Norwegian: Medisinsk Teknisk Utstyr

Chapter 5

Information security in the health sector

This chapter presents important legal requirements which are relevant for implementation of CARDIAC's EOC-system. On the basis of these requirements, certain security principles are identified. These security principles must be ensured when implementing information systems in the health sector in general, specially in connection with the integration of health information systems.

Throughout this thesis, a *security principle* is defined to be a security goal which is achieved by a set of security mechanisms.

Security mechanisms are the means which are used to satisfy one or more security principles.

In this chapter a brief presentation of both national and general security principles is given, before several security mechanisms for the achievement of the security principles are defined. The security mechanisms are then discussed in a health context.

5.1 Laws and regulations regarding observation charts

There is no legislation specifically for EOCs in the Norwegian law today. Still, EOC is a part of an EPR and a development of an EOC-system must therefore follow all laws and regulations regarding EPRs in general. In the following, a selection of relevant laws and regulations is presented:

Personal Data Act[1] [40] is supposed to protect natural persons from violation of their right to privacy through the processing of personal data. The act applies to the processing of personal data wholly or partly by automatic means.

Personal Data Regulations[2] [39] are determined under the provision

[1] Norwegian: Personopplysningsloven
[2] Norwegian: Personopplysningsforskriften

of the Personal Data Act, and it gives regulations about information security when personal data is processed.

The Personal Health Data Filling System Act[3] [37] deals with the employment of general decisions about patient records in the Personal Data Act and contains a number of rules directly relevant for EPRs.

The Health Personnel Act[4] [36] deals with health personnel's duties and responsibility in connection with their work. This includes relations attached to client confidentiality, the right to information, the duty to report and the documentation requirement.

Patient Record Regulations[5] [43] are determined under the provisions of i.a. the Health Personnel Act, and they give further rules about the contents of a patient record, the work with patient records and access to the information in the patient record.

The Archive Act[6] [42] says that all public sectors (government, county authority and municipal authority institutions) have a legal obligation to have an archive. Therefore, this act also applies to patient records and other information within public health enterprises.

5.2 Legal requirements

Combined together, the above mentioned laws and regulations make certain demands for management of patient records, which in turn affect CARDIAC's EOC-system. The legal requirements for patient records are described in the following sections.

5.2.1 Duty to keep patient records

According to the Health Personnel Act and the Patient Record Regulations, all Norwegian health enterprises, where health services are provided, are obliged to have a patient record system. One patient record should be established for each patient, and each patient should only have one record, even though several health enterprises contribute to the patient care. The patient care should be documented in the patient record immediately after the health service is performed. All patient record documentation should be dated and signed.

It is possible to employ digital signatures or different kinds of approval for EPRs, but there are no special requirements for approval of EPRs.

[3] Norwegian: Helseregisterloven
[4] Norwegian: Helsepersonelloven
[5] Norwegian: Pasientjournalforskriften
[6] Norwegian: Arkivloven

5.2.2 Responsibility

For each established patient record, health institutions should designate one person with superior responsibility for it, cf. the Health Personnel Act §39. The superior responsibility for the individual patient record includes making decisions relating to what information is to be entered into the patient record.

The information entered into the patient record is referred to as personal health data, and this kind of information has to be handled legally. The data controller is responsible for ensuring sufficiently secured health data, cf. the Personal Health Data Filing System Act §16: *"The data controller and the data processor shall by means of planned, systematic measures, ensure satisfactory data security with regard to confidentiality, integrity, quality and accessibility in connection with the processing of personal health data..."*.

According to this act, a data controller is the person who determines the purpose of the processing of personal health data and which means are to be used. The data controller also has the responsibility for the exchange of health data, cf. the Personal Health Data Filing System Act §16.

5.2.3 Processing of personal health data

All processing of personal health data shall have an explicitly stated purpose, cf. the Personal Health Data Filing System Act §11. The data controller shall ensure that the personal health data that are processed are relevant to and necessary for the stated purpose.

5.2.4 Access to personal health data

Access to personal health data shall, according to the Personal Health Data Filing System Act §13, only be granted to the data controller, the data processor or to persons working under the instructions of the controller or the processor. In addition, access should only be granted if this is necessary for the work of the person concerned and in accordance with the rules that apply regarding the duty of secrecy.

5.2.5 Correction and deletion

Rules for correction and deletion are given in the Health Personnel Act §§42-44.

Wrongful, deficient or improper information or comments in patient records should be corrected. Correction shall be carried out through re-entering the information of the patient records, or by adding a dated correction in the records. Corrections shall not be made by deleting information or comments.

Deletion of information in patient records should only be done if it can be done without implications to public interest, if it is not in accordance with the Archives Act sections 9 or 18, and if the information is wrong or misleading and felt to be a burden for the person they relate to or the information clearly is not necessary in order to provide health care for the patient.

Correction and deletion are only done upon demand from the person whom the information relates to, or of the health personnel's own accord.

5.2.6 Exchanging personal health data

Health personnel is according to the Health Personnel Act obliged to give necessary patient care. Also, health personnel is, if necessary, obliged to exchange personal health data, refer to §45 in the Health Personnel Act: *"Unless the patient objects thereto, health personnel as mentioned in section 39 may give the patient record or information therein to others who provide health care pursuant to this Act when this is necessary in order to provide health care in a responsible manner"*.

The Health Personnel Act does not specify how the information shall be exchanged, but in the comments to the regulations related to the patient record in the Patient Record Regulations §10 electronic transmission is mentioned. Information in the patient record can be transmitted electronically if the system in use has security solutions corresponding to the requirements in the Personal Health Data Filing System Act, Personal Data Act and Personal Data Regulations.

5.2.7 Auditing of access

Auditing of access is also an important matter, refer to §45 in the Health Personnel Act: *"..It shall be evident from the patient record that other health personnel has been given access to the patient records pursuant to the first sentence."*. Because it shall be evident from the patient record that other health personnel have been given access to the record, all EPRs should register the one that gave access and the one that has been given access.

Also the Personal Data Regulations §2 require auditing of access to EPRs: *"..Recording authorized and unauthorized use of information systems, must be stored in at least 3 months..."*.

5.3 Security principles

Chapter 2 in the Personal Data Regulations deals with information security in regard to information systems storing personal data, in this context personal health data. This section presents some security principles which are mentioned in these regulations and which will be taken into account

when including information security into the architectural description for CARDIAC's EOC-system.

5.3.1 Confidentiality

Confidentiality refers to the need to keep information secure and private [6]. It means ensuring that only authorized parties are able to understand the data. Unauthorized parties may be aware that there is some data, they can copy the data, but they should not be able to understand it [18].

"A breach of confidentiality is a disclosure to a third party [....].Disclosure can be oral or written, by telephone or fax, or electronically, for example, via e-mail or health information networks. The medium is irrelevant, although special security requirements may apply to the electronic transfer of information" [60].

According to §2-11 of the Personal Data Regulations there should be measures for the protection of confidentiality of personal health data: *"Measures shall be taken to prevent unauthorized access to personal data where confidentiality is necessary. The security measures shall also prevent unauthorized access to other data of significance for data security".*

This paragraph also mentions electronic transmission and storing of personal data: *"Personal data that are transferred electronically by means of a transfer medium that is beyond the physical control of the data controller shall be encrypted or protected in another way when confidentiality is necessary. As regards storage media that contain personal data where confidentiality is necessary, the need to protect confidentiality shall be shown by means of marking or in another way. If the storage medium is no longer used for the processing of such data, the data shall be erased from the medium".*

5.3.2 Availability

Availability applies to the flow of data and the accessibility of the system. It should be ensured that complete, updated, correct and relevant information is available for those who have a legitimate need for it. For example, this means that an attack that makes a system crash and the information unavailable should be avoided.

Within the health service, availability is sometimes really important. For example, in situations where patients' lives are in danger, it may be necessary to circumvent the access rights, giving unauthorized health personnel the availability to access (parts of) the patient record. This availability to circumvent the access rights in life-threatening situations is called *blue light access.*

It is important that auditing is done when blue light access is given. It is necessary to audit the reason for the blue light access, the operations that are going to be performed, and the identity of the person that is performing these

operations. The owner of the record should also be informed immediately. Auditing is described in Section 5.4.3.

§2-12 in the Personal Data Regulations is about securing accessibility, and it says that measures shall be taken to secure access to personal data where accessibility is necessary. In addition, preparations should be made for alternative processing in the event of the information system being unavailable for normal use. Information that is necessary to restore normal use shall be copied.

5.3.3 Integrity

In the context of computer security, integrity is defined as the prevention of unauthorized writing. No user of the system, even if authorized, should be permitted to modify data items in such a way that information is lost or corrupted [5].

Another integrity perspective is data integrity, meaning that the data stored in an information system is the same as the source documents and that it has not been exposed to accidental or malicious alternation or destruction.

In the context of communication security, integrity is defined as the detection and correction of modification, insertion, deletion, or replay of transmitted data including intentional manipulations and random transmission errors [10].

Measures to prevent unauthorized changes in personal data where integrity is necessary are imposed in §2-13 of the Personal Data Regulations. In addition, security measures should be taken to prevent unauthorized changes in other data of significance for data security. Security measures against malicious software should according to this paragraph also be taken.

Integrity in general is closely connected to another security principle, namely non-repudiation.

5.3.4 Non-repudiation

There are two types of non-repudiation. The first one implies that the sender is able to prove that the intended recipient actually has received a sent message. The second type implies that the recipient is able to prove that the alleged sender actually has sent the message [18]. Non-repudiation can also be explained as a method of proving either that a user has performed an action, or that the user has sent or received some information at a particular time.

Absolute non-repudiation is quite difficult to achieve because a comprehensive non-repudiation plan usually requires authentication, authorization, data integrity, and auditing.

Non-repudiation is not stated by any of the laws or regulations listed in this chapter. Still, it is described because it is important when it comes to

juridical information, e.g. during handling of complaints from patients due to medical malpractice suit.

5.4 Security mechanisms

On the basis of the legal requirements described in Section 5.2 and the security principles presented in Section 5.3, several security mechanisms are identified. These security mechanisms are described in the following sections.

5.4.1 Digital signing

As described in Section 5.2.1, all patient documentation should be dated and signed. There are no specific requirements for signing patient documentation, but it is possible to employ digital signatures for this purpose. According to the book *Computer Security* [10], a digital signature is *"a construct that authenticates both the origin and contents of a message in a manner that is provable to a disinterested third party"*. This means that digital signatures can be used to verify data integrity and to provide non-repudiation, respectively described in Section 5.3.3 and Section 5.3.4.

A digital signature confirms that a particular person has written and/or approved the document, and the receiver of the document is able to prove that this person really signed it and that the document has not been altered since the signing. In addition, digital signing may also be used for the purpose of authentication which is described below in Section 5.4.4.

Typically, a public key infrastructure (PKI) solution is used for digital signatures. PKI is a collective term for technology providing unique digital identities across networks, where the digital identity is used for authentication, digital signing of information and encryption of communication [65].

Within the health sector, digital signing is relevant for different purposes. During patient care, it is important to know which actions are performed by whom. In an EOC, all insertions and corrections should be digitally signed. Digital signatures are therefore essential for the traceability of registrations in the EOC-system. In addition, digital signing of prescriptions, medical certificates, requisitions and referrals is highly desirable in an EOC-system.

When it comes to integrated health information systems, digital signatures are even more important because integrated health information systems are more complex than single ones. Traceability of registrations and alterations is therefore essential, and preserving the data integrity within the integration is crucial.

5.4.2 Secure communication

In Section 5.2.6, it is stated that information in a patient record can be transmitted electronically if the system in use has a satisfactory security

solution. Secure transmission of information and mutual authentication of the entities participating in the communication are therefore important in health information systems.

By far the most important automated tool providing secure communication is encryption. Both *asymmetric* (also referred to as public-key) and *symmetric* (also referred to as conventional) encryption is in common use. With asymmetric encryption, different keys are used for encryption and decryption. The encryption key can be made public, while the decryption key has to remain private. With symmetric encryption, a *secret key* is used for both encryption and decryption [10].

For example, to begin an authenticated communication between two entities, respectively server A and server B, server A sends a request encrypted with server B's public key to server B. B decrypts the request with its private key and replies with a message encrypted with A's public key. Server A and server B can establish a private channel through a secret key algorithm, where server A chooses an encryption key and sends it to server B in the authentication message. Once the authentication is complete, all communication under the agreed secret key can be assumed to be secure.

The example above illustrates that symmetric encryption is used in the main part of the communication, while asymmetric encryption is commonly used only to exchange a secret key. Since asymmetric encryption and decryption requires more system resources than symmetric encryption, and since it is much easier to distribute keys with asymmetric encryption than with symmetric encryption, asymmetric encryption is used for exchanging secret keys, and symmetric encryption is used for the rest of the communication.

This example also illustrates that secure communication ensures confidentiality and integrity. It guarantees that transmitted information is neither accessed nor altered by unauthorized users.

Because an EOC contains a gathering of information needed to make a diagnosis and for proper patient care, CARDIAC's EOC-system will retrieve information from other health information systems such as the EPR-system, PAS and RoS. Roughly speaking all health information systems contain sensitive information, so secure communication and transmission of information between these systems are needed.

5.4.3 Auditing

As pointed out in Section 5.2.7, auditing is mandatory when dealing with patient records. All accesses and modifications should be traceable in an audit log.

Auditing is an important security functionality for traceability and detection of misuse and intrusions. Auditing is a posteriori technique for detection of security violations or other suspicious events, with the purpose of ensuring traceability within the system. This should include tracing access

to sensitive information stored in the information systems as well as access to the information systems themselves. Each access and/or access attempt should be recorded in an audit log for later analysis [10].

The audit log should include information about the one accessing information, services or resources, what he is allowed to access, the accessed information, services or resources, and the time they are accessed. For this purpose, it is desirable that each audit record contains the following fields [14]:

- **Subject:** The one that initiates an action. A subject is typically a user, but it might also be a process acting on behalf of users. The subject's identity should be recorded in this field.
- **Action:** The operation performed by the subject on or with an object. Operations might be login, read, execute, perform I/O, etc.
- **Object:** The resource that subjects perform their operations on.
- **Time-stamp:** The exact time the action took place.

In addition to the purpose of traceability and detection of misuse, the information in an audit log may also be used to ensure non-repudiation, since all actions are audited together with user information.

Within the health sector, it should also be possible to investigate misuse and intrusions without violations of the patients' right to privacy. Therefore, it is important to keep the audit log in a secure place, in addition to only allowing privileged users to access it.

When it comes to EOCs, all new registrations and changes of information should be audited. As mentioned earlier, integrated systems are more complex than single ones, and traceability of more users and more actions has to be handled.

5.4.4 Access control

According to the legal requirements defined in Section 5.2.4, access should only be granted if this is necessary for the work of the health personnel concerned.

Access is usually seen as the ability to interact with a computer resource. Access control is used to explicitly enable or disable the above mentioned ability in some way. According to the book *Role-Based Access Control* [6], access control is the most common and the most used security mechanism today. Access control can be divided into two main parts: authentication and authorization.

Authentication is concerned with verifying that the initiator of a request has the identity which he claims to have. The process of authentication

usually goes like this; a user identifies himself to the system, then he authenticates his identity by providing a second piece of information that only he can know, produce or provide.

Authorization determines whether the given identity is allowed to access a resource or not. Authorization is an important part of every security policy, i.e. a set of rules that states which actions are permitted or prohibited. The purpose of authorization is to protect information, services, resources, etc. against unauthorized use [10].

Access control ensures both availability and integrity of information. It is therefore crucial in all health information systems where sensitive information about a patient is processed and/or stored. This kind of information should be accessed by authorized users only. It is critical that health personnel gains access to the right information at the right time.

Several Norwegian health instances have given priority to the improvement of access control in health information systems. For example, access control is an important part of KITH's EPR-standard, where the objective is that the EPR (based on the standard) has an access control mechanism where the legislation's intentions are followed with as high usability and performance as possible. Access control is also mentioned in HEMIT's IT-architecture strategy, where single sign-on is required. Single sign-on means that the user only has to authenticate himself once per session regardless of which system he may access.

In the health service a combination of different types of access control is often used. Two of the mostly used access control types are described below.

Role-Based Access Control

Role-based access control (RBAC), is a special type of access control where all users of a system are assigned roles and access decisions are based on these roles. This is shown in Figure 5.1.

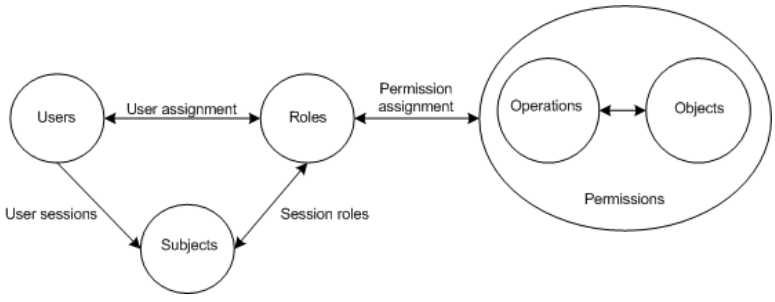


Figure 5.1: RBAC relationships [6].

In RBAC, users are granted membership in a role according to their competence and responsibility in the organization. The operations that the

users are permitted to perform are defined according to the role they belong to. Thus, a role is actually a set of permissions which are given to a user through the role [6].

RBAC supports, among others, the principle of least privilege. The principle of least privilege requires that users are given no more privileges than necessary to perform their tasks. In RBAC, this is achieved by a configuration that only allows users to perform an action if they are assigned a role which that action requires.

There are two types of roles which are used in RBAC:

- **Static roles** are used where a user is registered with one role in the system and keeps this role as long as he is registered in the system. Role assignment is done by a system administrator.

- **Dynamic roles** are used in cases where one person can change roles quite often. Role assignment in such cases is done automatically by the system. An example can be rosters used for assigning roles to users like *Active shift worker* or *Shift supervisor*.

Within the health domain dynamic roles are the most relevant.

The motivation for RBAC in health information systems is that health personnel is dependent on communication with other people for doing their work tasks. This dependency on the other people is based on their role, not identity. The identity is irrelevant as long as the person concerned has the knowledge associated with the particular role.

Context-based

In context-based access control, access decisions are based upon the user's context. The context could be time, the location of the user, people or technical devices the user is close to, communication channel or strength of user authentication. For example, if the context is the location of the user, access rights are dependent on the network address the user operates from.

Compared to RBAC, context-based access control is less specific, and it is more like a property than an access control mechanism.

Nevertheless, integrated health information systems should be context-sensitive with a combination of role-based and context-based access control, making access control even stronger and more dynamic. Health personnel should have roles based on superior profession combined with ward belonging, rosters and patient relations for the purpose of dynamic roles.

Users' ward belonging restricts access to information only belonging to patients associated with the particular ward. Health personnel may have duties on different wards and hospitals, and the ward belonging is dependent on this. For example, a nurse may have the responsibility as a head nurse on one duty and the responsibility as a regular nurse on another duty.

Rosters should constitute a part of the role definition because they include access with time constraints. A doctor on duty has the right to access particular patient information, while a doctor not on duty does not have the right to access patient information at all.

Roles based on patient relations take different patient relations, such as doctor-patient, specialist-patient or surgeon-patient, into account, where the doctor, specialist and surgeon may be the same person.

The access control mechanism within one single health information system is probably different from the mechanism within an integrated solution because an integrated solution consists of several health information systems with already existing access control mechanisms. An integrated solution might demand new principles, models and methods for an integrated access control mechanism. Also, it is most likely that an integrated solution has more users than a single system, and the access control mechanism within an integration solution is therefore more complex than the one within a single system.

5.5 Summary

This chapter has presented important legal requirements and security mechanisms which must be kept in mind when creating the architectural description for CARDIAC's EOC-system. In addition to being security focused, the architectural description for CARDIAC's EOC-system should also focus on integration. Different integration architectures are described in the following chapter.

Chapter 6

Integration architectures

Since CARDIAC's EOC-system will be integrated with other health information systems, an integration architecture is needed for the creation of the architectural description. Four common integration architectures are therefore described in this chapter.

HEMIT's proposed integration architecture will also be presented in this chapter because CARDIAC is supposed to deliver their first installation of the EOC-system within the Health Region for Central Norway where HEMIT's strategies prevail.

CARDIAC has developed a middleware platform, IMATIS Platform, which should be used in the realization of the integration architecture for the EOC-system. IMATIS Platform is therefore also described in this chapter.

6.1 Information-oriented system integration

Information-oriented system integration is a simple mechanism for the exchange of information between two or more systems. This process is shown in Figure 6.1.

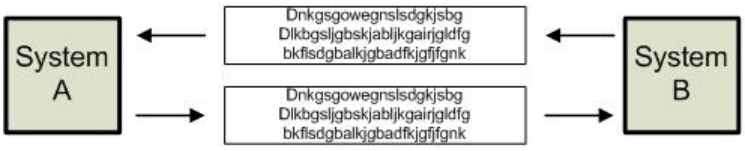


Figure 6.1: When using information-oriented system integration, information is moved between two or more systems [13].

Ideally, integration should occur between the databases or information-producing APIs of the integrating systems. The information exchanged between systems is plain information, not processes or system services.

Information-oriented system integration is easy to understand and in

wide use. It does not contain any notion of behavior which means that it does not deal with complex issues such as state, logic or sequence.

But, information-oriented system integration is not as simple as it might seem at first glance. The book *Next Generation Application Integration* [13] states that *"in order for information-oriented system integration to actually work, architects and developers need to understand all integrated systems in detail"*. One of the problems that might emerge when using this integration architecture is that different systems might have different semantics, and in such cases, systems might end up not understanding each other at all.

Information-oriented system integration does not regard business logic and methods within the source or target systems. In cases where business logic and methods are relevant, service-oriented system integration should be used. Service-oriented system integration is explained in Section 6.3.

6.2 Business process oriented system integration

Business process oriented system integration produces a layer of defined and centrally managed processes on top of existing processes, application services and information within any set of systems. The goal is to combine relevant processes to support the flow of information and to control the logic between them.

Business process oriented system integration is the ability to define a common business process model that addresses the sequence, hierarchy, events, execution logic and information flow between systems residing in the same organization and systems residing in multiple organizations. This common business process model is integrated with the underlying systems by having visibility into their internal system processes, if possible, or perhaps through more primitive layers, such as the database or application interface.

This integration architecture is complimentary to both information-oriented and service-oriented system integration, and even portal-oriented in some cases. As shown in Figure 6.2, business process oriented system integration is really a complete layer above the other integration architectures with the goal to abstract both the encapsulated system services and system information into a single controlling business process model. The architecture consists of three layers:

1. **Business process oriented system integration.** At this layer the system service of information movement is defined.

2. **Transformation, routing and rules.** At this layer information movement and formatting occur.

3. **Messaging service.** This layer is responsible for moving information between all participating systems.

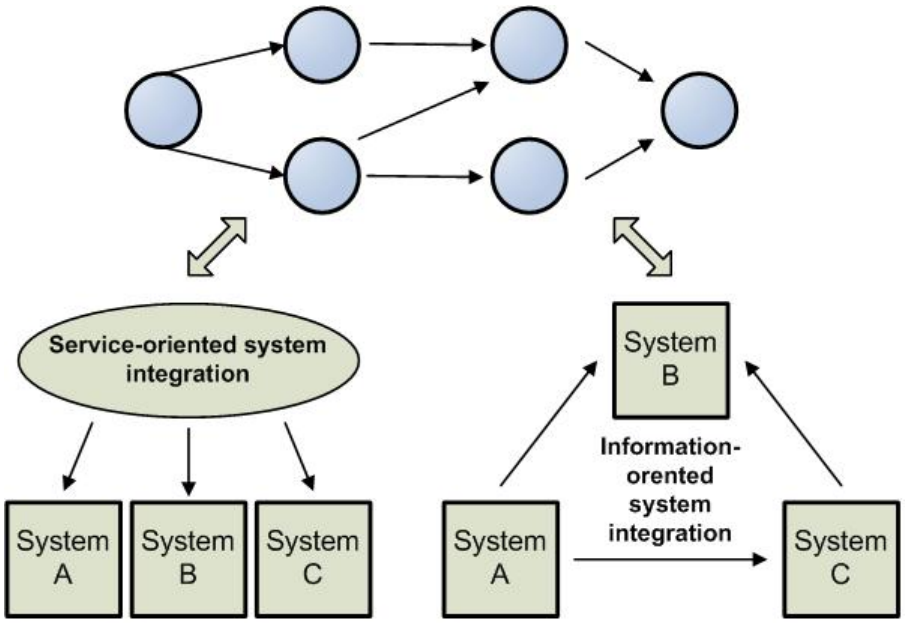


Figure 6.2: Business process oriented system integration provides another layer of control over information-oriented and service-oriented system integration.

With this three layered architecture, information ascends through the layers from the source system where it is processed and descends to the target system where it is delivered.

6.3 Service-oriented system integration

Service-oriented system integration provides a mechanism for binding different information systems together at the service layer. With service-oriented system integration organizations are allowed to share common system services as well as information. A system service is a procedure, method or object with a stable, published interface that can be invoked [1]. Service-oriented system integration is accomplished either by defining system services that can be shared, and therefore integrated, or by providing the infrastructure for such system service sharing. System services may be shared either by hosting them on a central server or by accessing them through distributed objects or standard Web services mechanisms [13].

The book *Next Generation Application Integration* [13] claims that a proper use of Web services is the future of system integration. Web services promise to move beyond the simple exchange of information, which is the dominating mechanism for system integration today, to the concept of accessing system services that are encapsulated within old and new systems.

This means that organizations cannot only move information from system to system, they can also access back-end services found in any number of systems, local or remote. This is the idea behind service-oriented system integration, shown in Figure 6.3.

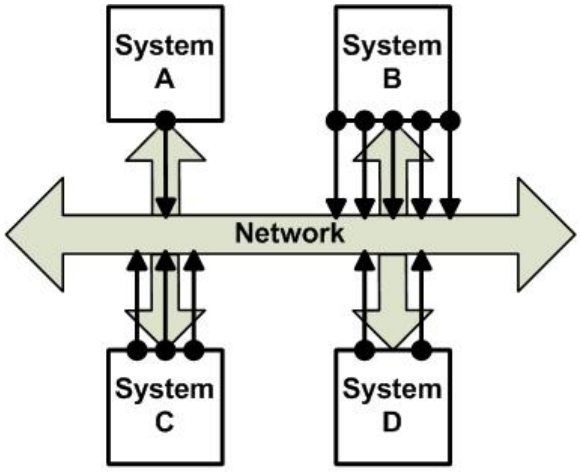


Figure 6.3: A realization of service-oriented system integration by means of Web services.

The definition of Web services varies from generic and all-inclusive interpretations, such as *"Web services provide access to remote application services through the Internet"* [13], to more specific and restrictive types, such as *"Web services describe a standardized way of integrating Web-based applications using the XML* [1]*, SOAP* [2]*, WSDL* [3] *and UDDI* [4] *open standards over an Internet protocol backbone. XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the services available, and UDDI is used for listening what services are available"* [62]. As shown in Figure 6.4, a basic Web services architecture is comprised of SOAP, WSDL and UDDI.

In short, Web services can be thought of as system services exposed by an organization or software program that are both discoverable and accessible by other programs or organizations that are in need of a particular service, such as purchasing an order or reserving a flight. These services are discrete business services with value for many organizations.

[1] XML is the abbreviation for eXtensible Markup Language
[2] SOAP is the abbreviation for Simple Object Access Protocol
[3] WSDL is the abbreviation for Web Services Description Language
[4] UDDI is the abbreviation for Universal Description, Discovery and Integration

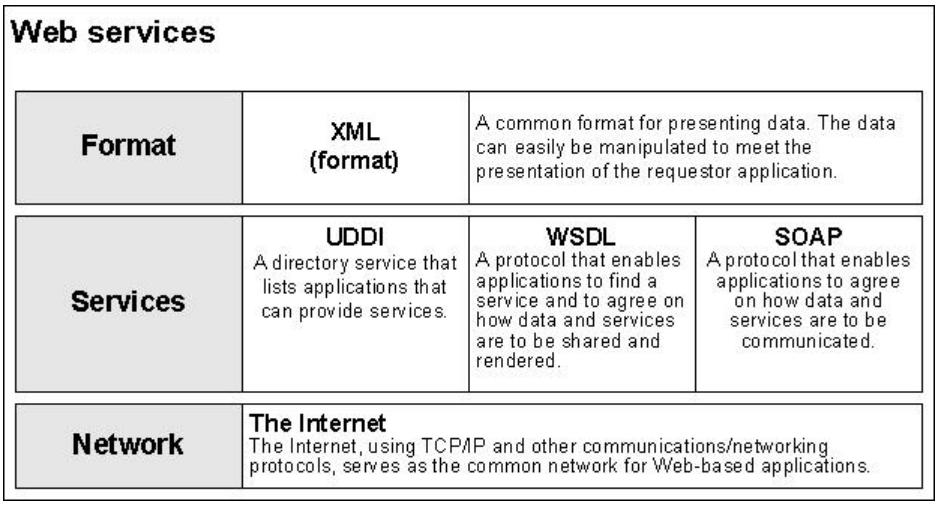


Figure 6.4: Critical elements of a basic Web services architecture [20].

6.4 Portal-oriented system integration

Portal-oriented system integration has become a common and widely used integration architecture by which system integration is accomplished. It is the concept of bringing together information from many different systems, both internal and external systems, within a single user interface.

Portal-oriented system integration avoids the back-end integration problem altogether by extending the user interface of each system to a common, aggregated user interface - most often a Web browser. All participating systems are integrated through the browser, although they are not directly integrated within or between the organizations; each back-end system is accessed through a point of integration e.g. a user interface, database or application server.

In reality, portals are Web-enabled applications consisting of the following components [13]:

- **Web clients** are a PC or any device running a Web browser. The Web browser makes requests to the Web server and processes the results from this server.
- **Web servers** are at the core file servers. Within the concept of portals, Web servers are essential because they enable access to information on database servers or application servers.
- **Database servers** respond to requests and return information.
- **Back-end systems** exist either within a single organization or across many organizations. Portals gather the appropriate information from

these back-end systems and externalize this information through the user interface typically by means of connectors, adapters or APIs.

- **Application servers** provide a middle layer between the back-end systems, databases and the Web servers. They provide the interface development environments (IDEs) for designing the user interface, programming environment for defining system behavior and back-end connectors for moving information in and out of back-end systems.

The portal-oriented integration architecture with its components is shown in Figure 6.5.

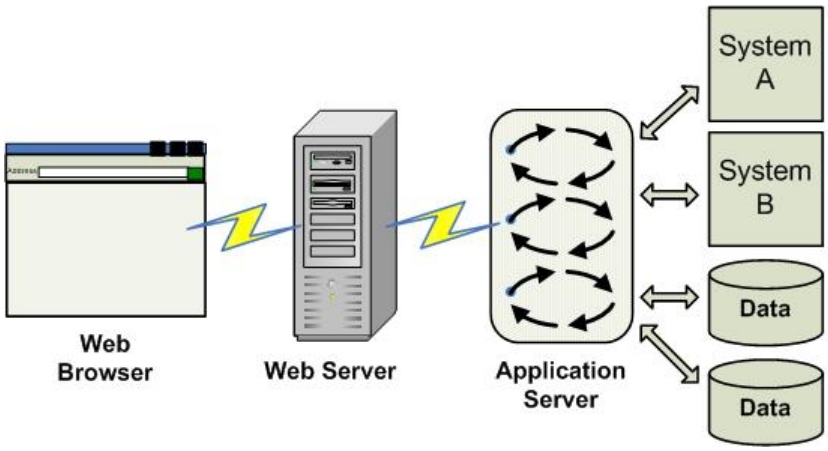


Figure 6.5: The portal-oriented integration architecture and its components [13].

6.5 HEMIT's integration architecture

HEMIT's strategies, which are mentioned in Section 3.2.3, are central in this thesis since HEMIT operates in the Health Region for Central Norway where the first installation of CARDIAC's EOC-system shall run. HEMIT has proposed an integration strategy [51] and an IT architecture strategy [47]. HEMIT recommends a service-oriented architecture where services may be presented through a portal. The service-oriented architecture implies that the integrated systems expose their functionality as services (Web services).

The suggested service-oriented architecture is divided into layers HEMIT: **Application layer** covers the presentation of data to the users. To reduce complexity regarding integration of health information systems, all different interfaces between systems have to be removed. One common user interface is required. The application layer consumes services from the service layer.

Service layer contains and organizes the business logic in a structured way. As shown in Figure 6.6, the services are grouped into logical service

areas, such as patient and record, which are related to the business processes relevant for the application layer.

The service layer should provide services implemented as Microsoft .NET components accessible as Web services. SOAP, WSDL and UDDI are the proposed Web service technology [47].

Source system layer comprises existing source systems, such as PAS, EPR-systems and laboratory systems.

The information model in the source system layer should be non-replication. Replication of data as a consequence of the integration need should be avoided. This means that it is not acceptable to replicate information from system A to another system B.

An example of what these layers may comprise is shown in Figure 6.6.

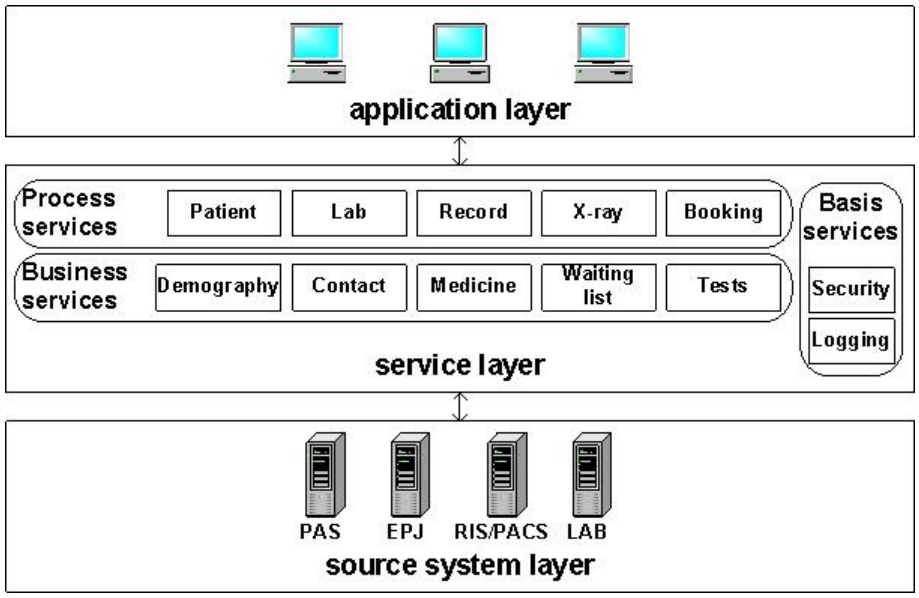


Figure 6.6: HEMIT's recommended service-oriented integration architecture.

6.6 IMATIS Platform

CARDIAC has developed a middleware platform called IMATIS Platform, which they are hoping to use when integrating the EOC-system with other health information systems. The IMATIS Platform is a system platform of different modules containing different software adapted to different businesses and their varying requirements. The IMATIS Platform provides clients with complex data systems a powerful means to integrate and combine data from numerous sources. The IMATIS Platform gives access to real-time data from laboratory, warehouse, Web services or other systems

through a common platform. Historical data and calculations are available through the Web browser and can be presented as trends, reports, tables or graphs.

Figure 6.7 illustrates the IMATIS Platform. As shown in this figure, IMATIS Platform is service-oriented and portal-oriented.

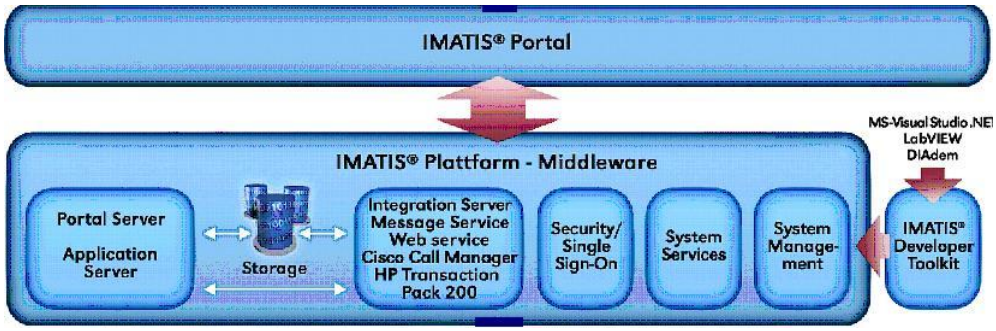


Figure 6.7: IMATIS Platform is a service-oriented and portal-oriented middleware platform.

6.7 Summary

Sometimes a combination of the integration architectures presented above is common to completely describe the architecture of integrated health information systems with different natures.

As mentioned earlier, HEMIT's two strategies have to be followed when developing CARDIAC's EOC-system. These two strategies aim for a service-oriented architecture where functionality is presented through a common user interface, a portal.

In addition, CARDIAC has already developed IMATIS Platform, a middleware platform which is based on a service-oriented and portal-oriented integration architecture, and they want to implement the EOC-system on this platform.

Following HEMIT's recommendations together with the other aspects mentioned above, CARDIAC's EOC-system should be developed on the basis of a service-oriented and portal-oriented integration architecture.

Chapter 7

Summary

Part II puts focus on the Norwegian health service in general and CARDIAC's EOC-system specifically.

CARDIAC's EOC-system is supposed to be integrated with several other health information systems. The following health information systems are chosen for integration: EPR-system, PAS, RoS, Medication, EQS and MTU.

The prestudy has also identified important security mechanisms which will be included in the architectural description. These mechanisms are: digital signing, secure communication, auditing and access control.

Finally, a service-oriented and portal-oriented integration architecture was chosen for CARDIAC's EOC-system.

All this information is necessary in order to get a solid basis for describing a security focused integration architecture for CARDIAC's EOC-system. In the next part of the report a methodology for creating an architectural description is presented.

Part III

Methodology

Chapter 8

Introduction

Creating an architectural description of a system is a complex task with many issues that need to be kept in mind. Therefore, it is often useful to have a methodology or a tool for guidance through the most important parts of an architectural description.

Model-based Architecture Framework For Information Integration Abstraction (MAFIIA) is an architectural description framework which is suitable as a method for creating an architectural description for CARDIAC's EOC-system.

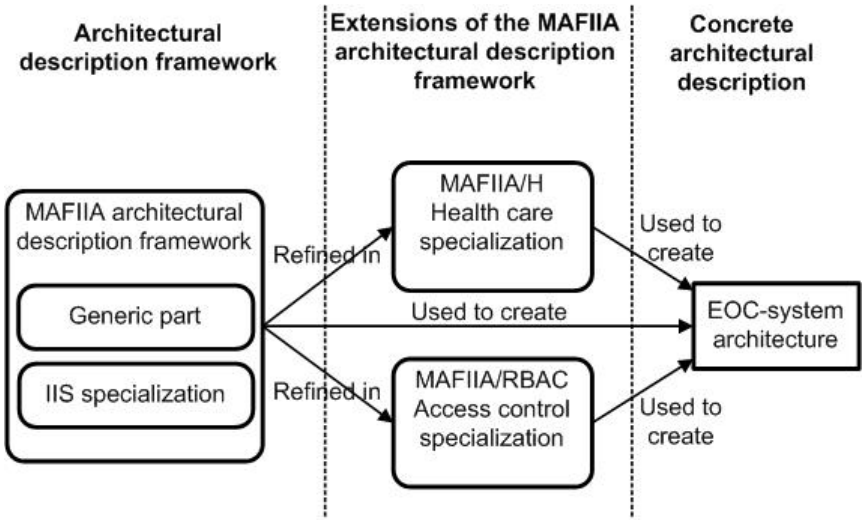


Figure 8.1: Overview of MAFIIA, specializations and use.

Figure 8.1 gives an overview of the architectural description frameworks and their relations. The MAFIIA architectural description framework consists of two parts: a generic part and a part which is specially target towards Information Integration Systems (IIS). In addition, two extensions of MAFIIA are shown in the figure, MAFIIA/H and MAFIIA/RBAC.

Chapter 9

MAFIIA

This chapter gives an overview of the Model-based Architecture description Framework for Information Integration Abstraction (MAFIIA) [16], which is an architectural description framework for software intensive systems.

MAFIIA architectural description framework assists software architects in developing the architecture of a system. The focus of the framework lies on development, documentation and specialization for domain and application types. MAFIIA is created by SINTEF Telecom and Informatics in Trondheim.

Although the MAFIIA architectural description framework consists of two parts, respectively a generic part and a part concerned with information integration systems, only the generic part will be described in this chapter. The reason for this is that integration is central in this thesis and the importance of the second part of the MAFIIA architectural description framework is enhanced by placing it in a separate chapter.

9.1 MAFIIA Concepts

The generic part of the MAFIIA architectural description framework, hereby referred to as *generic MAFIIA*, defines a number of concepts and explains their use in an architectural description. These concepts and their relations are shown in Figure 9.1. A description of each concept is found in the following sections.

9.2 Concerns

The generic MAFIIA mentions the possibility of having specific concerns when creating the system. Concerns deal with the documentation of the functionality of the system. Important functionality, being the system's capabilities and services of the system, should be treated as a concern.

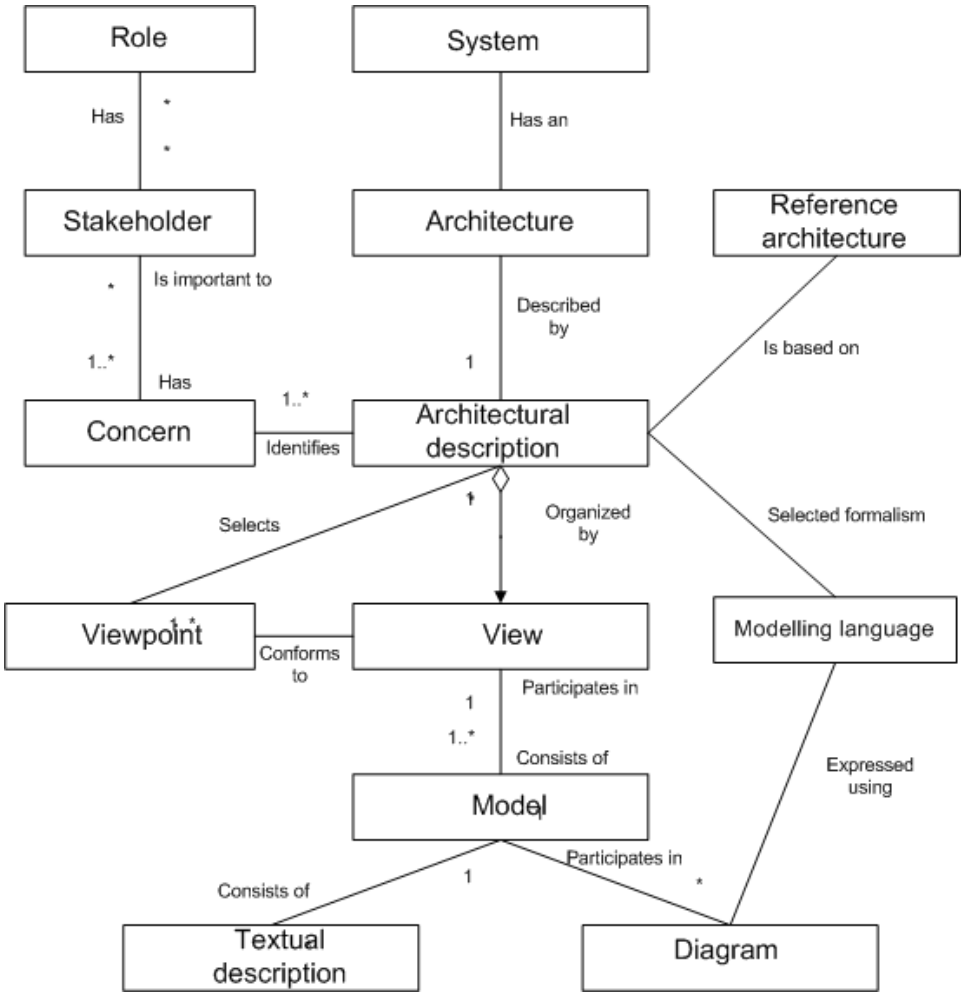


Figure 9.1: MAFIIA concepts and their interrelations. Adapted from the generic MAFIIA [16].

Some examples of concerns could be flexibility, performance, maintainability, safety and security.

The concerns in the generic MAFIIA are grouped into two groups: *Application Specific Functionality Concerns (ASFC)* and *Quality Related Functionality Concerns (QRFC)*. ASFC deals with functionality which would be necessary in an ideal world where system failures and performance problems are not considered. This kind of functionality may, or may not, be possible to implement in real life. QRFC is related to all types of functionality which is used to improve the system quality.

9.3 System assets

System assets are resources that are considered useful when creating the architectural description of the target system. These can be specific standards, general architectural patterns or (software) tools which can help the process of creating the architectural description.

Assets listed in the generic MAFIIA are *Dictionary*, *Standards* and *Patterns*.

9.4 Reference architecture

In this architectural description framework, a reference architecture is defined as *"a high-level architecture which is used as the basis in development of concrete system architectures"*.

An architecture of this type only separates the target system from the environment and the interface used between them. Figure 9.2 shows the generic reference architecture defined in MAFIIA.

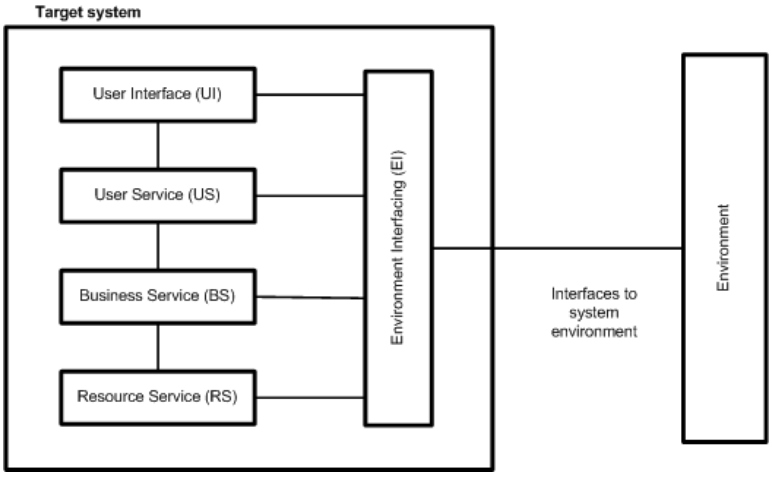


Figure 9.2: The reference architecture defined in the generic MAFIIA [16].

9.5 Stakeholders and roles

Different people who are involved in a project have different interests in the system which is being developed. This means that these people, the stakeholders, have different roles in the process of creating the architectural description of the system. The generic MAFIIA does not define specific stakeholders. The stakeholder definition is based on the description in the IEEE1471-2000 standard which says that the stakeholders normally fill two key roles. These are the role of the acquirer and the role of the architect.

9.6 Modeling language

A modeling language is a semi-formal notation used to express design [8]. An equally understood and agreed upon modeling language helps discuss and communicate the architectural description. The generic MAFIIA defines UML 1.4 as a modeling language, with the addition of textual descriptions, for the documentation of the different viewpoints. The use of UML is mandatory.

9.7 Viewpoints

Computer software systems are often so complex that it is difficult to understand them all at once. An addition to the complexity of discussing the structure of the system, is the different background and concerns of the stakeholders. To make the discussion of a system structure manageable, it is useful to restrict the attention to a smaller subset of the system structure, namely views and viewpoints.

A *view* is defined to be the representation of a coherent set of architectural elements, as written by and read by system stakeholders [2]. A *viewpoint* is a description on how to create a view. In other words, a view is what you see when looking at the system from a specific viewpoint [16].

In the generic MAFIIA, a viewpoint consists of one or more models, a description of how to view a system and recommendations for what diagrams to use in each of the models. Five viewpoints are defined in the generic MAFIIA:

- **Context viewpoint** is concerned with the environment of the system, such as stakeholders of the system and other systems which the actual system is connected to.
- **Requirement viewpoint** has its focus on functionality and quality of the system.
- **Component viewpoint** deals with the decomposition of the system into physical and logical components.
- **Distribution viewpoint** takes care of the logical organization of the system, e.g. the logical distribution of software and hardware components.
- **Realization viewpoint** describes the realization of the subsystems and possible constraints on implementation and deployment of the system's components.

Table 9.1 gives an overview of all viewpoints in the generic MAFIIA and the models recommended for use in each viewpoint.

Viewpoint	Models required	Description
Context viewpoint	**Business Aspects Model**	Gives an understanding of what problems the system is supposed to solve and what functionality it should implement. Only users of the system and the processes they carry out are included.
	Environment Systems Model	Concerned with other systems in the environment and their relations to the target system.
	Business to System Mapping Model	Shows the mapping of Business Aspects Model to technical solutions of the Environment System Model.
Requirement viewpoint	**Requirement Model**	Lists all the relevant requirements for the system.
	Target System Interface Model	Gives a more complete and easier specification of the requirements from the Requirements Model
Component viewpoint	**System Information Model**	Shows relationships between information objects in the system.
	System Decomposition Model	Shows the division of the system into subsystems and relations between them.
	System Collaboration Model	Concerned with how the subsystems or components interact with each other.
	Component and Interface Specification Model	Shows the details of the components and interfaces which were already defined in the System Decomposition Model and System Collaboration Model.
Distribution viewpoint	**System Distribution Model**	Gives an overview of the logical components of the system and how they are organized.
	Role Distribution Model	Shows the distribution of the roles which are defined in the system.
Realization viewpoint	**System Deployment Model**	Shows the physical relations between software and hardware components of the system.
	Technology Mapping Model	Shows how the system components map into the actual implementation in hardware and software.
	System Integration Test Model	Describes test scenarios for use in verification of the correctness of the system behavior.

Table 9.1: Overview of viewpoints and models in the generic MAFIIA.

9.8 Context viewpoint

The purpose of the context viewpoint is to describe the environments to the target system in terms of its business-related aspects, other involved technical systems and the mapping of business aspects to the target system. The context view shall only document the environments of the target system, and not the target systems itself.

Stakeholders that are addressed in this viewpoint are acquirers, such as buyers, customers, owners, users or purchasers and architects.

Required models in the context view are:

- Business Aspects Model
- Environment System Model
- Business to System Mapping Model

9.8.1 Business Aspects Model

The business aspects model shall document any business related concern that will increase the understanding of what problems the target system shall solve, or what functionality it shall implement. It takes customer supplied information as input, i.e. requirements to target system and business related information. This model may result in a textual description, a UML class diagram, a UML use case diagram or a UML collaboration diagram.

9.8.2 Environment Systems Model

The environment systems model shall document other technical systems (environment systems) that will be involved in the implementation of the Business Aspects Model, or influence the operation of the target system. The input to this model is customer supplied information, i.e. target system requirement specifications and environment system documentation. The output from such a model may be a textual description, a UML sequence diagram, a UML use case diagram or a UML collaboration diagram.

9.8.3 Business to System Mapping Model

The purpose of the Business to System Mapping Model is to document what parts of the documented Business Aspects Model are mapped to technical solutions constituted by environment systems and the target system, and how the different parts of the Business Aspect Model are mapped to the different involved systems. The input comes through customer supplied information, i.e. target system requirement specifications and requirements/needs related to the business aspects. The output of this model is either a textual description or a UML use case.

9.9 Requirement viewpoint

The requirement viewpoint deals with describing functional and non-functional (quality) requirements of the target system. It consists of the following models:

- Requirement Model
- Target System Interface Model

9.9.1 Requirement Model

The Requirement Model shall be complete in identifying and eventually specifying all relevant requirements to the target system where a requirement shall be verified. The input to this model comes from the context viewpoint and the output may be a textual description, a UML sequence diagram, a UML collaboration diagram, a UML use case or a UML class diagram.

9.9.2 Target System Interface Model

The Target System Interface Model should be a supplementary specification to the Requirement Model to obtain more complete and easier understandable specification of the target system's interfacing to its environments. Input to this model is the Business to System Mapping Model and the Requirement Model. This model may be presented by a UML sequence diagram, a UML use case diagram or a UML collaboration diagram.

9.10 Component viewpoint

Component viewpoint has its focus on information, system decomposition, system collaboration and component and interface specification.

The viewpoint consists of the following models:

- System Information Model
- System Decomposition Model
- System Collaboration Model
- Component and Interface Specification Model

9.10.1 System Information Model

The purpose of the System Information Model is to specify the relationships between and properties of the central information objects in the system that must always be true (invariants). Input to this model comes from the requirement viewpoint. UML class diagrams and textual descriptions are suggested outputs for this model.

9.10.2 System Decomposition Model

The System Decomposition Model describes how the system is divided into different subsystems or components, and how these are related to form a coherent whole. This model takes input from the requirement viewpoint. UML class diagrams are suggested as output for this model.

9.10.3 System Collaboration Model

The purpose of the System Collaboration Model is to describe the main interactions in the system as a set of collaborating components. Input to this model comes from the Target System Interface Model (Requirement viewpoint). Recommended output is a UML class diagram, a UML activity diagram, a UML sequence diagram, a UML collaboration diagram or a textual description (area of concern).

9.10.4 Component and Interface Specification Model

This models describes each of the main components and interfaces of the target system, including operation signatures and behavior. Input to this model should be the requirement viewpoint, the System Decomposition Model and/or the System Collaboration Model (Component viewpoint). Recommended output may be a UML class diagram, a UML state chart diagram or a textual description.

9.11 Distribution viewpoint

The distribution viewpoint deals with the logical separation of components. The following models are a part of this viewpoint:

- System Distribution Model
- Role Distribution Model

9.11.1 System Distribution Model

The System Distribution Model shall describe logical units or components that must be distributed and deployed together. Input to this model is the System Decomposition Model and the System Collaboration Model (Component viewpoint). Possible output may be a UML deployment diagram or a textual description (rationale).

9.11.2 Role Distribution Model

This model describes the distribution of the different roles that are a part of the target system. The input to this model may be the Business Aspects

Model (Context viewpoint) and the System Distribution Model (Distribution viewpoint). Recommended output is a UML deployment diagram or a textual description.

9.12 Realization viewpoint

The realization viewpoint is concerned with the implementation of the target system. It consists of the following models:

- System Deployment Model
- Technology Mapping Model
- System Integration Test Model

9.12.1 System Deployment Model

The purpose of this model is to describe the set of system deployment configurations. Input to this model could be the System Distribution Model (Distribution viewpoint) or the Requirement Model (Requirement viewpoint). Possible output from this model may be a UML deployment diagram or a textual description.

9.12.2 Technology Mapping Model

The Technology Mapping Model shall describe how system components map to technological solutions, concepts and mechanisms. Recommended input to this model may be the Requirement Model (Requirement viewpoint), Component and Interface Specification Model (Component viewpoint), System Distribution Model (Distribution viewpoint) and System Deployment Model (Realization Viewpoint). Possible output may be a UML component diagram, a UML deployment diagram or a textual description.

9.12.3 System Integration Test Model

The purpose of the System Integration Test Model is to describe a set of test scenarios to be conducted during system deployment (subsystem integration). The input to this model comes from the Requirement Model (Requirement viewpoint) and the System Deployment Model (Realization viewpoint). The output may be given in textual descriptions.

9.13 Summary

This chapter has introduced and described the generic part of the MAFIIA architectural description framework. The generic MAFIIA defines a set of

concepts and prescribes how to use these in the architectural description of a system. Central parts of the generic MAFIIA are: *Concerns*, *Assets*, *Reference architecture*, *Stakeholders and roles*, *Modeling language* and *Viewpoints*.

As mentioned earlier, the MAFIIA architectural description framework consists of one generic part and one specific part for integration systems. The generic part is the one presented in this chapter, and it is later referred to as the generic MAFIIA. The specific part is introduced in the following chapter and referred to as MAFIIA for IIS.

Chapter 10

MAFIIA for Information Integration Systems (IIS)

The MAFIIA architectural description framework contains a part which is specially targeted towards Information Integration Systems (IIS). Still, throughout this thesis MAFIIA for IIS is handled as a separate architectural description framework because this thesis has a special focus on integration.

IIS deals with integration of environment systems into a target system, meaning that two or more environment systems must be interfaced with the target system and that the functionality of the target system will perform physical and/or logical integration of information from the environment systems. In addition, the target system may produce information or events that shall be communicated to the environment systems.

MAFIIA for IIS contains guidance and recommendations for the development and documentation of the architecture of IIS systems.

10.1 IIS specific concepts

The structuring of this chapter follows the structure of the generic MAFIIA and contains the same concepts, namely:

- Concerns
- System assets
- Reference architecture
- Stakeholders and roles
- Modeling language
- Viewpoints

10.2 IIS specific concerns

Two specific concerns are presented in an information integration context, *functionality* and *security*. All new functionality introduced must be included as functional requirements in the requirement viewpoint of the framework and be reflected in the other viewpoints. Security is considered to be very important when integrating information from different systems and should therefore be treated as a separate concern in MAFIIA for IIS.

10.3 IIS specific system assets

A separate dictionary is created for integration of information systems. Data warehouses, federated databases, information transformation and integration are concepts which are defined in the dictionary.

Use of different standards for middleware services, communication protocols and databases and information exchange is suggested.

MAFIIA for IIS also contains suggestions for use of several architectural and design patterns. The use of architectural and design patterns helps capturing best practice solutions; they reuse experience. The patterns suggested in MAFIIA for IIS are:

- **Adapter pattern** converts interfaces of existing resources to the expected interfaces [9].
- **Blackboard pattern** assemblies knowledge from several specialized subsystems to build a possibly partial or approximate solution.
- **Client-Server-Dispatcher pattern** provides location transparency between clients and servers [3].
- **Composite pattern** lets clients treat individual objects and compositions of objects uniformly [9].
- **Façade pattern** constructs a façade interface, which is a unified interface to a set of interfaces in a subsystem. Clients only access the subsystem through the façade [9].
- **Flyweight pattern** uses sharing to support large numbers of fine-grained objects efficiently [9].
- **Forwarder-Receiver pattern** provides transparent interprocessing communication for software systems with a peer-to-peer interaction model [3].
- **Master-Slave pattern** constructs a master component which distributes work to identical slave components and computes a final result from the results these slaves return [3].

- **Pipes and Filters pattern** divides tasks of a system into a sequence of processing steps [3].
- **Proxy pattern** makes the clients of a component communicate with a representative rather that the component itself [3].
- **Publisher-Subscriber pattern** synchronizes the state of cooperating components by a one-way propagation of changes: one publisher notifies any number of subscribers about changes to its state [3].
- **Reflection pattern** provides a mechanism for changing the structure and behavior of software systems dynamically [3].
- **Wrapper Façade pattern** constructs a wrapper façade interface which encapsulates functionality only available through non-object-oriented API's [4].

10.4 IIS specific reference architecture

Figure 10.1 shows a basic reference architecture for IIS systems. The only difference between this reference architecture and the generic reference architecture presented in Chapter 9, is the environment part of the reference architecture. The IIS specific reference architecture is targeted towards integration of systems, and it therefore connects the environment interface of the target system towards the user interface of other systems.

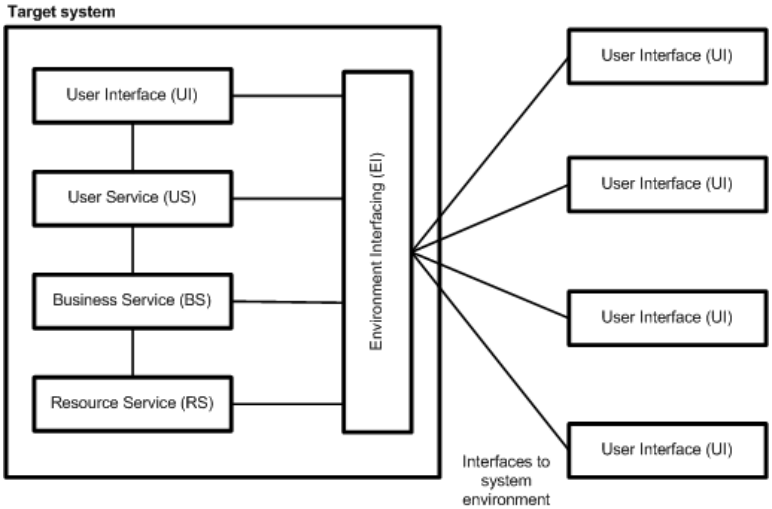


Figure 10.1: IIS specific reference architecture.

10.5 IIS specific viewpoints

MAFIIA for IIS suggests two new models, namely the System Security Model in the component viewpoint and the System Security Model in the distribution viewpoint. These two models are described in the subsequent sections.

10.5.1 System Security Model (Component viewpoint)

The purpose with the System Security Model is to describe how security concerns are handled by the components in the target system. Input to this model is the requirement viewpoint and the Decomposition Model (Component viewpoint). Recommended output from this model is a UML class diagram, a UML sequence diagram, a UML state chart diagram or textual description.

10.5.2 System Security Model (Distribution viewpoint)

The System Security Model in the distribution viewpoint shall describe the effects of the security concerns on the other models defined in the distribution viewpoint. Input is the System Security Model from the component viewpoint, while output may be a UML use case diagram, UML collaboration diagram or textual description.

10.6 Summary

MAFIIA for IIS is a specialization of the generic MAFIIA. MAFIIA for IIS contains guidance and recommendations for development and documentation of the architecture of IIS systems.

MAFIIA for IIS recommends that security is specified as a concern. MAFIIA for IIS also extends the generic MAFIIA with two new models: *System Security Model* in the component viewpoint and *System Security Model* in the distribution viewpoint.

Chapter 11

MAFIIA/H

MAFIIA/H [17] is an extension of the MAFIIA architectural framework which is specifically targeted towards the health domain. MAFIIA/H is developed by SINTEF Telecom and Informatics in Trondheim. It describes system assets and concerns that are typical for IIS systems in the health domain. These system assets and concerns should be taken into account when developing an architectural description for IIS systems, and they are therefore described further below.

11.1 Health care specific assets

In the following, relevant assets for the health domain will be described.

11.1.1 Dictionaries

Some available health care dictionaries are listed below:

- **Systemized Nomenclature of Human and veterinary Medicine (SNOMED)** is recognized globally as a comprehensive, multiaxial, controlled terminology created for the indexing of the entire medical record [54].
- **Unified Medical Language System (UMLS)** supports the development of systems that help health professionals and researchers retrieve and integrate electronic biomedical information from a variety of sources and make it easy for users to link disparate information systems [53].
- **The GALEN project** aims at the development of a reference model for medical concepts [52].
- **International Classification of Diseases (ICD)** is an international standard diagnostic classification for all general epidemiological and

many health management purposes [55]. Translated to Norwegian by KITH and others.

- **International Classification of Primary Care (ICPC)** is a two-axis system primarily oriented towards body systems [55]. Translated to Norwegian by KITH and others.

When integrating information from semantically heterogeneous sources, it is necessary to use a dictionary. The original information sources must be related to the dictionary according to certain rules for translations in order to provide a common information model for the integrated system. The mapping or translation can be done by using standard patterns for information integration.

The interfacing systems' information models and dictionaries should be identified and documented as part of the context viewpoint, while the strategy for mapping or translation should be described in the System Information Model as part of the component viewpoint.

11.1.2 Standards

There are many types of standards that have to be followed when architecting a new IIS. Some of them are international, while others are national, e.g. legislations regarding documentation of work done by health care providers.

Some relevant Norwegian legislations and regulations are listed below:

- **Health Personnel Act** [1] [36].
- **Patients' Rights Act** [2] [38].
- **Patient Record Regulations** [3] [43].

MAFIIA/H also lists some international standards for standardizing the information models within the health domain:

- **CEN TC251** is a standardization to achieve compatibility and interoperability between independent systems and to enable modularity [48].
- **Health Level 7 (HL7)** is an ANSI[4] standard for health care specific data exchange between computer applications. The name *Health Level 7* refers to the top layer (Level 7) of the Open Systems Interconnection (OSI) layer protocol for the health environment [44].

[1] Norwegian: Helsepersonelloven
[2] Norwegian: Pasientrettighetsloven
[3] Norwegian: Pasientjournalforskriften
[4] American National Standards Institute

- **ISO 18308: Health Informatics standard** technically specifies how to assemble and collate a set of clinical and technical requirements for an EPR architecture that supports using, sharing, and exchanging EPRs across different health sectors, different countries, and different models of health care delivery. It suggests requirements for the architecture but not the specifications of the architecture itself [45].
- **Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT)** is used as a messaging standard in health care [66].
- **DICOM** is a picture standard that covers all areas related to the transportation, storage and display of medical pictures [66].

In the health domain, the World Health Organization (WHO) plays an important role in the standardization process. In Norway, KITH is responsible for the standardization work.

11.2 Health care specific concerns

Health informatics involves a high degree of sensitive information. Security is therefore treated as a separate concern in MAFIIA/H. In MAFIIA/H, four major aspects of medical security is identified. These aspects are presented in the subsequent sections.

11.2.1 Reliability

More and more health care organizations become dependent on the functioning of their information systems. This implies that reliability is a major concern for these kinds of systems.

11.2.2 Data completeness

When integrating data from heterogeneous information sources, some record fields may become incomplete. Incomplete data may result in uncertainty. When data are not found in the patient record, this might mean that no abnormalities were found, the data were not available or collected, or that it was lost during integration. Thus, data completeness must be handled thoroughly during information integration.

11.2.3 Data accuracy

MAFIIA/H characterizes data accuracy as correctness or conformity. Correctness is a measure of the error rate of the data. Errors can be made during data collection and during integration of data. Conformity of data

pertains to following standards of classification systems for data recording. Sometimes instructions of the classification systems are not conformed.

Integration of data from systems being operated and maintained by different departments may lead to data inaccuracy and errors.

11.2.4 Data precision

Data precision deals with the degree of refinement or granularity by which a measurement is expressed, such as the number of decimal places.

11.3 Summary

In this chapter, MAFIIA/H is introduced. MAFIIA/H is an extension of MAFIIA, and it describes concerns and assets related to the health domain. These concerns and assets should be taken into account when developing an architectural description of an IIS system.

Chapter 12

MAFIIA/RBAC

MAFIIA/RBAC is the result of a student project done by Andreas G. Furuseth and Mirela Divic, during the fall of 2004 [32]. MAFIIA/RBAC is an extension of the MAFIIA architectural description framework [16], which was described in Chapter 9 and Chapter 10. The purpose with MAFIIA/RBAC is to add a specific concern to MAFIIA, and this concern is to achieve access control with the use of Role-Based Access Control (RBAC).

As stated in MAFIIA/RBAC [32], this extension is a theoretically created architectural description framework and future work in MAFIIA/RBAC suggests to use it in creating an architectural description of a real system. One of the motivations behind this thesis is to try MAFIIA/RBAC on a real case within the health domain and verify the quality of this framework.

12.1 RBAC specific concepts

MAFIIA/RBAC follows the structure and naming principle in the generic MAFIIA. But with special attention on RBAC, MAFIIA/RBAC presents certain changes to some of the concepts from MAFIIA and adds some new models and ideas to the framework.

12.2 RBAC specific concerns

In MAFIIA/RBAC, RBAC is handled as a separate concern.

12.3 RBAC specific system assets

12.3.1 Standards

MAFIIA/RBAC presents two standards that are relevant for RBAC, namely:

- **The proposed NIST standard** presents a reference model which defines sets of basic RBAC elements and relations. Basic RBAC elements are subjects, roles, permissions and operations [6].
- **Common Criteria** is a standard that defines criteria for evaluation of IT security. Different criteria for evaluation of IT security have been developed in USA, Canada and Europe. Common Criteria combines all these criteria into one common international standard [26].

12.3.2 Role-cards

MAFIIA/RBAC introduces the concept of *role-cards* for use when identifying roles relevant for the target systems. This technique is based on the UML use case collaboration process from the book *UML Distilled* [8] and the Class-Responsibility-Collaboration (CRC) card technique, which is explained more in *Using CRC cards* [61]. Role-cards can be used on wide-ranging roles, but their strength is seen when moving toward design level roles. An advantage of using the role-card method is that stakeholders are presented and familiarized with roles.

An example of role-cards is shown in Figure 12.1. The fields SSD and DSD in the figure are concerned with separation of duties in RBAC.

SSD is an abbreviation for static separation of duties and puts constraints on role assignment to users. A user who is assigned to one role may be prevented from being a member of a second role.

DSD stands for dynamic separation of duties. It allows users to be authorized for roles that may conflict, but it introduces limitations while the user is active in the system [6]. DSD may for example deny a user to be active in both roles at the same time, if the roles are in conflict (e.g. mutually exclusive).

Role id 5	**Role name** Supervising Physician	**Inherits from** Physician
Role description A Supervising Physician is reviewing tasks performed by a Physician Assistant (PA), and therefore needs access to drug orders, consultations and EPR for the patient the PA is treating.		
SSD with Physician Assistant		**DSD with**

Figure 12.1: Sample figure of a role-card used for documenting/discussing different roles. Idea to the sample role taken from CAPA [70].

12.3.3 RBAC specific patterns

While no patterns are suggested in the generic MAFIIA, three different patterns are suggested for use in MAFIIA/RBAC. These patterns are:

- **Single access point pattern** provides only one entry point for accessing a system or application.
- **Check point pattern** provides a structure for security checks according to the given security policy.
- **Role-based access control pattern** assigns access rights to users according to their roles in the organization.

12.4 RBAC specific reference architecture

MAFIIA/RBAC does not take environment systems into account. The reference architecture of MAFIIA/RBAC therefore does not include interfacing towards environment systems. The reference architecture of MAFIIA/RBAC is shown in Figure 12.2.

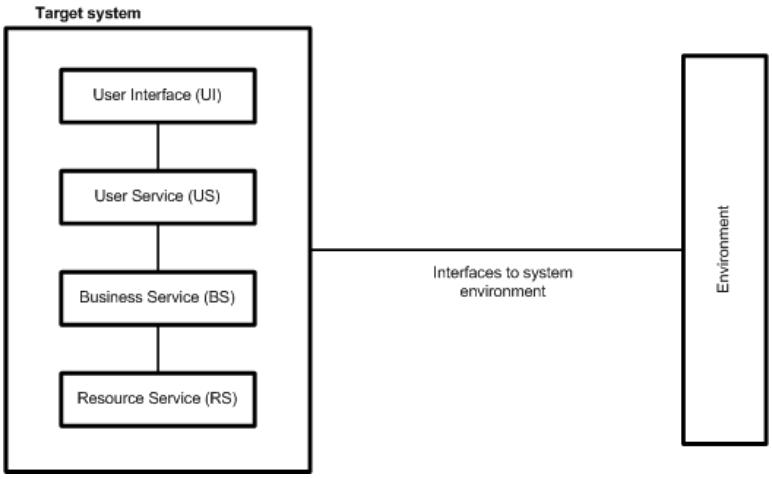


Figure 12.2: The reference architecture of MAFIIA/RBAC.

12.5 Modeling language

In the generic MAFIIA it is mandatory to use Unified Modeling Language (UML). MAFIIA/RBAC suggests use of UMLsec as a modeling language in addition to the generic UML. *UMLsec* is a profile which allows security-related information and concepts (e.g. smart cards, encryption, secrecy, access control, integrity, etc.) in UML models. The profile contains stereotypes

with tags and constraints to communicate security requirements and security attainment. MAFIIA/RBAC only uses a subset of the defined stereotypes from UMLsec. These stereotypes are: *protected*, *role*, *right* and *guard*. UMLsec is described more thoroughly in C.10.

12.6 RBAC specific viewpoints

As mentioned earlier, MAFIIA/RBAC follows the structure and naming principle of the generic MAFIIA and defines the same concepts and the same viewpoints. In the case where the same models are used in MAFIIA/RBAC as in the generic MAFIIA, they are used in a *new* way, with RBAC in mind. For more information on MAFIIA/RBAC, or the changes from generic MAFIIA to MAFIIA/RBAC, see [32]. In addition to adding RBAC to the models in the generic MAFIIA architectural description framework, MAFIIA/RBAC also adds two new models to the framework. These models are described in the subsequent sections.

12.6.1 Target Organization Security Policy Model

Target Organization Security Policy Model in the requirement viewpoint identifies and connects the architectural description to the security policy of the target system's organization.

12.6.2 System Access Control Model

System Access Control Model in the component viewpoint describes the relationship and properties of components in the system that form the access control subsystem and how they enforce the security policy.

12.7 Summary

MAFIIA/RBAC is an extension of the MAFIIA architectural description framework which is concerned with access control, more precisely RBAC. MAFIIA/RBAC only focuses on RBAC inside the target system and does not include interfacing environment systems in its reference architecture. MAFIIA/RBAC adds role-cards, UMLsec, Target Organization Security Policy Model (Requirement viewpoint) and System Access Control Model (Component viewpoint).

Chapter 13

Summary

This part of the report has presented the architectural description frameworks which will be used as tools in creating the architectural description for CARDIAC's EOC-system. Throughout the rest of this report, the architectural description for CARDIAC's EOC-system will primarily be based on the generic MAFIIA. Where there is a special need for integration input, MAFIIA for IIS will be used. Accordingly, where the health domain or role-based access control play an important role, the extensions MAFIIA/H or MAFIIA/RBAC will be used.

Part IV

Architectural Description

Chapter 14

Introduction

The aim of this thesis is to create a description of a security focused integration architecture for CARDIAC's EOC-system. This part constitutes the architectural description for CARDIAC's EOC-system. The architectural description is based on the MAFIIA architectural description frameworks, and this part will therefore be organized following the structure in these frameworks.

First concerns, assets and reference architecture are described. Then, each viewpoint in MAFIIA is presented in subsequent chapters. The description of each viewpoint is a set of models, textual descriptions, tables and diagrams which present various approaches to the architectural description.

Chapter 15

Concepts

This chapter presents some of the MAFIIA concepts in relation to the system that shall be developed based on this architectural description. The particular system is CARDIAC's EOC-system. The EOC-system shall contain a gathering of patient information from several different health information systems.

Concepts described in this chapter are concerns, assets, reference architecture, stakeholders, roles and modeling language.

15.1 Concerns

An integration of health information systems in general has several advantages:

- redundancy of information is prevented
- manual typing errors are reduced
- a common and comprehensive information base is available

The goal of such an integration is better quality of patient care and increased efficiency of provided health services within the hospital.

But despite the above mentioned improvements, there are certain challenges that have to be managed when integrating health information systems. There are several laws and regulations concerning registration of a patient's medical history. In short, all information about a patient's medical treatment has to be stored in a patient record. The information in a patient record is extremely sensitive and therefore needs to be highly protected. It is therefore important to maintain a high level of security in the health information systems which constitute a patient record.

This section identifies security concerns that are of particular importance for an integration of health information systems, specially those systems which compose an EOC-system. The listed concerns will be specially

treated in the viewpoints and models of the MAFIIA architectural description frameworks, and they will influence the architectural description for CARDIAC's EOC-system.

15.1.1 Digital signing

The value of a human life is immeasurable. When a person is hospitalized, he is treated by several different doctors and nurses during one hospital stay. It is therefore extremely important to be able to find out which doctor or which nurse performed the treatment that may have caused a patient's death or deterioration in the clinical picture of a patient. Every action performed towards a patient must therefore be registered in the EOC and signed by the person responsible for it.

When integrating health information systems, the number of users increases dramatically, and digital signing becomes even more important.

The mechanism used for digital signing of certificates or other documents is a digital signature. A digital signature confirms that a particular person has written and/or approved a document, and the receiver of the document is able to prove that this person really signed it and that the document has not been altered since the signing.

Typically, public key infrastructure (PKI) is used for digital signatures. PKI is a collective term for technology providing unique digital identities across networks, where digital identity is used for authentication, digital signing of information and secure communication [65].

15.1.2 Secure communication

An integration of health information systems implies connecting the systems through a network. In order to protect the sensitive patient information which flows between the integrated systems, secure communication must be enforced.

Secure communication includes secure transmission of information and mutual authentication of the entities participating in the communication.

By far the most important automated tool providing secure communication is encryption. Both *asymmetric* (also referred to as public-key) and *symmetric* (also referred to as conventional) encryption is in common use. With asymmetric encryption, different keys are used for encryption and decryption. The encryption key can be made public, while the decryption key has to remain private. With symmetric encryption, a *secret key* is used for both encryption and decryption [10]. Most commonly, symmetric encryption is used in the main part of the communication, while asymmetric encryption is used only to exchange a secret key.

15.1.3 Auditing

When dealing with several integrated health information systems, it is important to have the possibility to analyze the behavior of users or possible intruders, in order to detect security violations and improve the protection of the system's security. This analysis is often performed automatically on the basis of an audit log.

Auditing is a posteriori technique for detection of security violations or other suspicious events, with the purpose of ensuring traceability within the system. This should include tracing access to sensitive information stored in the information systems as well as access to the information systems themselves. Each access and/or access attempt should be recorded in an audit log for later analysis [10].

The audit log should include information about the one who is accessing information, services or resources inside the system, the actual information, services or resources which are accessed, and the time they are accessed.

15.1.4 Access control

Access control is known to be one of the most complex security mechanisms in the health domain. Different users should have different access rights to the information inside a system. Role-based access control (RBAC) is mostly used within health information systems.

With RBAC, access decisions are based on users' role(s) within the organization. Access rights are grouped by role name, and access is restricted to users authorized to assume the associated role. In the health sector, users should have roles based on superior profession combined with ward belonging, rosters and patient relations for the purpose of dynamic roles. By adding context-based constraints, such as rosters, to RBAC, access control is determined dynamically based upon the current context of the request, rather than just the role the user holds.

When integrating health information systems access control becomes even more complex than described above. A person may be registered as a user in two different systems. When these systems are integrated the user should be able to access information in both systems. Keeping track of one person, with separate userIDs in each subsystem, is a challenge. Making sure that every user can access all the information he is allowed to access - no more and no less - is a complex task.

Access control is usually implemented by means of two security mechanisms; authentication and authorization. Authentication is concerned with verifying that the initiator of a request has the identity which he claims to have. Authorization determines whether the given identity is allowed to access a resource or not. Authentication can be achieved with digital signatures, while authorization can be ensured by use of roles.

15.2 Assets

This section presents several definitions, standards, strategies, laws and regulations, role-cards and patterns relevant for integration of health information systems. Information sources specially relevant for information security are emphasized.

15.2.1 Dictionary

To avoid semantic misunderstandings, the following provides definitions for key concepts used in this report:

- General health care concepts are presented and defined in Table 15.1.
- Information security related concepts are presented and defined in Table 15.2.
- Architecture related concepts are presented and defined in Table 15.3.

Concept	Definition
CAVE	Latin: avoid. Information about medicine allergy, other allergies, etc.
Doctor	A person who is authorized and licensed as a doctor.
Electronic Observation Chart (EOC)	An observation chart in which information is stored electronically and can be retrieved and reused by means of suitable software. An EOC is a part of the electronic patient record (EPR).
Electronic Patient Record (EPR)	A patient record in which information is stored electronically and can be retrieved and reused by means of suitable software [36].
Enrolled nurse	Health personnel who assists in the nursing care under the direction and supervision of a registered nurse.
EPR-system	A health information system with focus on document handling for medical judgments, descriptions and conclusions.
Health information	Both medical and administrative information about a patient.
Health information system	An information system that electronically stores health information.
Health personnel	Personnel with an authorization or a license, personnel in the health services or in pharmacies who perform acts as mentioned in §3, or pupils and students who in training as health personnel perform acts as mentioned in §3 in the Health Personnel Act [36].
Lifeline	A patient's medical history from birth to death shown as a time scale in the EOC-system.
Navigation caremap	A read-only, navigation window in CARDIAC's EOC-system.
Patient	A person that applies to the National Health Service about health care or a person that the National Health Service in individual cases give or provide health services to [58].
Patient chart form	A part of the EOC-system where users are allowed to register information.
Registered nurse	Health personnel who assists in patient care under the direction of a doctor.

Table 15.1: Definition of general health care concepts essential for the architectural description.

Concept	Definition
Access Control	Access control consists of authentication and authorization. It is protection of resources against unauthorized access; a process by which use of resources is regulated according to a security policy and is permitted by only authorized system entities according to that policy [19].
Auditing	A posteriori technique for detection of security violations or other suspicious events ensuring the traceability within the system [10].
Authentication	The process of determining that a user's claimed identity is legitimate [6].
Authorization	The process of giving someone permission to do or have something done [6].
Availability	Security ensuring that a service fulfils definite stability requirements such that relevant information is available when needed [7].
Confidentiality	Security ensuring that only authorized persons gain access to sensitive or classified information, and that the person in advance is validity identified and authenticated [7].
Digital signature	A construct that authenticates both the origin and contents of a message in a manner that is provable to a disinterested third party [10].
Integrity	Security ensuring that the information and information processing is complete, precise and valid, and a result of authorized and controlled activity [7].
Information security	Protection against violations of confidentiality, availability, integrity and non-repudiation for the information processed by the system and for the information in the system itself [7].
Permission	Permissions are authorizations to perform some action on the system [6].
Principle of least privilege	The principle of least privilege is the practice of selectively assigning permissions to users such that the user is given no more permission than is necessary to perform his job function [6].
Public key infrastructure (PKI)	PKI is a collective term for technology providing unique digital identities across networks, where digital identity is used for authentication, digital signing of information and secure communication [65].

Table 15.2 – continued from previous page	
Concept	**Definition**
Role/Role-based Access Control (RBAC)	A role is a set of transactions that one user or a set of users can perform within the context of an organization. RBAC is a mean for controlling user access through roles [23].
Secure communication	Secure transmission of information and mutual authentication of the entities participating in the transmission.
Security policy	A set of rules and practices that specify or regulate how a system or organization provides security services to protect resources [19].

Table 15.2: Definition of information security related concepts essential for the architectural description.

Concept	**Definition**
Architecture	The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution [16].
Architectural description	A collection of products to document an architecture [16].
Architectural description framework	Assistance for the development and documentation of the architecture.
Integration	The process by which software systems are combined into a functional whole.
Target system	The system that is the goal of the specific architectural description.
User	The person using the target system.

Table 15.3: Definition of architecture related concepts.

15.2.2 Standards

The purpose with standards is to ensure interoperability and compatibility between solutions based on the standards. This enables information exchange and integration.

This section identifies standards which are relevant for the architectural description of CARDIAC's EOC-system. Because the architectural description is based on both service-oriented and portal-oriented integration architecture, some security standards relevant for this type of integration are listed below [1]:

- W3C's **XML Signature specification** [64] aims to provide data integrity and authentication (both message and signer authentication) features, wrapped inside an XML format.

- W3C's **XML Encryption specification** [68] addresses the issue of data confidentiality using encryption techniques. Encrypted data is wrapped inside XML tags defined by the XML Encryption specification.

- OASIS' **WS-Security** [46] defines a mechanism for including confidentiality, integrity and single message authentication features within a SOAP message. WS-Security makes use of the XML Signature and XML Encryption specifications and defines how to include digital signatures, message digests, and encrypted data in a SOAP message.

- OASIS' **Security Assertion Markup Language (SAML)** provides a means for partner systems to share user authentication and authorization information. This is essentially the single sign-on feature. With the advent of SAML, authentication information can be wrapped inside XML in a standard way, so that cookies in HTTP communication are not needed and interoperable single sign-on can be achieved.

- OASIS' **eXtensible Access Control Markup Language (XACML)** makes it possible to express access control policies in XML. XACML defines a vocabulary to specify subjects, rights, objects and conditions.

In addition to these security standards, the architectural description is based on the following standards for health informatics and electronic patient records:

- **KITH's EPR-standard.** KITH, a centre of competence for ICT in the National Health Service, has developed a standard for EPRs. The EPR-standard includes fundamental principles for access control in health information systems, and it is therefore essential for this architectural description [49].

- **CEN TC251**. European standardization of health informatics - Standardization in the field of health ICT to achieve compatibility and interoperability between independent systems and to enable modularity [48].
- **Health Level 7 (HL7)**. One of several ANSI[1] standards for health care specific data exchange between computer applications. The name "Health Level 7" refers to the top layer (Level 7) of the Open Systems Interconnection (OSI) layer protocol for the health environment [44].
- **ISO 18308: Health Informatics standard.** This standard technically specifies how to assemble and collate a set of clinical and technical requirements for an EPR architecture that supports using, sharing, and exchanging EPRs across different health sectors, different countries, and different models of health care delivery. It suggests requirements for the architecture but not the specifications of the architecture itself [45].
- **EDIFACT**. Electronic Data Interchange for Administration, Commerce and Transport. Used as a messaging standard in health care [66].
- **DICOM**. Picture standard that covers all areas related to the transportation, storage and display of medical pictures [66].

15.2.3 Strategies

For this architectural description, some strategies have to be followed. Because the target system is an EOC-system which integrates information from several different health information systems, integration strategies are of particular interest. The strategies presented in this section are formulated by HEMIT, and they are in force within the Health Region for Central Norway.

The most relevant strategies are:

- **HEMIT's integration strategy.** The purpose with this integration strategy is to integrate health services independently of the technological platform they are implemented on. The integration strategy is a requirement specification for IT-vendors when purchasing or developing new systems or IT-equipment. This strategy should be seen in combination with the IT architecture strategy.
- **HEMIT's IT architecture strategy.** The IT architecture strategy documents that the Health Region for Central Norway has chosen a service-oriented architecture model, which all future solutions shall be realized on. The IT architecture strategy requests all systems to

[1] American National Standards Institute

provide its functionality as services (Web services). Microsoft .NET is chosen as basis technology for the implementation of a service-oriented architecture model.

15.2.4 Laws and regulations

When developing information systems for electronic processing of sensitive health information, it is important that the information processing is in accordance with prevailing laws and regulations within the area.

The legal basis for patient records, and implicit for EOCs, is constituted of the following laws and regulations:

- **Personal Data Act**[2] [40] is supposed to protect natural persons from violation of their right to privacy through the processing of personal data. The act applies for the processing of personal data wholly or partly by automatic means.

- **Personal Data Regulations**[3] [39] are determined under the provision of the Personal Data Act, and they give regulations about information security when personal data is processed.

- **The Personal Health Data Filling System Act**[4] [37] is a special law in proportion to the Personal Data Act. It deals with the employment of general decisions about patient records in the Personal Data Act and contains a number of rules directly relevant for EPRs.

- **The Health Personnel Act**[5] [36] deals with health personnel's duties and responsibility in connection with their work. This includes relations attached to client confidentiality, the right to information, the duty to report and the documentation requirement.

- **Patient Record Regulations**[6] [43] are determined under the provisions of i.a. the Health Personnel Act, and they give further rules about the contents of a patient record, the work with patient records and access to the information in the patient record.

These laws and regulations primarily have an influence on the architectural description by demanding a certain level of information security, which the technical solutions have to support and ensure. Laws and regulations presented here have been the basis for choosing the security concerns described in Section 15.1.

[2]Norwegian: Personopplysningsloven
[3]Norwegian: Personopplysningsforskriften
[4]Norwegian: Helseregisterloven
[5]Norwegian: Helsepersonelloven
[6]Norwegian: Pasientjournalforskriften

15.2.5 Role-cards

Establishment of roles is an important aspect with access control in health information systems. As mentioned in MAFIIA/RBAC, *role-cards* can be helpful in the process of identifying roles for users of the system. The concept of role-cards is based on the UML use case collaboration process from the book *UML Distilled* [8] and the Class-Responsibility-Collaboration (CRC) card technique, which is explained in detail in [61].

The most important users of the EOC-system are doctors, registered nurses and enrolled nurses. Figure 15.1 shows a role-card which describes the role of a doctor, Figure 15.2 describes the role of a registered nurse, while Figure 15.3 describes the role of an enrolled nurse.

Role id	**Role name**	**Inherits from**
2	Doctor	Health Personnel

Role description
A Doctor is responsible for examination of the patient. A Doctor is allowed to prescribe medication, write requisitions and referrals. A Doctor may also give the patient medicine, observe the patient, perform measurements, in addition to follow up prescriptions, requisitions and referrals.

A Doctor needs read, write and edit access to the EOC-system. A Doctor also needs access to the Medication system and to RoS when respectively writing prescriptions and requisitions.

Figure 15.1: Role-card for the role of a doctor.

Role id	**Role name**	**Inherits from**
3	Registered Nurse	Health Personnel

Role description
A Registered Nurse is responsible for the planning, coordination and instruction of the patient care. A Registered Nurse also nurses the patient, which may include giving the patient medicine, observe the patient, perform measurements, prescribe non-prescription medication and following up requisitions and referrals.

A Registered Nurse needs both read, write and edit access to the EOC-system. A Registered Nurse also needs access to the Medication system for prescribing non-prescription medication.

Figure 15.2: Role-card for the role of a registered nurse.

Role id	Role name	Inherits from
4	Enrolled Nurse	Health Personnel

Role description
An Enrolled Nurse is allowed to observe the patient and perform measurements.

An Enrolled Nurse needs both read, write and edit access to the EOC-system.

Figure 15.3: Role-card for the role of an enrolled nurse.

15.2.6 Patterns

A (software) pattern is a model of a standardized solution on a problem/challenge which often occurs during software development. This section identifies relevant patterns, and gives a brief introduction of them, while they are described in more detail in Appendix B.

The following patterns are of current interest for the system which shall be developed based on this architectural description:

Adapter

Adapter pattern converts interfaces of existing resources to the interface the client expects. Adapters let classes work together that could not otherwise because of incompatible interfaces [9].

Façade

Façade pattern provides a unified interface to a set of interfaces in a subsystem. It constructs a higher-level interface, and the clients only access the subsystem through this façade interface. The façade interface uses the original interfaces of the subsystem [9].

Single access point

Single access point pattern provides a security module and a way to log in to the system. It prevents the use of multiple entries and back doors by providing only one single choke point [30].

Check point

Check point pattern is applicable when authentication and authorization of system users are necessary. Authentication and authorization checks have to be done according to the given security policy, i.e. a set of rules that states which actions are permitted and which actions are prohibited. When enforcing the security policy, the check points can respond to violations.

Role-based access control

Role-based access control pattern is a mean for controlling user access through roles that individual users have as part of an organization.

15.3 Reference Architecture

A reference architecture is a high-level, generic architecture which is used as the basis in development of concrete system architectures.

The reference architecture for this architectural description is based on the reference architecture of MAFIIA for IIS, as described in Section 10.4. As shown in Figure 15.4, a reference architecture of this type connects the environment interface of the target system towards the user interface of the environment systems.

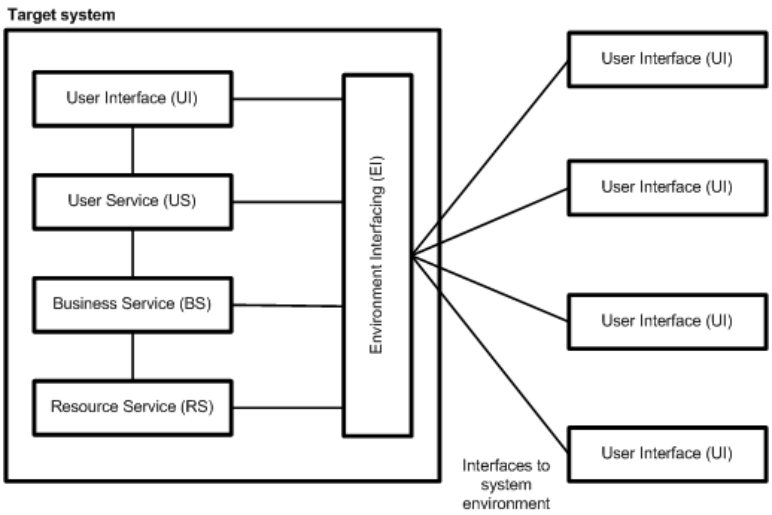


Figure 15.4: IIS specific reference architecture.

The target system in this thesis is based on a combination of service-oriented and portal-oriented integration architecture. These integration architectures are fitted into the MAFIIA reference architecture in Figure 15.5. The reference architecture for a portal-oriented system integration

is shown in the upper-right corner, while the reference architecture for a service-oriented system integration is shown in the upper-left corner. These two integration architectures are mixed together and fitted into MAFIIA's reference architecture at the bottom of the figure.

Figure 15.5 hints on the use of the Adapter pattern, which is described in Appendix B. Adapters may be included in the business service or the resource service tier.

Also, Figure 15.5 has been extended with environment systems, and the use of an Application Server in the Environment Interfacing to the environment systems. The shaded rectangles with dashed lines inside the Application Server are examples of the functionality an application server may provide.

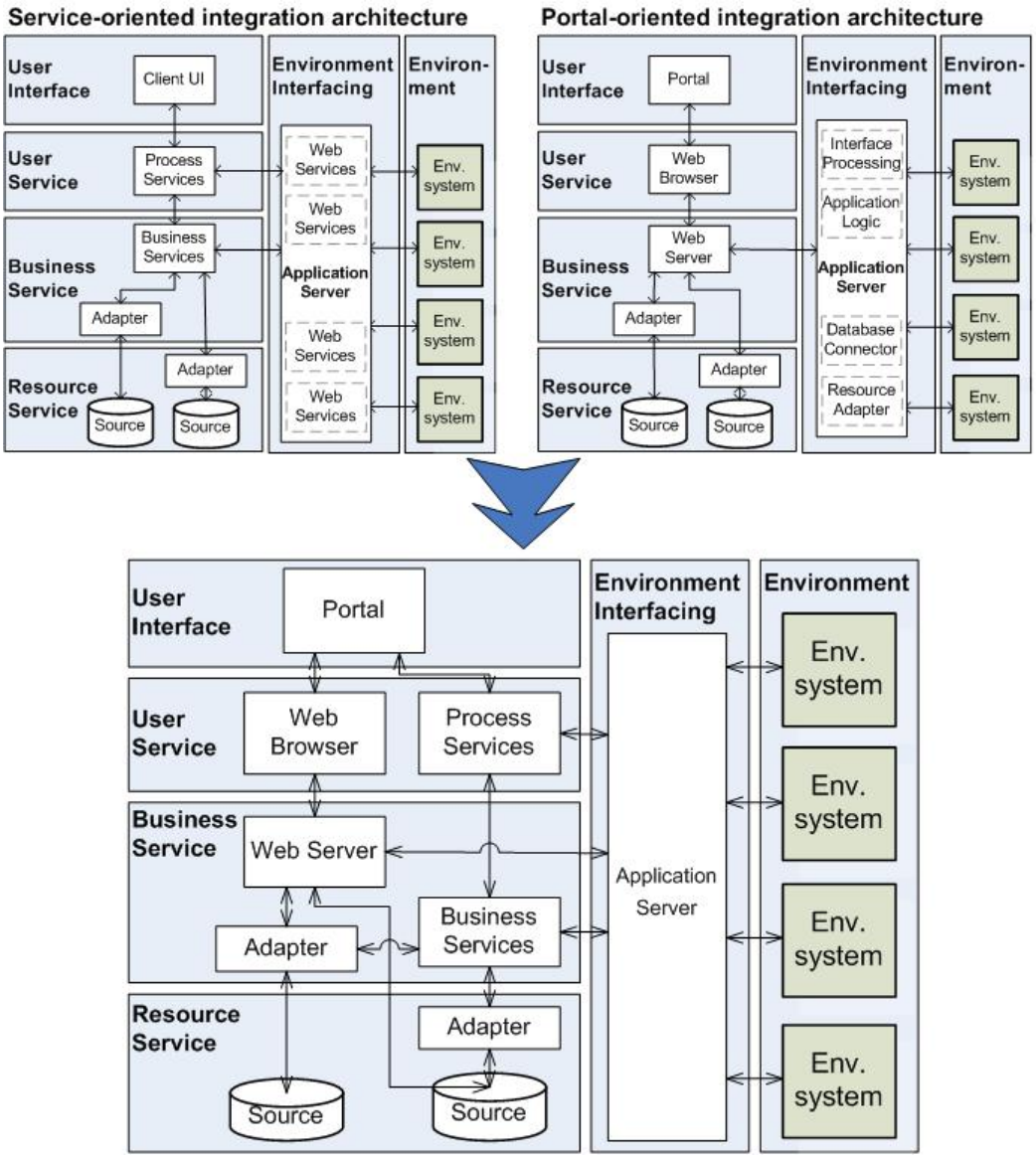


Figure 15.5: Reference architecture for a combination of service-oriented and portal-oriented system integration.

15.4 Stakeholders and roles

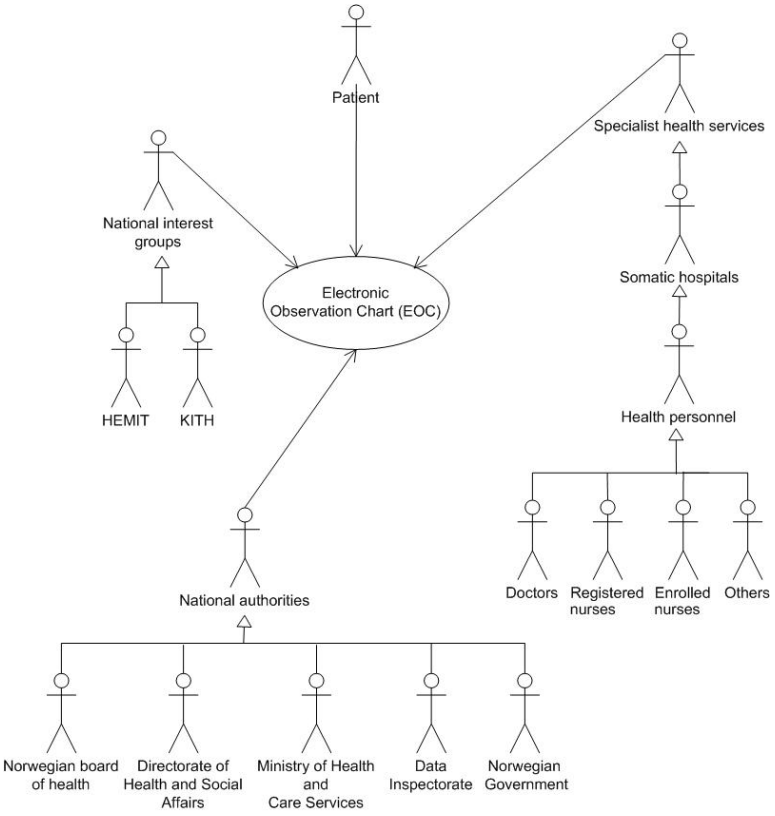


Figure 15.6: Overview of stakeholders for the EOC-system.

Figure 15.6 shows the most important stakeholders in relation to the EOC-system. Four groups of stakeholders are identified, and each group consists of a number of stakeholders with different roles in relation to the EOC:

- **National interest groups**
  - HEMIT is a special IT-unit which gathers all IT-departments within the Health Region for Central Norway.
  - KITH is a centre of competence for ICT in the National Health Service.
- **National authorities**
  - The Norwegian Government formulates laws and regulations with impact on the EOC.

- The Norwegian Board of Health [7] is an independent supervision authority, with responsibility for general supervision of health and social services in the country [57].
- The Ministry of Health and Care Services [8] holds the superior responsibility concerning health policy, public health, health care services and health legislation in Norway [69].
- The Directorate of Health and Social Affairs [9] is an administrative and competence body which contributes to the implementation of national politic within the health and social sector. It is a advisory service for central authorities, health enterprises etc [59].
- The Data Inspectorate [10] is an independent administrative body under the Norwegian Ministry of Labor and Government Administration. It was set up in 1980 to ensure enforcement of the Data Register Act of 1978, now made obsolete by the commencement of the Personal Data Act of 2000 [56].

- **Specialist health services**
  - Somatic hospitals are hospitals that investigate and treat patients with physical diseases or injuries.
  - Health personnel relevant for the EOC-system are:
    - Doctors
    - Enrolled Nurses
    - Registered Nurses
    - Others

- **Patient**

15.5 Modeling language

In the generic MAFIIA it is mandatory to use Unified Modeling Language (UML). MAFIIA/RBAC suggests use of a special security related profile, UMLsec, in addition to UML. This architectural description makes use of both UML and UMLsec. Therefore, it is presumed that the reader has basic knowledge of UML version 1.4 and has a certain knowledge of UMLsec. A short introduction to UML and UMLsec is given in Appendix C. For more information on UML, see [8] and for UMLsec read [12].

[7] Norwegian: Statens helsetilsyn. In short Helsetilsynet
[8] Norwegian: Helse- og omsorgsdepartementet
[9] Norwegian: Sosial- og helsedirektoratet
[10] Norwegian: Datatilsynet

15.6 Summary

This chapter has detailed some of the concepts applied in the architectural description for CARDIAC's EOC-system. Certain concerns and assets relevant for the development of the EOC-system are identified. The security mechanisms digital signing, secure communication, auditing and access control are important aspects of this architectural description. When developing the architectural description, information sources from the National Health Service, HEMIT, KITH and others are used. This chapter also gives an example for how a service-oriented and portal-oriented integration architecture may be mapped to the reference architecture. Stakeholders and roles are identified, and UML version 1.4 is chosen as formal description modeling language for the models in the following chapters.

Chapter 16

Context Viewpoint

The context viewpoint documents the target system's environment. This includes identification of stakeholders and their relation to the target system. Context viewpoint primary consists of three different models used to describe different parts of the environment:

- Business Aspects Model documents information, stakeholders and processes related to the target system.
- Environment System Model identifies other systems in the target system's environment.
- Business to System Mapping Model describes what the target system shall realize based on the two other models.

16.1 Business Aspects Model

Model	**Business Aspects Model**
Purpose	Shall document any business related concern that will increase the understanding of what problems the target system shall solve, or what functionality it shall implement.
Input	Customer supplied information (i.e. requirements to target system, business related information).
Output	UML class diagram, UML use case diagram, UML collaboration diagram, Textual description.

Table 16.1: Business Aspects Model as described in the generic MAFIIA.

As mentioned in Section 15.1, the EOC-system shall help in reducing the redundancy of information and preventing manual typing errors. The overall goal of the target system is to provide:

- better quality of patient care
- increased efficiency of provided health services within the hospital
- a common and comprehensive information base

The above mentioned aspects will be implemented by functionality such as:

- single sign-on against relevant health information systems.
- shortcuts to several health information systems.
- automatic data acquisition from relevant medical technical equipments and health information systems.
- flexible observation charts for different wards.
- a time-dependent overview of the patient care.

These functions will be described more thoroughly throughout this architectural description.

Non of the recommended outputs for the Business Aspects Model were suitable for documenting the problems that the target system should solve. Therefore, a UML activity diagram is the chosen output.

Figure 16.1 shows an activity diagram highlighting important work processes for health personnel related to the EOC during patient care. As mentioned earlier, the EOC is mostly used by registered nurses, enrolled nurses and doctors. A doctor is responsible for examining the patient and deciding what kind of medical treatment the patient shall receive during his stay in the hospital. On the other hand, registered nurses and enrolled nurses are responsible for carrying out most of the suggested medical treatment.

More precisely, regarding CARDIAC's EOC-system patient care should start when a hospital ***receives the patient***. If the patient has been hospitalized before, the health personnel should be able to ***retrieve the navigation caremap*** from the EOC-system and ***view the lifeline***, which will give an overview of the patient's medical history. If the patient has not been hospitalized before, patient information, such as biographical data and CAVE, should be retrieved and presented in the EOC-system. Health personnel should be able to ***configure observation charts*** according to the needs of the particular ward or patient.

During the ***patient care***, health personnel will have to follow some ***quality procedures***. They will have to digitally sign that they have followed the proposed quality procedure, or if the entire procedure or just parts of it is omitted.

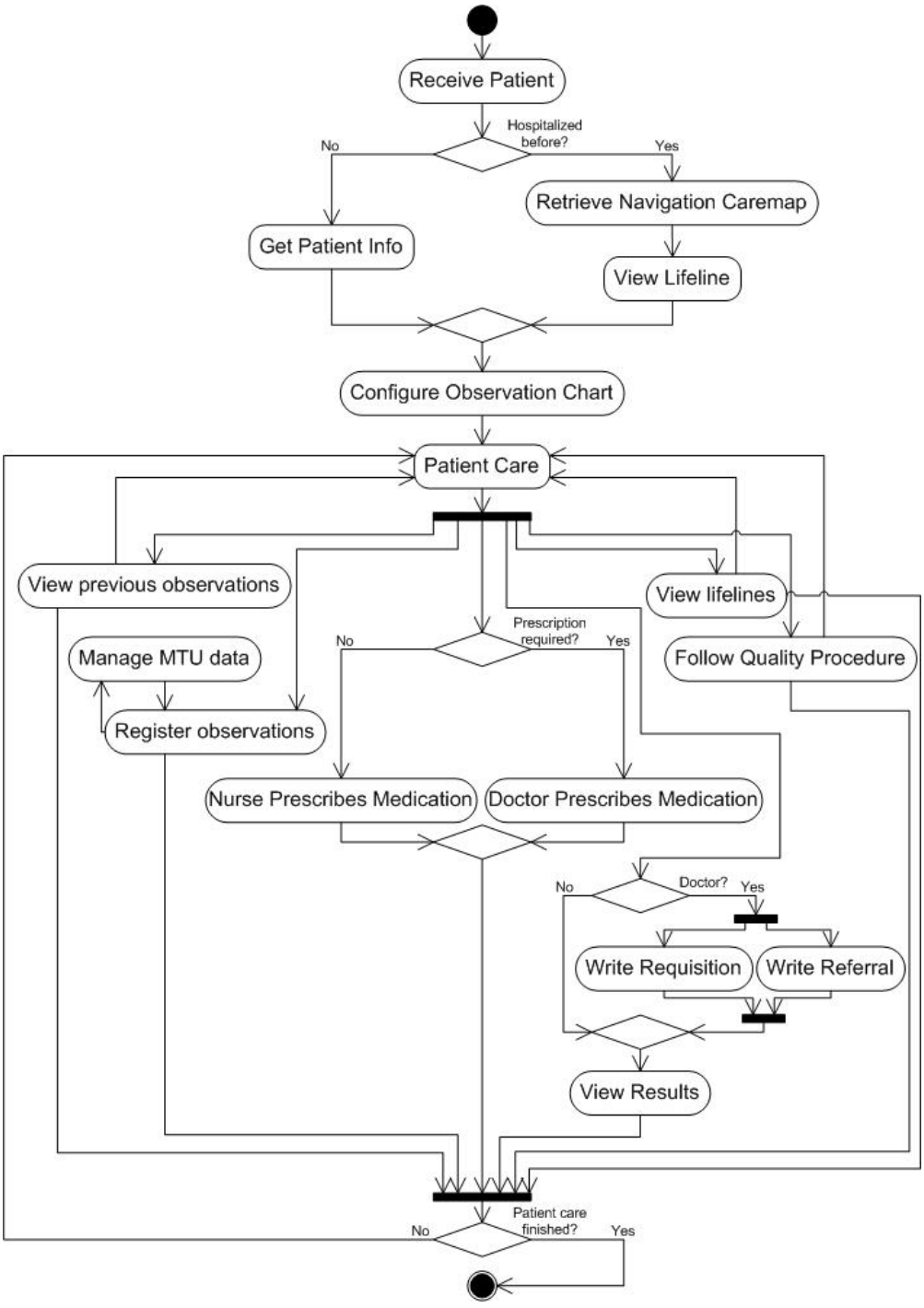


Figure 16.1: Business Aspects Model showing the work processes in an EOC system.

Patient care includes both nursing and examining the patient, which again includes registration of observations, prescription of medication, writing requisitions and referrals, in addition to viewing results, previous observations and lifelines.

Both registered nurses, enrolled nurses and doctors should have the possibility to ***register observations*** in the patient chart form-part of the EOC-system, but will mostly be done by registered or enrolled nurses. Registration of observations may involve manual registration of data from medical technical equipment, shown in Figure 16.1 as ***manage MTU data***. All registrations will have to be signed digitally by the health personnel that registered them.

Usually, doctors should be able to ***prescribe medication*** in the EOC-system, but some registered nurses may have medical qualifications to do this. A registered nurse will only be allowed to prescribe non-prescription medication, shown as ***nurse prescribes medication*** in the figure. Health personnel should digitally sign all prescriptions of medication.

Only doctors should be allowed to ***write requisitions*** and to ***write referrals***, while all health personnel related to the EOC should be allowed to ***view the results***. Requisitions result in laboratory results, X-rays, etc., while referrals result in epicrisis. Both requisitions and referrals will require a digital signature from the doctor who wrote them.

Patient care is a continuous process. This is shown at the bottom of Figure 16.1 as a branch labeled *"Patient care finished?"*. If patient care is not finished, health personnel will carry on with their work tasks. If the patient care is finished, the patient will usually be transferred to another ward or be discharged.

16.2 Environment Systems Model

Model	**Environment Systems Model**
Purpose	Shall document other technical systems (environment systems) that will be involved in the implementation of the Business Aspects Model, or influences the operation of the target system.
Input	Customer supplied information (i.e. target system requirement specifications, environment system documentation)
Output	UML sequence diagram, UML use case diagram, UML collaboration diagram, Textual description.

Table 16.2: Description of the Environment System Model as described in the generic MAFIIA.

Recall that CARDIAC's EOC-system shall retrieve information from other health information systems. This information retrieval implies that CARDIAC's EOC-system is integrated with these health information systems. Such integration contributes to a common information base and improved continuity of patient care. CARDIAC's EOC-system shall give health personnel an overview of the patient condition, and it shall serve as the main navigation system for health personnel during the period of patient care. The probability for incorrect registrations is also reduced because manual typing of information is reduced.

Figure 16.2 identifies health information systems that constitute a part of CARDIAC's EOC-system. This is shown in a package diagram where the identified systems are shown as packages. The EOC-system is dependent on an electronic patient record system (EPR-system), a patient administrative system (PAS), a requisition and response system (RoS [1]), a medication system, a quality procedure system (EQS [2]) and medical technical equipment (MTU [3]).

As already mentioned, CARDIAC has developed IMATIS Medical Data Acquisition System, which retrieves data from MTU. Thus, it is important to remember that the MTU-package actually represents the IMATIS Medical Data Acquisition System. IMATIS Medical Data Acquisition System is hereby referred to as MTU throughout this architectural description.

Package diagrams are suitable when documenting environment systems, and this kind of diagrams is therefore chosen as output for the Environment System Model although it is not recommended in the generic MAFIIA.

[1] Norwegian: Rekvisisjon og Svar
[2] Extend Quality System
[3] Norwegian: Medisinsk Teknisk Utstyr

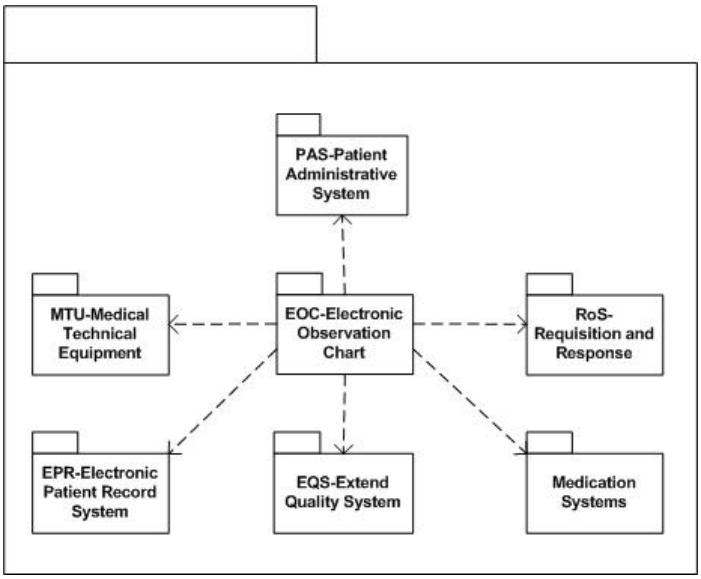


Figure 16.2: Health information systems integrated in the EOC-system.

16.3 Business to System Mapping Model

Model	**Business to System Mapping Model**
Purpose	Shall document what parts of the documented Business Aspects Model are mapped to technical solutions constituted by environment systems and the target system, and how the different parts of the Business Aspects Model are mapped to the different involved systems.
Input	Customer supplied information (i.e. target system requirement specifications, requirements/needs related to the business aspects).
Output	UML use case diagram, Textual description.

Table 16.3: Description of the Business to System Mapping Model following the format from the generic MAFIIA.

Table 16.4 combines the processes from the Business Aspects Model with the environment systems, which were identified in the Environment Systems Model, and shows how they are mapped together.

Health information system	Process
EPR-system	Provides information about the patients earlier medical treatments and information about allergies, i.e. CAVE.
EOC	Gives a possibility to configure an EOC on the basis of predefined templates. Takes as input all observations and registrations which are performed and signed by health personnel. Generates lifelines on the basis of data which is collected from other health information systems.
PAS	Provides biographical data about the patient, e.g. name, address, date of birth and social service number.
RoS	Handles requisitions from the doctor. Provides response in the form of lab results, test results, pictures, etc.
Medication system	Handles medication prescription which is mainly performed by a doctor.
EQS	Improves the quality of patient treatment by providing the health personnel with procedures for best practice treatment of patients.
MTU	Information from some of the instruments, sensors and monitors is captured automatically in the system, while other information must be registered in the EOC-system manually .

Table 16.4: Information and processes in the systems which comprise an EOC-system.

Based on the Business Aspects Model relevant users of CARDIAC's EOC-system are identified in Figure 16.3. According to CARDIAC, the patient is left out from the diagram because a patient never shall be alone when looking into his EOC. A registered nurse, enrolled nurse or a doctor should always be present in order to explain the meaning of test results and observations for the patient and in order to avoid misunderstandings. The nurse or the doctor should log in to the system and show relevant information for the patient.

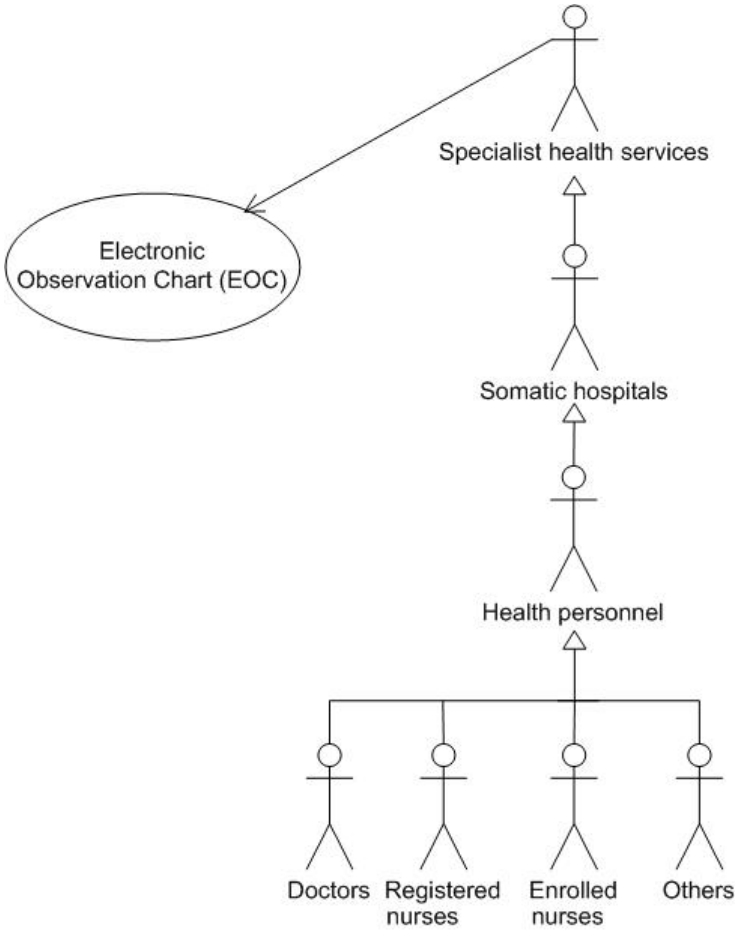


Figure 16.3: Stakeholders which are included in the process of making and processing an EOC.

16.4 Summary

The context viewpoint documents which work processes within a hospital the EOC-system shall support and which health information systems it shall be integrated with. Registered nurses, enrolled nurses and doctors are also identified as users of the EOC-system.

Chapter 17

Requirement Viewpoint

The requirement viewpoint specifies requirements to the target system. The main basis for the requirements is the context viewpoint documentation, which implicitly represents requirements to CARDIAC's EOC-system.

The requirement viewpoint consists of three models as specified in the following sections:

- Requirement Model identifies requirements related to the target system.
- Target System Interface Model concerns the target system's interfacing to its environment.
- Target Organization Security Policy Model describes the target organization's security policy.

17.1 Requirement Model

Model	**Requirement Model**
Purpose	Shall be complete in identifying and eventually specifying all relevant requirements to the target system where a requirement shall be verified.
Input	Context viewpoint.
Output	UML sequence diagram, UML collaboration diagram, UML and UMLsec use case, UMLsec class diagram, Textual description.

Table 17.1: Requirement Model as described in the generic MAFIIA and MAFIIA/RBAC.

Requirements specified in the Requirement Model are listed in Table 17.2.

REQ. ID	REQUIREMENT IDENTIFICATIONS
R1	**Viewpoint related requirements**
R1.1	**Context viewpoint**
R1.1.1	The context of the target system shall be within somatic hospitals primarily in the Health Region for Central Norway.
R1.2	**Requirement viewpoint**
R1.2.1	Target system shall fulfil the requirements listed in this table.
R1.3	**Component viewpoint**
R1.3.1	Target system shall have the possibility to be decomposed into subsystems, components and information objects.
R1.3.2	Target system shall be based on both service-oriented and portal-oriented system integration architecture.
R1.4	**Distribution viewpoint**
R1.4.1	The distribution of components must be transparent for the users.
R1.5	**Realization viewpoint**
R1.5.1	Target system shall be realized on the technological platform required by HEMIT.
R1.5.2	Target system shall be realized on CARDIAC's IMATIS Platform.
R2	**Concern related requirements**
R2.1	**Application functionality concerns**
R2.1.1	Target system shall retrieve relevant information from the environment systems.
R2.1.2	Target system shall present the retrieved information to the users through a common user interface, a portal.
R2.1.3	Target system shall present all user functionality through the portal.
R2.1.4	Target system shall support automatic synchronization of information.
R2.1.5	Target system shall support writing of requisitions.
R2.1.6	Target system shall support writing of referrals.

Table 17.2 – continued from previous page

REQ. ID	REQUIREMENT IDENTIFICATIONS
R2.1.7	Target system shall support prescription of medication.
R2.1.8	Target system shall support reading of information.
R2.1.9	Target system shall support registration of information.
R2.1.10	Target system shall support editing of information within the system.
R2.1.11	Target system shall support notification.
R2.1.12	All users shall have the possibility to choose the amount of information presented through the target system's zoom in and zoom out functionality.
R2.2	**Quality related concerns**
R2.2.1	*Security* obtained by means of digital signing, secure communication, auditing, authentication and authorization.
R2.2.1.1	*Digital signing*
R2.2.1.1.1	Each user shall have an unique digital identity.
R2.2.1.1.2	Target system shall support digital signatures.
R2.2.1.1.3	Target system shall demand users to digitally sign all user registrations, followed/omitted quality procedures, requisitions, referrals and prescriptions.
R2.2.1.2	*Secure communication*
R2.2.1.2.1	Target system shall enforce mutual authentication of entities (systems, system resources, etc.) participating in the communication.
R2.2.1.2.2	Target system shall run within a hospital, i.e. within a hospital network with secure zones.
R2.2.1.2.3	Target system shall provide secure transmission of information, i.e. encryption, between entities participating in the communication.
R2.2.1.2.4	Target system shall support strong encryption, i.e. asymmetric encryption with minimum 1024 bits key size or symmetric encryption with minimum 128 bits key size.
R2.2.1.3	*Auditing*
R2.2.1.3.1	Auditing shall be performed on the service-level. *Comment:* The service which receives the request is responsible for auditing.
R2.2.1.3.2	Audit log shall be automatic and computer generated.

Table 17.2 – continued from previous page

REQ. ID	REQUIREMENT IDENTIFICATIONS
R2.2.1.3.3	Audit log shall include a time stamp (date/time), full name or userID of the user initiating the action/operation, the action/operation performed and the object that the user performs the action/operation on.
R2.2.1.3.4	Audit log shall record all user logins.
R2.2.1.3.5	Auditing shall be performed during retrieving of information.
R2.2.1.3.6	Auditing shall be performed during registration of information.
R2.2.1.3.7	Auditing shall be performed during editing of information.
R2.2.1.3.8	Audit logs shall be available for review and copying by regulatory authority.
R2.2.1.3.9	Audit logs shall be available for review and copying by a system administrator.
R2.2.1.3.10	Audit logs shall not be available for modifications or deletions.
R2.2.1.3.11	Audit log reports shall be read-only reports.
R2.2.1.3.12	Audit log shall be stored in a secure place.
R2.2.1.3.13	Audit log shall minimum be stored in three months.
R2.2.1.3.14	Target system shall detect and record all attempted accesses that fail identification, authentication or authorization requirements.
R2.2.1.3.15	All failed attempted accesses shall be notified daily.
R2.2.1.4	*Access Control*
R2.2.1.4.1	Target system shall only provide user authentication through its portal.
R2.2.1.4.2	Target system shall provide two-factor user authentication, i.e. the user authenticates himself with something that he knows, such as username and password, in addition to something that he has, such as a token or digital certificate.
R2.2.1.4.3	If passwords are used, target system shall provide suitable password management. Strong passwords shall be used, i.e. a minimum number of characters in the username and password is required and the password shall be a combination of numbers and characters. Mandatory change of passwords is required at regular intervals.
R2.2.1.4.4	Target system shall support registration of new users.

Table 17.2 – continued from previous page

REQ. ID	REQUIREMENT IDENTIFICATIONS
R2.2.1.4.5	Target system shall identify all of its users before allowing them to use its capabilities.
R2.2.1.4.6	Target system shall re-verify the identity of all of its users before allowing them to update their user profile.
R2.2.1.4.7	Target system shall identify all of its clients before allowing them to use its capabilities.
R2.2.1.4.8	Target system shall provide single sign-on, i.e. an individual user is not required to identify himself multiple times during a single session.
R2.2.1.4.9	Target system shall allow each user to obtain access to all of his own personal settings.
R2.2.1.4.10	Target system shall not allow any user to access any personal information of any other user.
R2.2.1.4.11	Target system shall only allow authorized users to access information and system resources.
R2.2.1.4.12	Target system shall provide complete, updated, correct and relevant information to those who have a legitimate need for it.
R2.2.1.4.13	Target system shall ensure the access control mechanism within each of the underlying systems.
R2.2.1.4.14	Target system shall provide for a combination of role-based and context-based access control.
R2.2.1.4.15	Target system shall support dynamic role assignment.
R2.2.1.4.16	Target system shall support delegation of access rights, i.e. one user can delegate access rights to another user.
R2.2.1.4.17	Target system shall automatically log the user out of the system after a certain time of inactivity.
R2.2.1.5	*Security Threats*
R2.2.1.5.1	Target system shall prevent and detect unauthorized logins.
R2.2.1.5.2	Target system shall prevent and detect unauthorized use of the system, i.e. illegitimate reads, writes or edits.
R2.2.1.5.3	Target system shall prevent tapping of communication lines, e.g. security attacks releasing message contents or modification of messages.

Table 17.2 – continued from previous page

REQ. ID	REQUIREMENT IDENTIFICATIONS
R3	**Model assets related requirements**
R3.1	**Standards**
R3.1.1	Follow KITH's EPR-standard.
R3.1.2	Use UMLsec to emphasize security related elements in models.
R3.1.3	Follow the Common Criteria - RBAC protection profile [26].
R3.2	**Strategies**
R3.2.1	Follow HEMIT's integration strategy.
R3.2.2	Follow HEMIT's IT architecture strategy.
R3.3	**Pattern**
R3.3.1	Use of Adapter pattern.
R3.3.2	Use of Façade pattern.
R3.3.3	Use of Single access point pattern.
R3.3.4	Use of Check point pattern.
R3.3.5	Use of Role-based access control pattern.
R4	**Reference architecture related requirements**
R4.1	**Environments**
R4.1.1	*Environment systems*
R4.1.1.1	The target system shall integrate the following systems: • Electronic Patient Record System (EPR-system) • Patient Administrative System (PAS) • Requisition and Response (RoS) • Medication system • Extend Quality System (EQS) • Medical Technical Equipment (MTU)
R4.1.2	*Environment systems interface*
R4.1.2.1	The communication between the target system and its environment systems shall be transaction based.

Table 17.2 – continued from previous page

REQ. ID	REQUIREMENT IDENTIFICATIONS
R4.2	**Target System**
R4.2.1	*Target System Configuration*
R4.2.1.1	
R4.2.2	*Target System Interfaces*
R4.2.2.1	
R4.2.3	*Functionality and data in target system components*
R4.2.3.1	The user interface functionality includes a portal.
R4.2.3.2	The portal shall provide the following (user service) functionalities: - role-based access control. - easily accessible shortcuts to the subsystems: EPR-system, PAS, RoS, Medication, EQS and MTU. - presentation of the navigation caremap and the patient chart form.
R4.2.3.3	Target system shall retrieve, filter and present patients' biographical data from PAS in the navigation caremap.
R4.2.3.4	Target system shall retrieve, filter and present information about patient diagnosis from PAS in the navigation caremap.
R4.2.3.5	Target system shall retrieve, filter and present information about patient care incidences from PAS in the navigation caremap.
R4.2.3.6	Target system shall retrieve and present CAVE-information from the EPR-system in the navigation caremap.
R4.2.3.7	Target system shall retrieve documents from the EPR-system for the particular time interval the patient chart form shows.
R4.2.3.8	Target system shall provide functionality for reading of the retrieved documents from the EPR-system.
R4.2.3.9	Target system shall present documents retrieved from the EPR-system as icons in the navigation caremap.
R4.2.3.10	Target system shall retrieve and present laboratory results, X-ray results and test results from RoS in the navigation caremap.
R4.2.3.11	Target system shall provide functionality for writing of requisitions to RoS.

Table 17.2 – continued from previous page	
REQ. ID	**REQUIREMENT IDENTIFICATIONS**
R4.2.3.12	Target system shall retrieve graphs, trend curves, observations and measurements from MTU.
R4.2.3.13	Target system shall show the retrieved graphs and trend curves in real time and historically in the navigation caremap.
R4.2.3.14	Target system shall support automatic registrations of observations and measurements from MTU in the patient chart form.
R4.2.3.15	Target system shall support entries in EQS.
R4.2.3.16	Target system shall support user registrations of observations and measurements in the patient chart form.
R4.2.3.17	Target system shall retrieve, filter and present the medication history of a patient from the Medication system in the navigation caremap.
R4.2.3.18	Target system shall provide functionality for prescription of medication to the Medication system.
R4.2.3.19	Target system shall retrieve and present information from the patient chart form in the navigation caremap.
R4.2.3.20	Target system shall have the possibility to generate and present graphical representations of the retrieved information from the environment systems in the navigation caremap.

Table 17.2: Requirement Model

17.2 Target System Interface Model

Model	**Target System Interface Model**
Purpose	To be a supplementary specification to the Requirement Model to obtain more complete and easier understandable specification of the target system's interfacing to its environments.
Input	Context viewpoint (Business to System Mapping Model), Requirement Model.
Output	UML sequence diagram, UML use case diagram, UML collaboration diagram.

Table 17.3: Target System Interface Model as described in the generic MAFIIA.

As described in the generic MAFIIA, Target System Interface Model is supposed to document the target system's interfacing to its environment by emphasizing *who* performs the *actions*, what *responses* are given and what functionality is included in this process. This is described further in the UML use case diagrams in Section 17.2.1.

As a supplement to the Requirement Model, this Target System Interface Model also identifies some security threats. Security threats are illustrated in the UML misuse case diagrams in Section 17.2.2.

17.2.1 Use cases

Figure 17.1 gives an overview of which actions a user of the EOC-system should be able to perform. The user will have to log in to the EOC-system before ***reading***, ***writing*** or ***editing*** information. Each action is shown in connection with some security mechanisms which have to be ensured during the performance of the action.

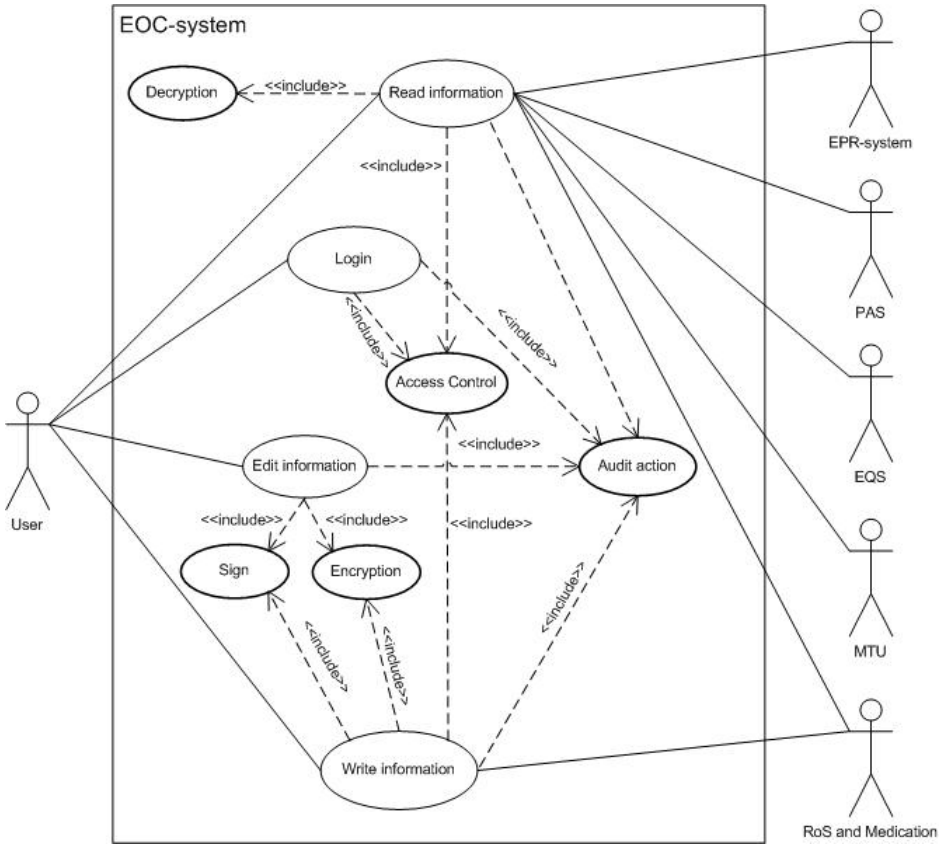


Figure 17.1: Use case showing the four general actions which may be performed in the EOC-system in connection with the security mechanisms.

More precisely, when a user logs in to the EOC-system, the EOC-system will have to check whether the ***login*** is valid or not. This check should be done in accordance to the ***access control*** mechanisms within the EOC-system and the environment systems. As stated in the Requirement Model, all logins shall be audited. This is shown by the ***Audit action*** use case in Figure 17.1.

When a read-action is performed, information stored in the EOC-system or the environment systems should be ***decrypted*** and presented to the user. The access control mechanism must ensure that the user only reads information he is authorized for. All read-actions should also be audited.

The user should be able to write and store information inside the EOC-system, i.e. observations and measurements. The user should also be allowed to write information towards RoS and the Medication system. Still, writing of information towards RoS and the Medication system will not imply any sending of information from the EOC-system to these systems. Instead, writing should be performed by opening the user interface from RoS or the Medication system in the EOC interface. Then the user should be allowed to write information directly into RoS or the Medication system.

Writing of information towards RoS implies writing requisitions, while writing towards the Medication system implies prescribing medication. In addition to these two write-actions, a user must be allowed to write referrals. So far there is no health information system handling referrals electronically, but writing referrals is visualized anyway, because it belongs to the group of work processes which should be performed by a doctor.

All write-actions in the EOC-system shall be ***access controlled***, ***signed***, ***audited*** and ***encrypted***, as shown in the Figure 17.1.

The user should not be allowed to delete or edit any information presented in the EOC-system, except when e.g. typing errors are discovered. Then editing, not deletion, of information will be allowed, but only if the changes are audited and signed. It is important to note that editing of information presupposes a decryption and a read-action. The edit-action itself is similar to the write-action and should be followed by an encryption.

The read-action, write-action, and edit-action will be shown more thoroughly in the following use case diagrams. Figure 17.2 shows what actions a doctor will be able to perform. Figure 17.3 shows the actions which a registered nurse should be allowed to perform in the EOC-system, while Figure 17.4 shows the actions should be performed by an enrolled nurse. The difference between the enrolled nurse and the registered nurse is that the enrolled nurse will not be allowed to prescribe any medication at all. The doctor will be the only one who is allowed to prescribe medication, but a registered nurse will be able to prescribe non-prescription medication, e.g. pain relieving medication.

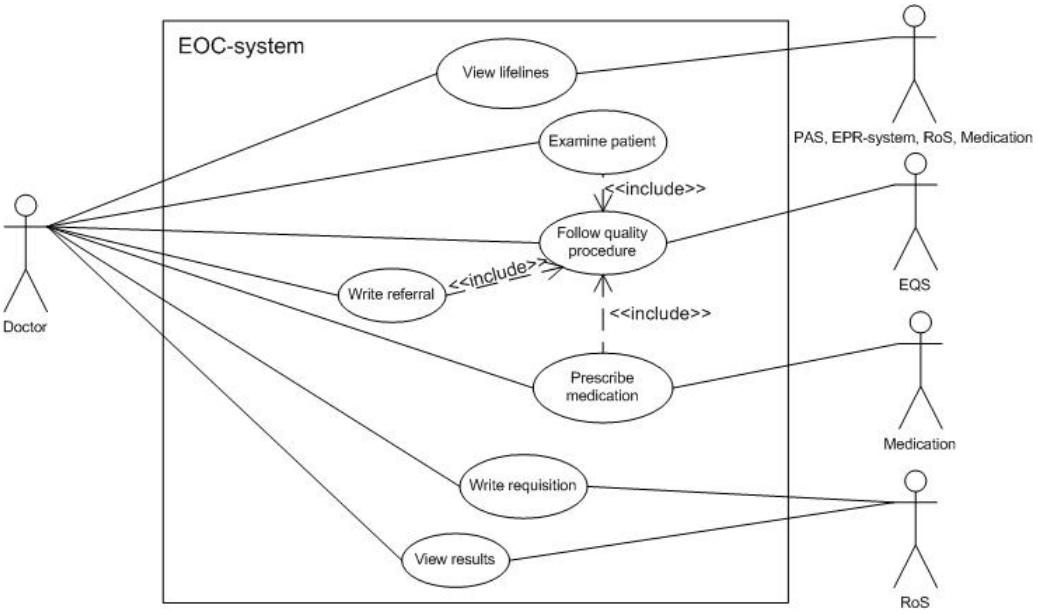


Figure 17.2: Use case showing the general actions performed by a doctor using the EOC-system.

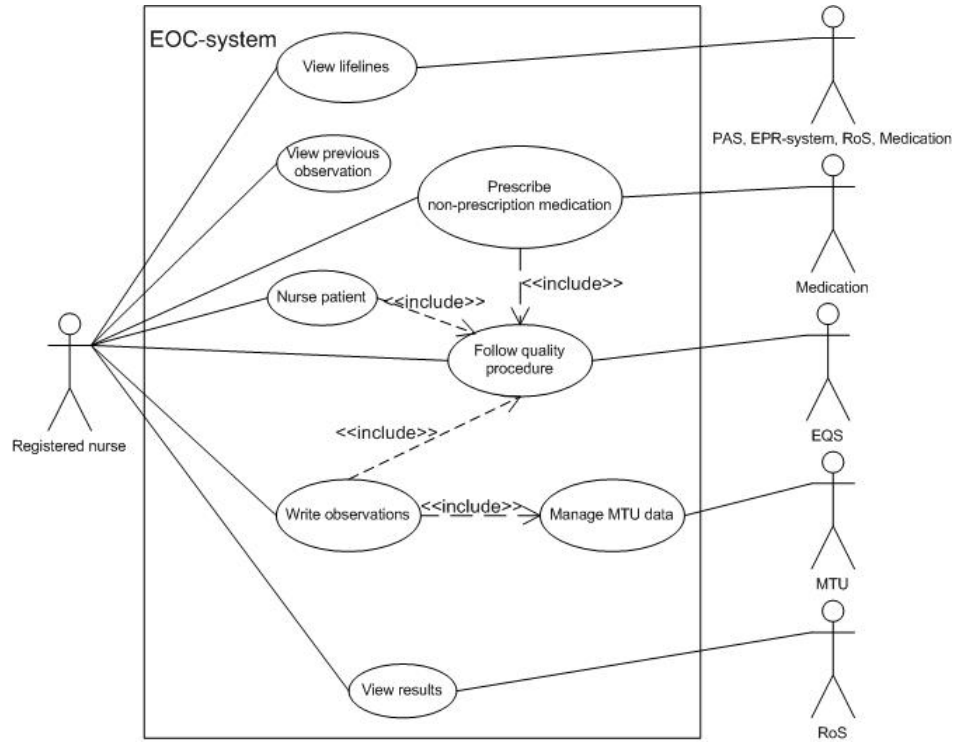


Figure 17.3: Use case showing the general actions performed by a registered nurse using the EOC-system.

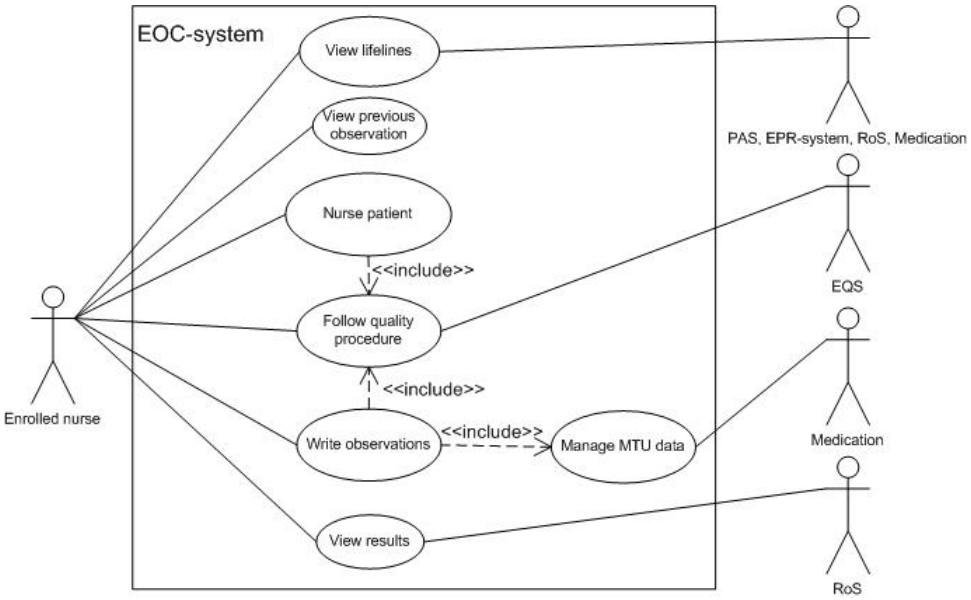


Figure 17.4: Use case showing the general actions performed by an enrolled nurse using the EOC-system.

Figure 17.5 shows a use case diagram documenting a typical read-action in the EOC-system, together with the security mechanisms which should be connected to such an operation. It is assumed that the user is logged in to the EOC-system before this use case is triggered. The EOC-system should generate a lifeline on the basis of information registered in PAS and present it in the navigation caremap. The first lifeline presented to the user will therefore just be an overview of all patient care incidences in the patient's medical history. By using the zoom in functionality, the user should be able to choose one patient care incidence and get a more detailed view of events, registrations, etc. for this particular incidence. When the user requests more information by using the zoom in functionality, the EOC-system will retrieve this information from the other health information systems which are listed in the diagram. The user shall also have the possibility to scroll forwards and backwards inside the lifeline, e.g. by moving one second, minute, hour, day, month or year per step.

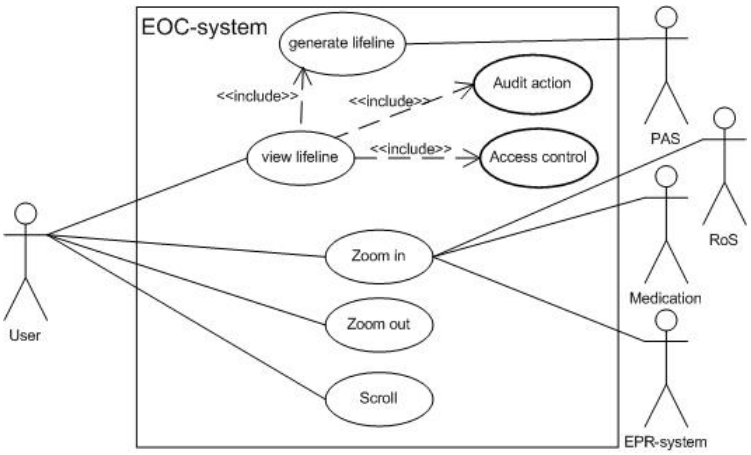


Figure 17.5: Use case showing an example of a read-action.

Figure 17.6 shows a use case diagram documenting a typical write-action in the EOC-system and the security mechanisms which have to follow this operation. It is assumed that the user is logged in to the EOC-system before this use case is triggered. As mentioned earlier, all write-actions will be signed and audited. Health personnel will be forced to follow certain quality procedures when treating a patient. If the procedures are omitted or if they are not followed step by step, then a comment must be written. Regardless of whether the quality procedures are followed or omitted, the user will need to digitally sign the followed/omitted procedure.

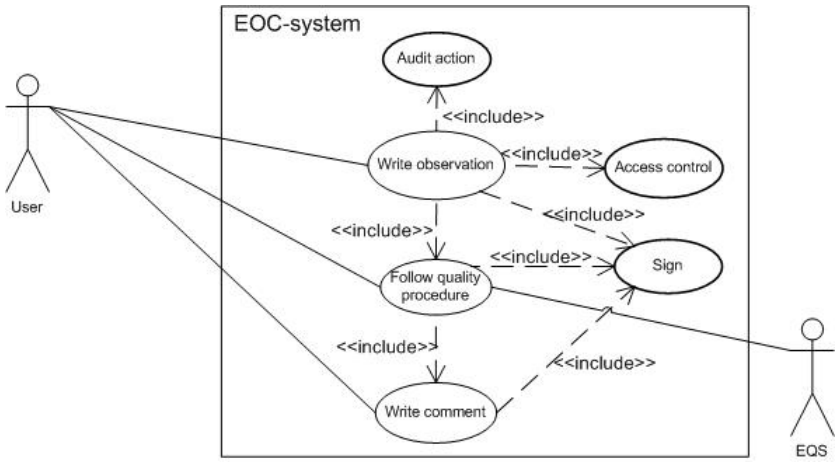


Figure 17.6: Use case showing an example of a write-action.

Figure 17.7 shows a use case diagram documenting an edit-action in the EOC-system. It is assumed that the user is logged in to the EOC-system before this use case is triggered. When e.g. a typing error is detected, the user should be allowed to edit the registration. All edits will be audited and must be signed before they are valid. When editing is done, an edit history will be required for traceability in the EOC. This edit history should be visible for all users who are later reading the information which has been edited.

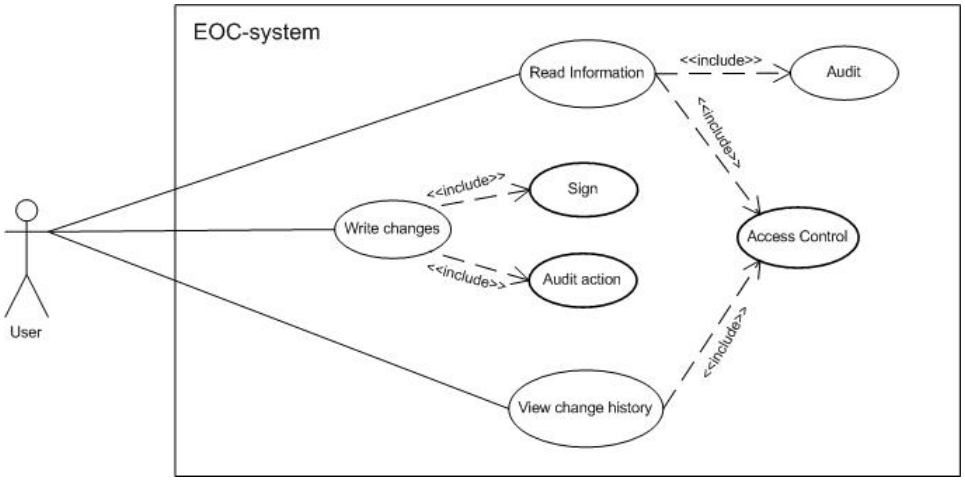


Figure 17.7: Use case showing an example of an edit-action.

17.2.2 Misuse cases

Use cases concentrate on what the system should do, while misuse cases are the inverse of use cases, concentrating on system behavior that should be avoided. In a misuse case diagram, misuses are depicted as inverted use cases, while the actor who initiates the misuses, the ***crook***, is depicted as an inverted actor. Misuse cases are described further in Appendix C in Section C.2.

The Requirement Model in Section 17.1 identifies five security threats that the target system shall prevent, detect or both prevent and detect. Each of the security threats can be seen in connection with some of the target system functionality. Misuse cases with corresponding use cases are illustrated in Figure 17.8 - Figure 17.10. For making the figures more understandable, the environment systems are not shown.

In Figure 17.8, login functionality is utilized for ***unauthorized login*** by the crook. Unauthorized login should be prevented or detected by respectively an extensive access control mechanism or auditing mechanism within the EOC-system.

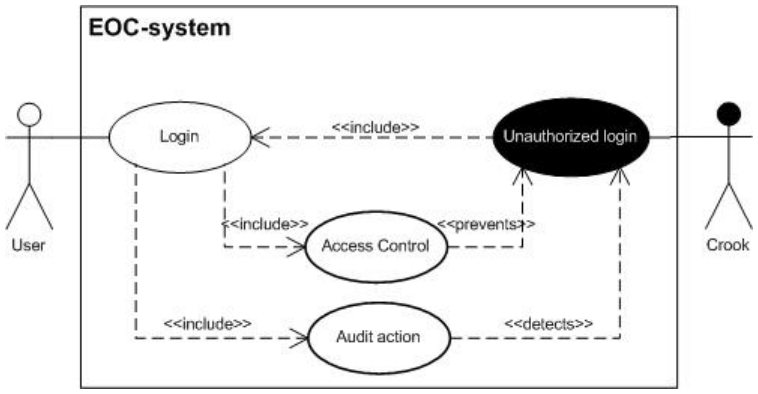


Figure 17.8: Misuse case showing unauthorized login to the target system.

In Figure 17.9, the crook might misuse the EOC-system by ***illegitimate reading*** of information. Illegitimate reading has to be prevented by the access control mechanism within the EOC-system and by encryption of information. Illegitimate reading of information shall be detected in an audit log.

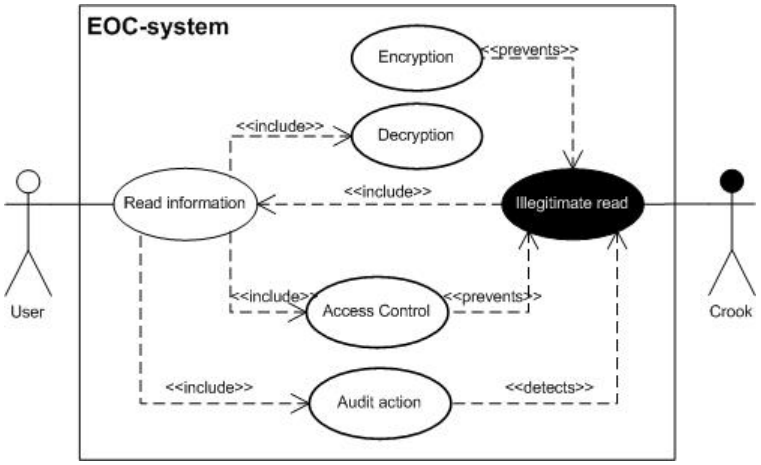


Figure 17.9: Misuse case showing illegitimate reading of information in the target system.

Figure 17.10 shows both illegitimate writes and edits of information in the target system. Write and edit actions are quite similar; they both require the user to digitally sign registrations or modifications, and they both involve encryption after the information is registered or modified.

The use cases Write information or Edit information may be utilized for respectively ***illegitimate writes*** or ***illegitimate edits*** by the crook. Both illegitimate writes and edits should be prevented by the access control mechanism within the EOC-system. Additionally, they should be detected by the auditing mechanism within the EOC-system.

Figure 17.10 also shows how the crook may misuse the EOC-system by ***tapping communication lines***. This should be prevented by encryption of information.

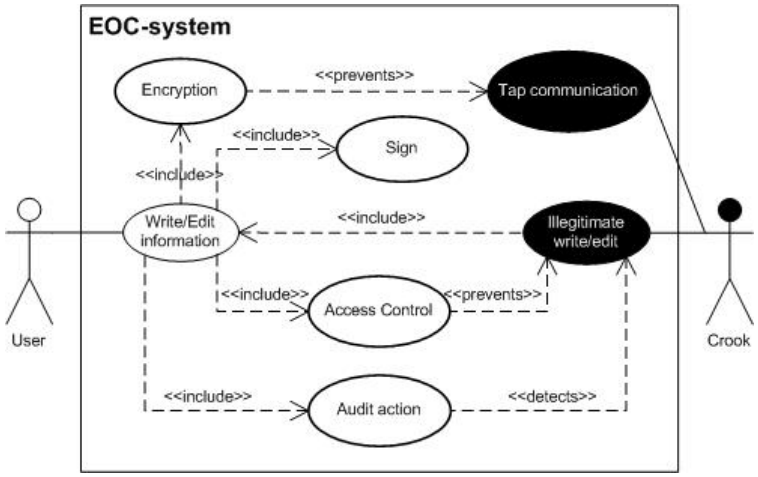


Figure 17.10: Misuse case showing illegitimate writes and edits, in addition to communication line tapping.

17.3 Target Organization Security Policy Model

Model	**Target Organization Security Policy Model**
Purpose	Identify and connect the architectural description to the security policy of the target system's organization.
Input	Customer supplied information (i.e. security policy, Governmental requirements, etc.).
Output	Textual description.

Table 17.4: Description of the Target Organization Security Policy Model as described in MAFIIA/RBAC.

Target Organization Security Policy Model should document the security policy of the target system's organization. CARDIAC primarily aims to implement their EOC-system within the Health Region for Central Norway, and they therefore have to follow the security policy of this health region.

The Health Region for Central Norway is one of the five health regions within the National Health Service where national standards, strategies, laws and regulations for information security prevail. CARDIAC therefore has to follow these national standards, strategies, laws and regulations in addition to the particular security policy within the Health Region for Central Norway.

Some of the documentation that forms the security policy of the target system's organization was described in Section 15.2. In addition to the assets from Section 15.2, there is an ongoing development of a norm for information security in the health sector [50]. The goal of this norm is to ensure a satisfactory level of security around health- and personal information.

17.4 Summary

The requirement viewpoint identifies both functional and non-functional requirements to the target system. Functional requirements are further depicted in UML use case diagrams, while some security related requirements are depicted in UML misuse case diagrams.

Chapter 18

Component Viewpoint

The component viewpoint describes the target system by its subsystems, components and information objects. The component viewpoint also describes the interaction between the target system's subsystems, components and information objects. These descriptions are at a functional level and as far as possible technology independent.

The component viewpoint includes six different models:

- System Information Model defines the information semantics for relevant concepts that must be understood in a common way in the system.
- System Decomposition Model decomposes the target system into subsystems and components.
- System Collaboration Model describes how subsystems and components collaborate to form central mechanisms in the target system.
- Component and Interface Specification Model details each interface and component that was identified in the two previous models.
- System Security Model describes how security is handled by the components.
- System Access Control Model concerns how access control is handled within the target system.

18.1 System Information Model

Model	**System Information Model**
Purpose	Specify the relationships between and properties of the central information objects in the system that must always be true (invariants).
Input	Requirement viewpoint.
Output	UML and UMLsec class diagram, Textual description.

Table 18.1: System Information Model as described in generic MAFIIA and MAFIIA/RBAC.

Figure 18.1 gives an overview of concrete domain information objects in the target system. A ***patient*** gives ***health personnel*** the ***consent*** to process and insert patient care information in an ***EOC***. Health personnel has the possibility to base the particular EOC on several ***templates***, which provide for flexible observation charts for different patients, wards and hospitals.

The EOC is a gathering of patient care information from several different health information systems. Health personnel registers requested tests and examinations by writing ***requisitions***, ***referrals*** or ***prescriptions***. In return they get ***test results***. ***Medication*** related information, ***observations*** and ***measurements*** regarding the patient are also registered in the EOC. In addition, it is possible to view ***textual documentation***, ***graphs*** and ***trend curves***. When patient care is given, health personnel has to follow some ***quality procedures***.

Figure 18.2 describes the meta-information that is needed to integrate and process the information objects in the target system. Rules for selection, transformation, insertion and securing the information are shown in the figure. These rules refer to the requirement viewpoint, more precisely to the concern related requirements. For example, selection rules are concerned with the retrieving and presentation of information, while transformation rules are concerned with the sending, synchronization and editing of information. Insertion rules deal with registration of information, while security rules deal with quality related concerns, such as digital signing, secure communication, auditing and access control.

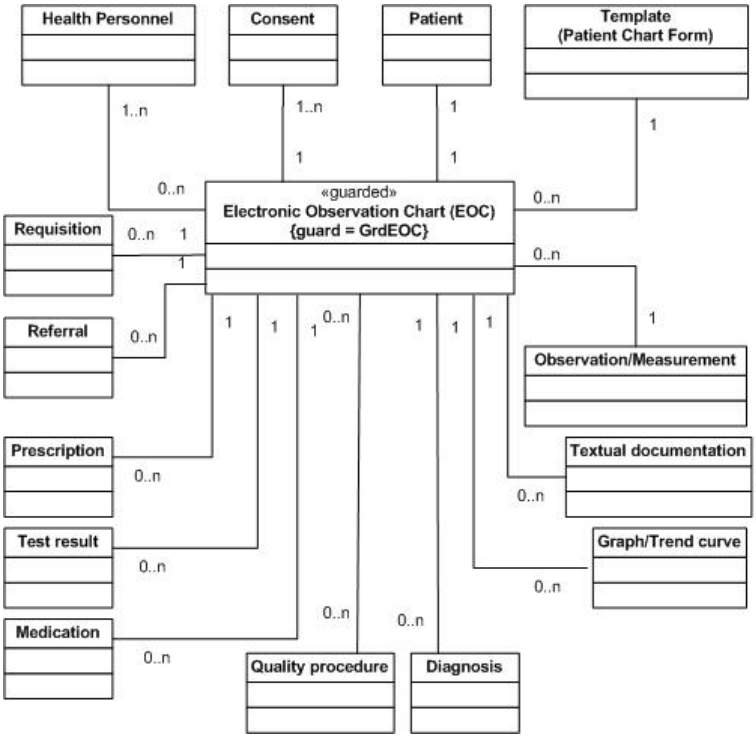


Figure 18.1: System Information Model showing concrete domain information objects and their relations.

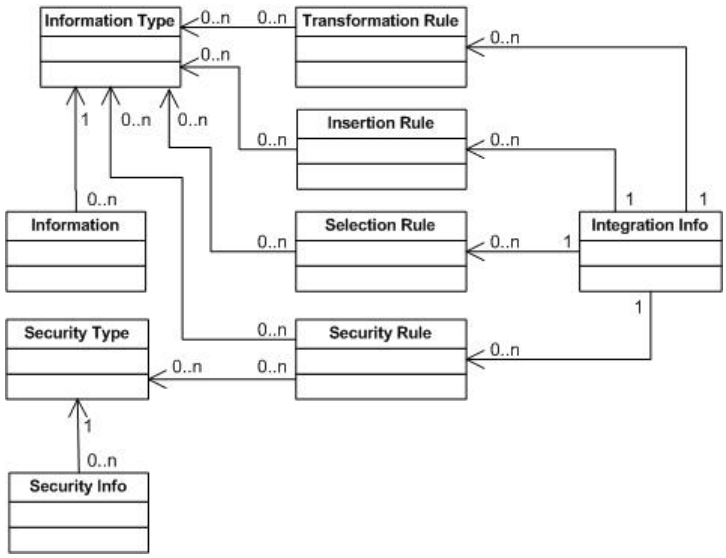


Figure 18.2: System Information Model showing generic meta-information in the target system.

18.2 System Decomposition Model

Model	**System Decomposition Model**
Purpose	Describe how the system is divided into different subsystems or components, and how these are related to form a coherent whole.
Input	Requirement viewpoint.
Output	UML and UMLsec class diagram.

Table 18.2: System Decomposition Model as described in generic MAFIIA and MAFIIA/RBAC.

Figure 18.3 shows a decomposition of the target system into subsystems. Each of these subsystems is described in the subsequent sections where they will be further decomposed into components.

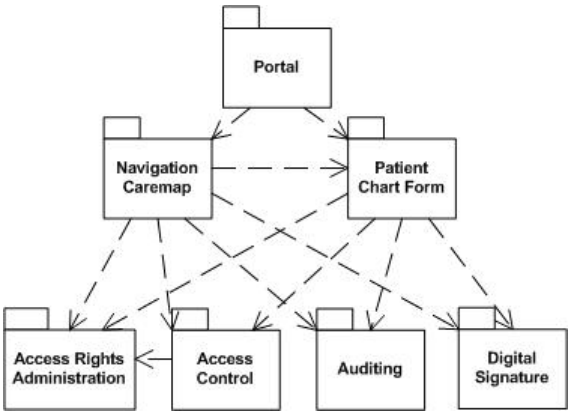


Figure 18.3: The target system divided into subsystems.

The security subsystems Access Control, Access Rights Administration, Auditing and Digital Signature respectively preserve the security concerns identified in Section 15.1.

18.2.1 Navigation Caremap

The Navigation Caremap subsystem consists of several components, shown in Figure 18.4. These components are stereotyped, some of them from the reference architecture and some of them newly defined. The reference architecture stereotypes are *UserInterface*, *UserService*, *BusinessService*, *ResourceService* and *EnvironementInterfacing*, while the newly defined stereotype is *ApplServer*.

Reference architecture stereotypes refer to the different layers in the reference architecture. Components with these stereotypes reflect functionality and services provided in the particular layer. ApplServer is a generalization of the stereotype EnvironmentInterfacing. ApplServer is an abbreviation for application server, and components with this stereotype are concerned with the target system's interfacing to the environment. For a combination of service-oriented and portal-oriented integration architecture, application servers are important integration means. Application servers host system logic, interface processing and resource connections, which may include support for Web services.

The components in the Navigation Caremap subsystem, which are depicted in Figure 18.4, are described below. Dependencies between these components are shown as arrows.

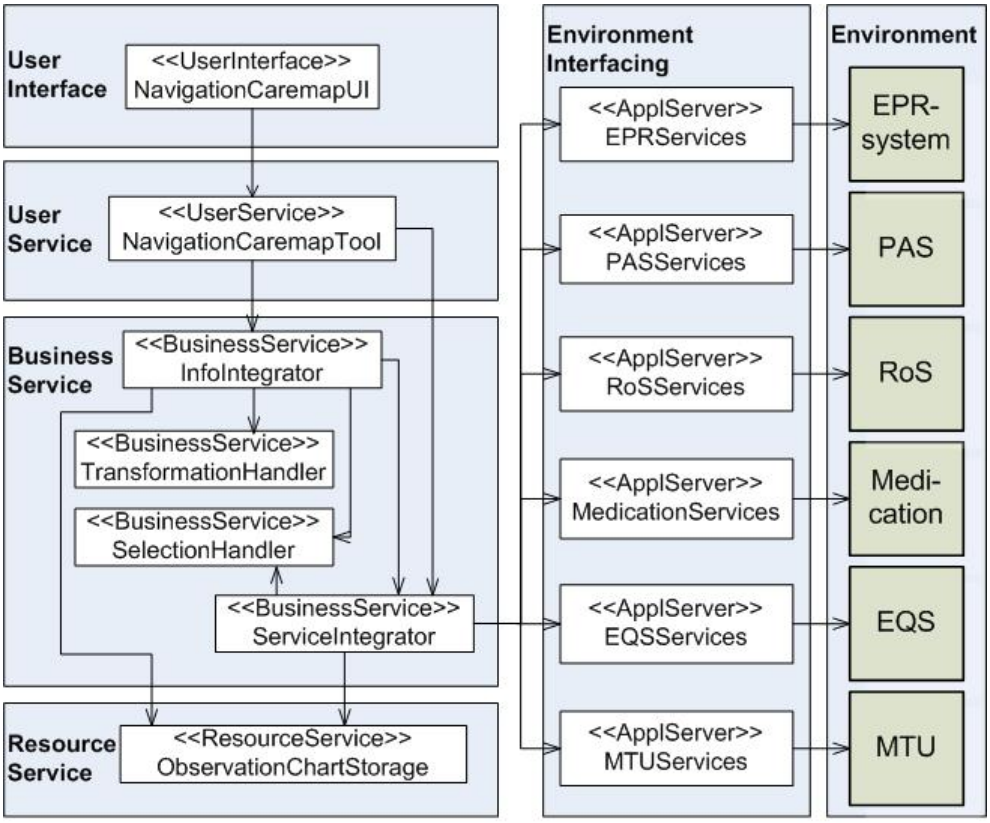


Figure 18.4: System Decomposition Model showing the Navigation Caremap subsystem of the target system.

The ***NavigationCaremapUI*** component indicates the Navigation Caremap user interface.

The ***NavigationCaremapTool*** component handles user related logic,

and it provides a categorical overview of the patient care. For this purpose, the NavigationCaremapTool is dependent on two BusinessService components, the InfoIntegrator and the ServiceIntegrator.

The ***InfoIntegrator*** component integrates information. For this purpose, it is dependent on the ServiceIntegrator, which separates information from services. InfoIntegrator is also dependent on the TransformationHandler and the SelectionHandler because information is selected and in some cases transformed during integration.

The ***ServiceIntegrator*** component invokes services provided by the environment systems. When the invocation is done, it may separate information from the services. Similar to the InfoIntegrator, the ServiceIntegrator component is dependent on the SelectionHandler. The ServiceHandler also depends upon services provided by environment systems, e.g. EPRServices, PASServices, etc.

The ***SelectionHandler*** selects and retrieves information and services for the purpose of information and service integration. Information is mainly retrieved from the ObservationChartStorage, while services are requested from the environment systems.

The ***TransformationHandler*** transforms information after it is retrieved from the environment systems or the ObservationChartStorage or before it is stored in the ObservationChartStorage.

Services are provided by the ApplServer stereotyped components, ***EPRServices***, ***PASServices***, ***RoSServices***, ***MedicationServices***, ***EQSServices*** and ***MTUServices***, which correspond to the following information sources/ environment systems: EPR-system, PAS, RoS, Medication, EQS and MTU.

Figure 18.4 also shows that registered information is stored in the ***ObservationChartStorage***.

The Façade pattern is used in this subsystem. Figure 18.4 shows how the InfoIntegrator is used as façade against more detailed components. The InfoIntegrator gives coarse-grained access to information, while the SelectionHandler and TransformationHandler give fine-grained access. By splitting up components for read and update, their implementation is simplified. In addition, this provides a better overview of the components. The Façade pattern is described in Appendix B.

18.2.2 Patient Chart Form

As shown in Figure 18.5, it is possible to decompose the Patient Chart Form subsystem into several components. These components are quite similar to the components in the Navigation Caremap subsystem, which were shown in Figure 18.4. Still, there are some differences because the navigation caremap is read-only and the patient chart form is not.

The Patient Chart Form subsystem supports locally adapted observation charts through configurable templates, which are handled by the components

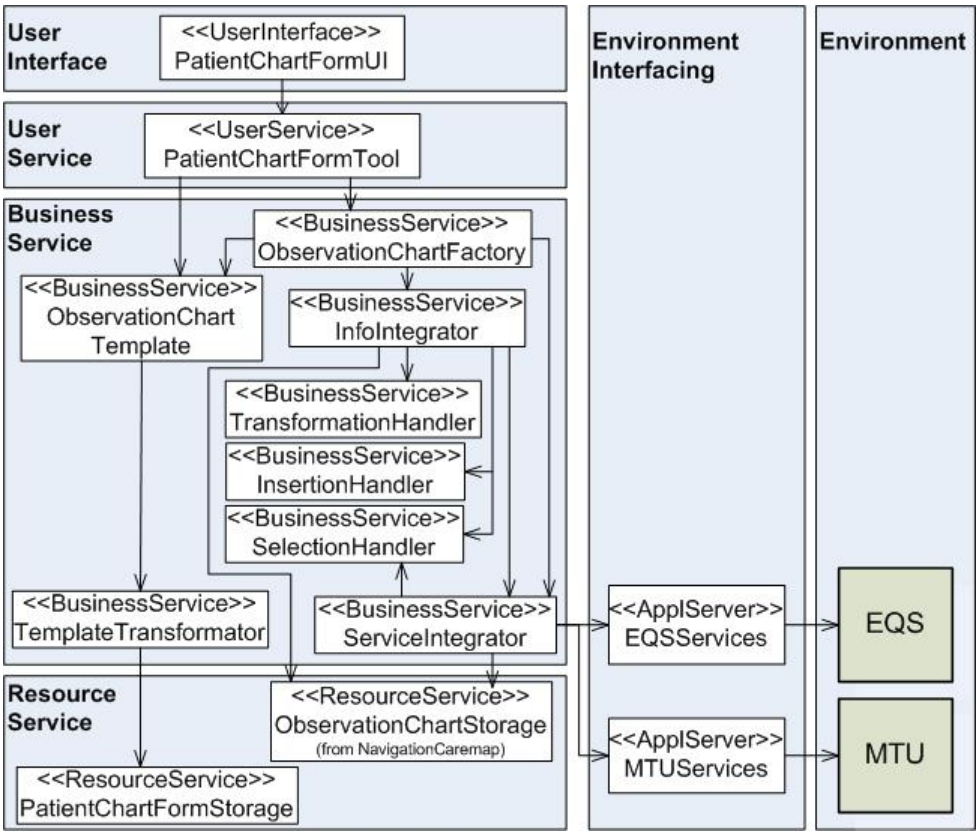


Figure 18.5: System Decomposition Model showing the Patient Chart Form subsystem of the target system.

ObservationChartTemplate and ***TemplateTransformator***. The ObservationChartTemplate component provides configurable observation charts templates, while the TemplateTransformator component provides support for configuration of the templates. TemplateTransformator needs to retrieve observation chart templates from the PatientChartFormStorage. This indicates that TemplateTransformator is dependent on PatientChartFormStorage, which is shown by an arrow in Figure 18.5.

The ***ObservationChartFactory*** component makes new observation charts based on different templates.

The components ***InfoIntegrator***, ***ServiceIntegrator***, ***TransformationHandler*** and ***SelectionHandler*** were described in the Navigation Caremap subsystem and will not be mentioned again here.

InsertionHandler indicates that insertions or registrations of information are allowed in the Patient Chart Form subsystem.

The environment systems EQS and MTU provide services. The ***EQSServices*** component indicates services provided by EQS, while the ***MTUSer-***

vices component indicates services provided by MTU. EQSServices relates to quality procedures which have to be followed during patient care, while MTUServices relates to the retrieval of observations and measurements from MTU. These observations and measurements are automatically registered in the EOC-system.

ObservationChartStorage stores patient information in the EOCs, while ***PatientChartForm*** stores observation chart templates. ObservationChartStorage is the same component as ObservationChartStorage in the Navigation Caremap subsystem.

Similar to the Navigation Caremap subsystem, the Façade pattern is used in this subsystem. The InfoIntegrator gives coarse-grained access to information resources, while the SelectionHandler, TransformationHandler and InsertionHandler give fine-grained access to these resources. In addition, the ObservationChartTemplate functions as a façade for the TemplateTransformator. The Façade pattern is described in Appendix B.

18.2.3 Portal

The Portal subsystem is the main user interface in the target system. All user accesses are handled by the Portal.

As shown in Figure 18.6, the Portal can be decomposed into the following components:

PortalUI is the user interface for the target system.

PortalTool handles user related logic in the target system. The Portal subsystem is responsible for the gathering of all target system functionality, and the PortalTool component is therefore dependent on the ***NavigationCaremapTool*** (from the Navigation Caremap subsystem) and the ***PatientChartFormTool*** (from the Patient Chart Form subsystem). The PortalTool component makes use of the NavigationCaremapTool to present an overview of the patient care. The PatientChartFormTool is used to register measurements and observations in the observation charts and to create different observation charts based on some templates. In addition, shortcuts to the environment systems should be provided in this subsystem. The PortalTool is therefore dependent on the ShortcutHandler component.

The ***ShortcutHandler*** makes shortcuts to different parts of the target system easily accessible. These shortcuts are provided by the components ***EPRShortcut***, ***PASShorcut***, ***RoSShortcut***, ***MedicationShortcut***, ***EQSShortcut*** and ***MTUShortcut***.

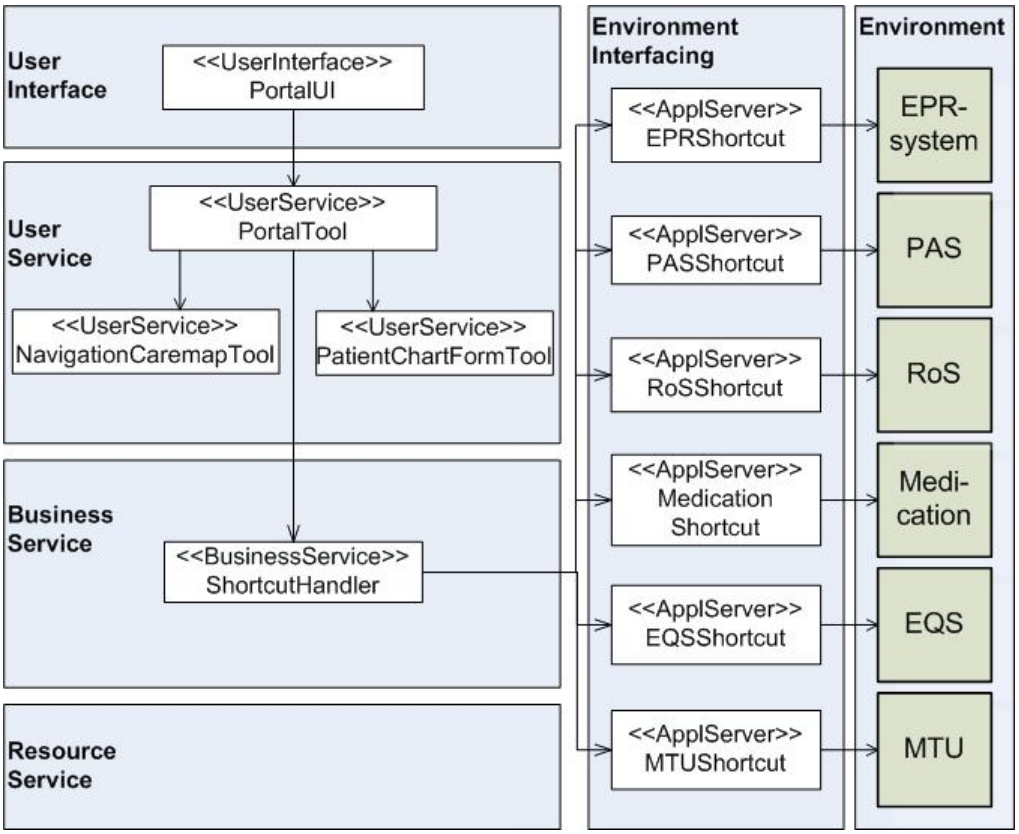


Figure 18.6: System Decomposition Model showing the Portal subsystem of the target system.

18.2.4 Access Control

The Access Control subsystem preserves the security concern Access Control, which is described in Section 15.1. As shown in Figure 18.7, this subsystem consists of several components.

The ***LoginUI*** component provides the user interface for login in the target system.

The ***LoginController*** handles the user related logic for login. It serves as an authentication component which verifies against the SSOStorage component that login information is correct. If this information is incorrect the LoginUI component will be used to inform the user about the login failure and suggest a second try.

AccessManager is responsible for the mapping of users towards resources on the basis of the users' role and access rights. It is dependent on the RoleManager and ObservationChartStorage. The AccessManager knows which access rights a certain user has on an information resource. The Ac-

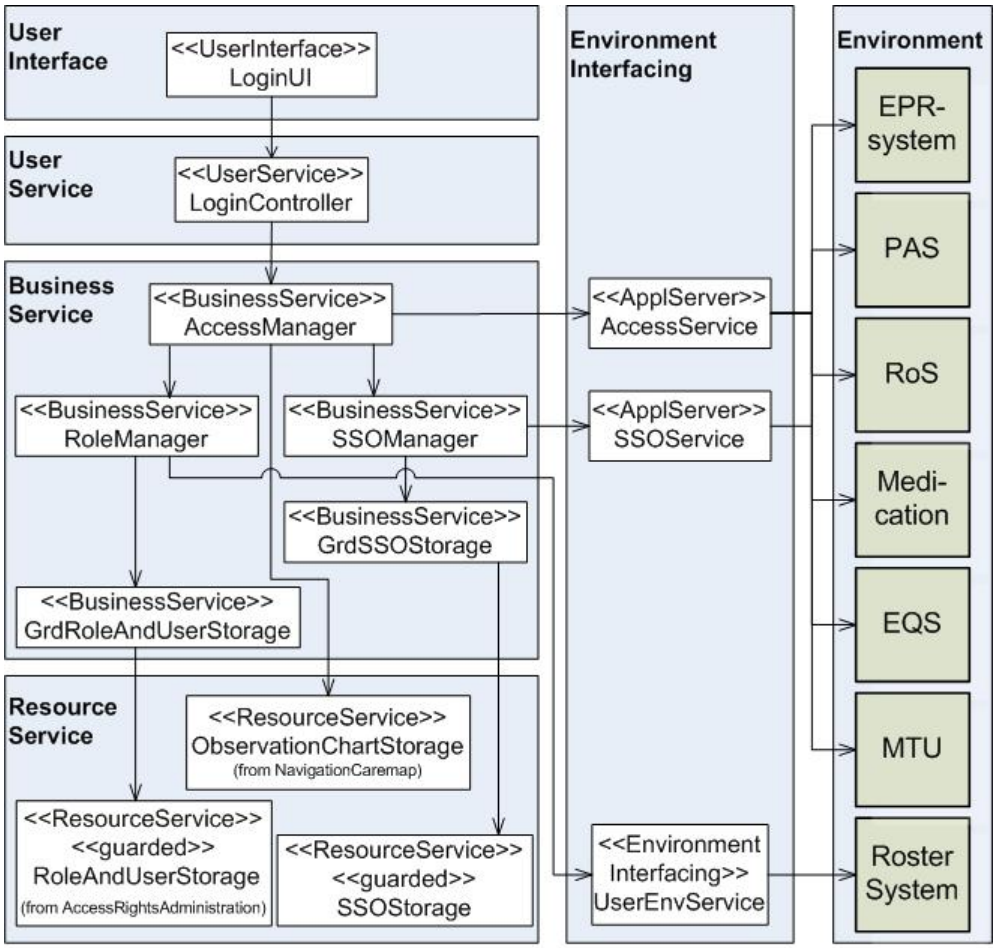


Figure 18.7: System Decomposition Model showing the Access Control subsystem of the target system.

cessManager is also dependent on the SSOManager for the purpose of access control during single sign-on. In addition, the AccessManager is dependent on the ***AccessService*** component for the retrieval of access rights within the environment systems. AccessService is a service provided by each of the environment systems for the purpose of ensuring the access control mechanism within the environment systems. Refer to the requirement viewpoint where requirement R2.2.1.4.13 states that the target system shall ensure the access control mechanism within each of the environment systems.

RoleManager handles administration of roles and access rights. It retrieves roles and access rights from GrdRoleAndUserStorage, which is a guard for the RoleAndUserStorage component, and maps these towards one another. It also handles role assignments by mapping users to correct roles dependent on the patient in care. For this task, the RoleManager needs

information about the user and the environment where the user operates. Therefore, it is dependent on the ***UserEnvService*** component for the retrieval of user environment information, such as rosters, employee relations, ward belonging, etc. The UserEnvService component might be dependent on several environment systems such as a roster system, employee system, etc. But, in Figure 18.7 only a ***Roster system*** is shown.

RoleAndUserStorage is guarded and needs to be accessed through its guard, ***GrdRoleAndUserStorage***. RoleAndUserStorage stores information about users, roles and access rights defined in the Access Rights Administration subsystem, hence labeled with *(from AccessRightsAdministration)*.

SSOManager is responsible for the mapping of userIDs from the target system towards the corresponding userIDs in the environment systems, so that the user only has to authenticate himself once per session, i.e. single sign-on. The SSOManager is dependent on SSOService and SSOStorage for the retrieval of usernames and passwords in the environment systems and the target system.

SSOService is a service provided by each of the environment systems, and it is therefore dependent on these systems, i.e. EPR-system, PAS, RoS, Medication, EQS and MTU.

SSOStorage is guarded and needs to be accessed through its guard, ***GrdSSOStorage***. SSOStorage stores all usernames and the belonging passwords for all users of the target system, i.e. it stores usernames and passwords of all the users who are using the target system in addition to usernames and passwords in the environment systems of the target system.

ObservationChartStorage stores all patient information stored in an EOC. This information is stored in the Navigation Caremap subsystem or in the Patient Chart Form subsystem, hence labeled with *(from Navigation-Caremap)*.

As indicated in the description of the AccessManager and RoleManager, the Role-based access control pattern is used in this subsystem. The intention of this pattern is to assign rights to users according to their roles in the organization. This pattern is described in Appendix B.

Login

Figure 18.8 shows how the Access Control subsystem handles logins. As stated in requirement R2.2.1.4.2 in the requirement viewpoint, the target system shall provide a two-factor user authentication. This is shown in the figure where the user has to authenticate himself with username and password, in addition to a digital certificate.

Target system users, mainly ***Health Personnel***, log in to the system through the ***LoginUI***. Login information is forwarded to the ***LoginController*** which separates the certification information and forwards it to the

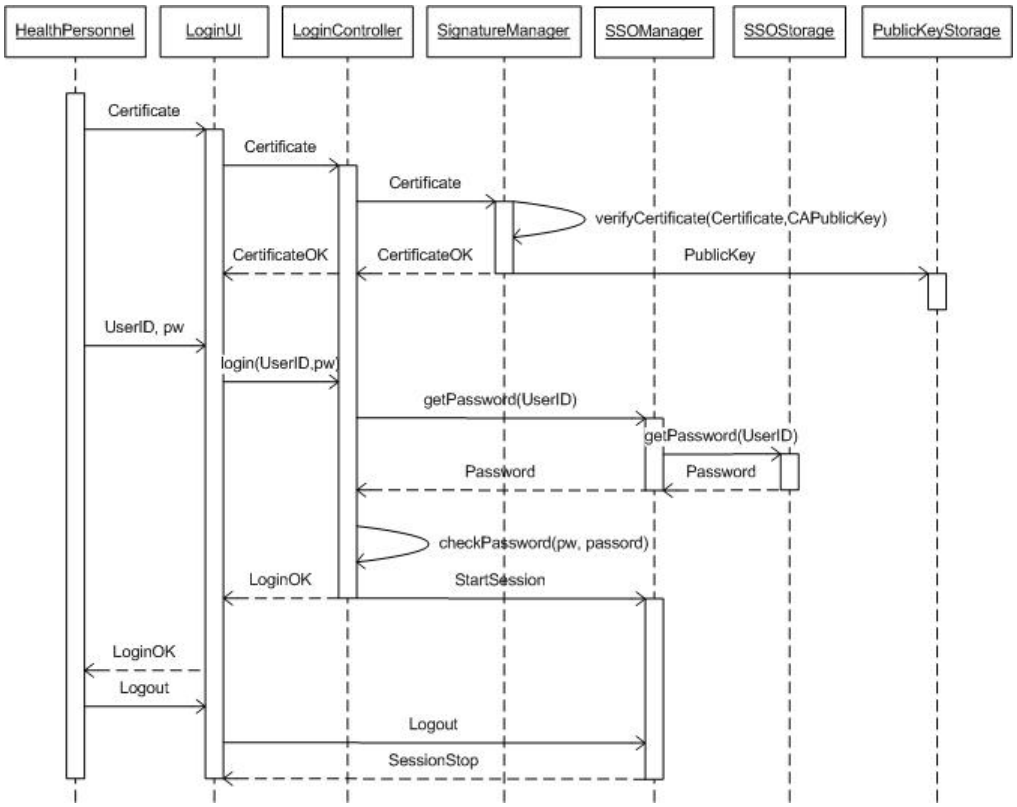


Figure 18.8: Sequence diagram for target system login.

SignatureManager. The SignatureManager is described in further detail in Section 18.2.5. The certificate is validated through the method ***verifyCertificate()*** by checking the Certification Authority's (CA's) digital signature by means of the CA's public key. If the certificate is valid, the public key is stored in the ***PublicKeyStorage***, also described in Section 18.2.5. Through the method ***checkPassword()*** the LoginController validates the rest of the login information. Username and password are validated against the ***SSOStorage*** by the ***SSOManager***.

When login information is correct, the LoginController informs the user and starts a session, which is handled by the SSOManager. When the user logs out, a logout message is sent to the SSOManager, and the session is terminated.

Figure 18.9 shows three different scenarios for how the Access Control subsystem handles failed logins.

In Scenario A, the certificate is not valid and the user is notified before he has to type in username and password.

In both Scenario B and Scenario C it is assumed that the certificate is valid. In Scenario B, the username is not found in the SSOStorage and

therefore assumed to be incorrect. The user is notified that the password is not valid. In Scenario C, the username is found and its belonging password is compared with the password the user typed in. The two passwords do not match and the user is notified.

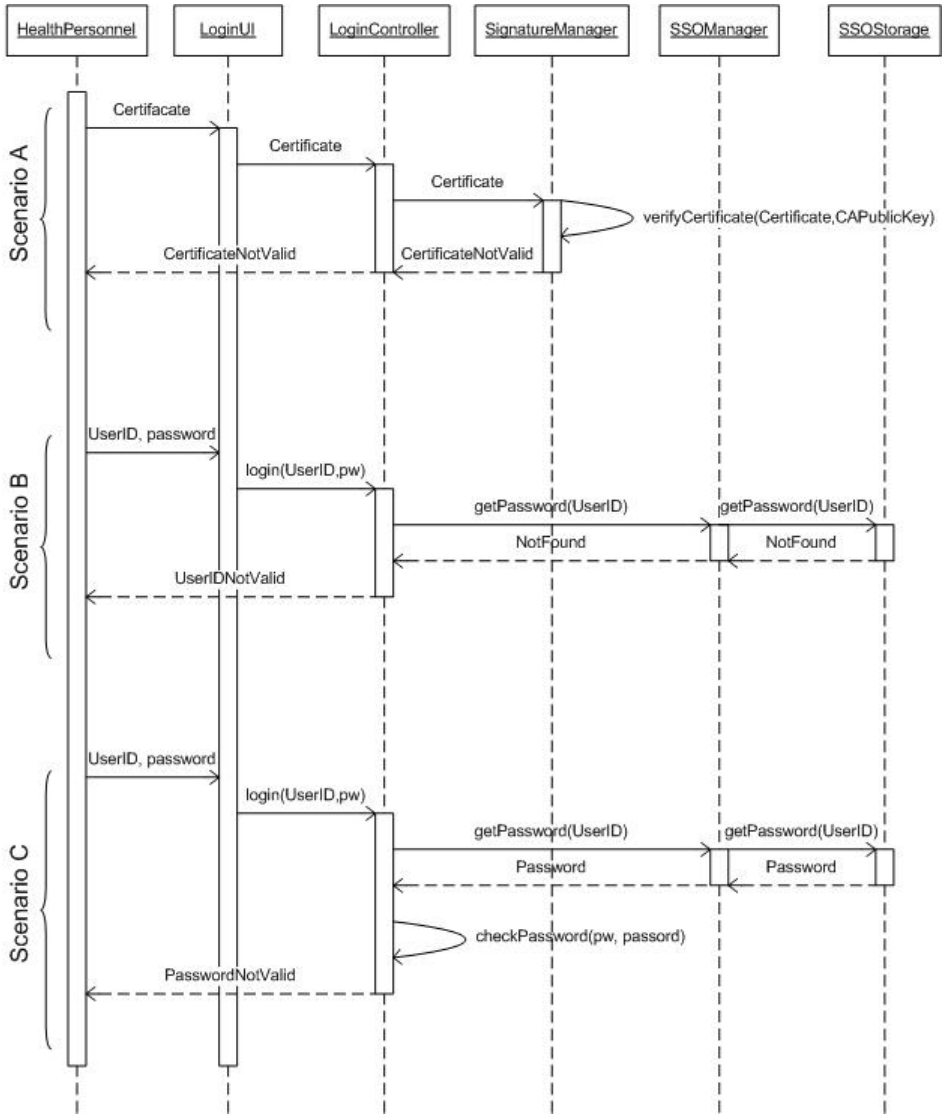


Figure 18.9: Sequence diagram for target system failed logins.

Select

Figure 18.10 shows how the AccessManager component in the Access Control subsystem handles selection of authorized information. The AccessManager first has to retrieve access rights for the selected information. Then, it has

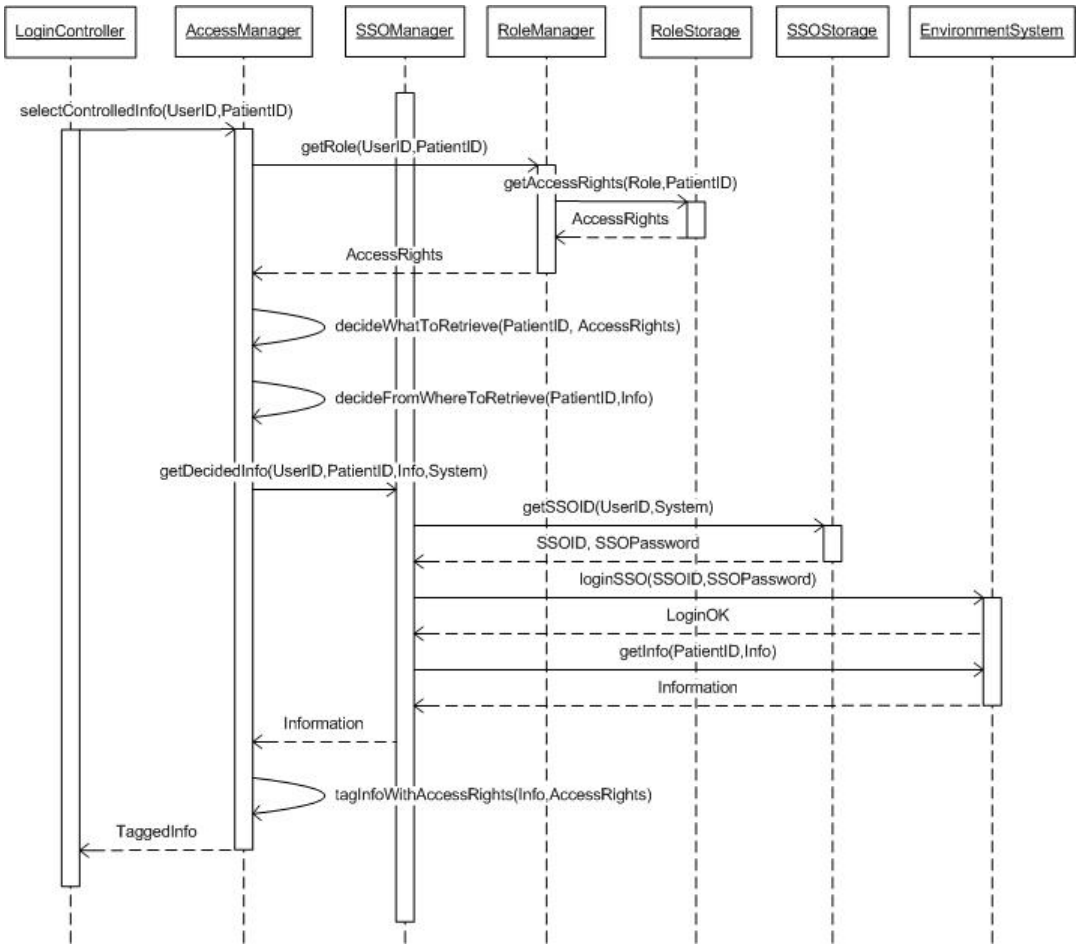


Figure 18.10: Sequence diagram for selection of authorized information.

to decide what information to retrieve based on the user's request and access rights, e.g. it is only allowed to retrieve a subset of the requested information. After that the AccessManager has to decide where the information can be retrieved, e.g. in one of the environment systems. Single sign-on is handled by the SSOManager, and the information is then retrieved. The retrieved information is tagged with meta-information, indicating which parts of the information that only can be read or edited. In addition, the meta-information might indicate where it is allowed to add new information.

Transform

Figure 18.11 shows how the AccessManager component in the Access Control subsystem handles transformation of authorized information. Before any transformation is allowed, the AccessManager retrieves the access rights for the particular user and maps these access rights to the inquired information.

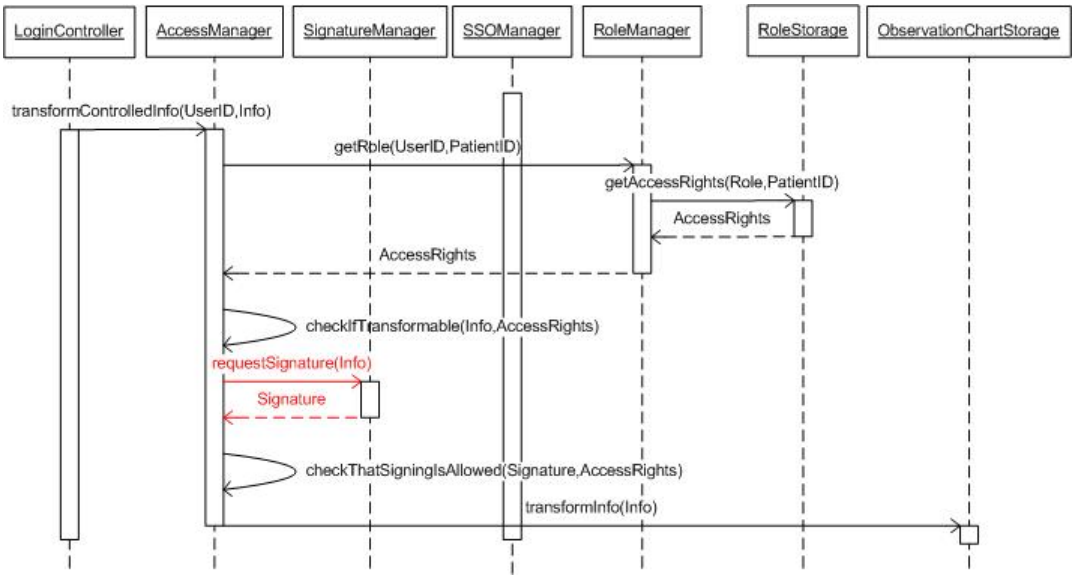


Figure 18.11: Sequence diagram for transformation of authorized information.

It then has to check if the transformations are allowed pursuant to these access rights. If the transformations are allowed, the user has to digitally sign the transformation. The method ***requestSignature()*** and its response ***Signature*** are shown in red because it is further depicted in Figure 18.14 in Section 18.2.5. If the user is allowed to sign this transformation, the actual transformations are performed and the transformed information is stored in the ObservationChartStorage.

Insert

Figure 18.12 shows how the AccessManager component in the Access Control subsystem handles insertion of authorized information. The AccessManager first has to retrieve the access rights for the particular user and check if he is allowed to insert information to this patient's observation chart. If the user is allowed to insert information, the target system requests him to digitally sign the insertion. The method ***requestSignature()*** and its response ***Signature*** are shown in red because it is further depicted in Figure 18.14 in Section 18.2.5. If the user is allowed to sign the insertion, the information is inserted in the ObservationChartStorage.

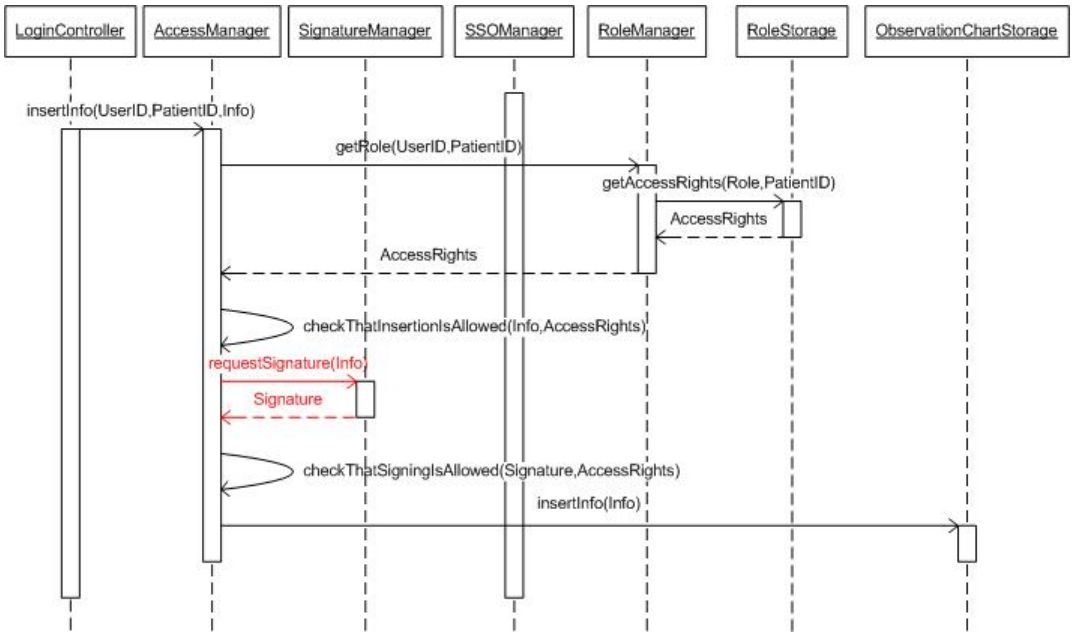


Figure 18.12: Sequence diagram for insertion of authorized information.

18.2.5 Digital Signature

The Digital Signature subsystem is related to the security concern Digital Signing described in Section 15.1. To preserve digital signing in the target system, this subsystem is decomposed into two components. This is shown in Figure 18.13.

SignatureManager creates signatures, verifies certificates and stores public keys in the PublicKeyStorage. For the creation of signatures, it is dependent on ***LoginController*** from the Access Control subsystem, which is described in Section 18.2.4. For the verification of certificates, it is dependent on a Certification Authority, ***CA***. The CA is a trusted third party which signs the certificate with its private key. The SignatureManager verifies the certificate using the CA's public key.

PublicKeyStorage stores public keys obtained from the certificates.

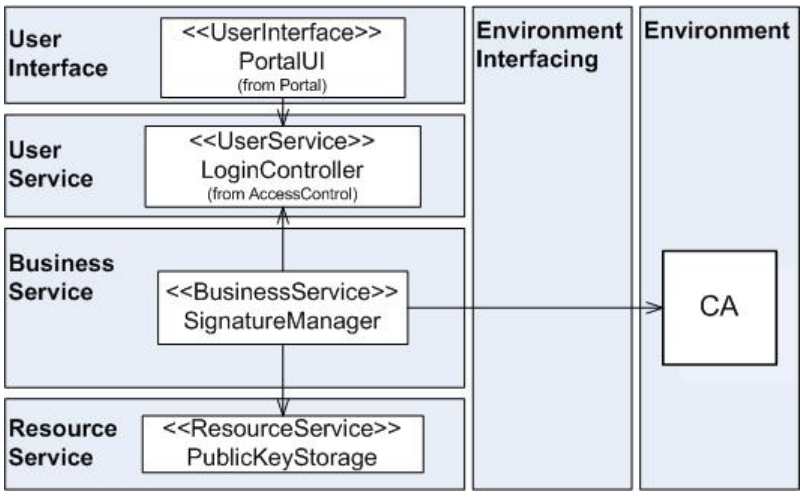


Figure 18.13: System Decomposition Model showing the Digital Signature subsystem of the target system.

Transformation and insertion

Figure 18.14 gives a more detailed illustration of how digital signing of transformations and insertions is handled by the ***SignatureManager***. This figure has to be seen in relation to Figure 18.11 and Figure 18.12, and arrows depicted in both figures are shown in red.

The SignatureManager receives a request from the AccessManager, ***requestSignature()***. The SignatureManager then requests the LoginController for the user's username and password. The user digitally signs the insertion by typing in his username and password, and the LoginController verifies the password and forwards it to the SignatureManager if it is valid. The SignatureManager then creates an *authenticator*, which is a small block of bits of the transformed or inserted information that is hashed, in method ***createAuthenticator()***. The authenticator is encrypted with the user's private key, more precisely his password, in method ***createSignature()***. The encrypted authenticator serves as a signature that verifies the origin of the information, and the SignatureManager sends this ***Signature*** to the AccessManager.

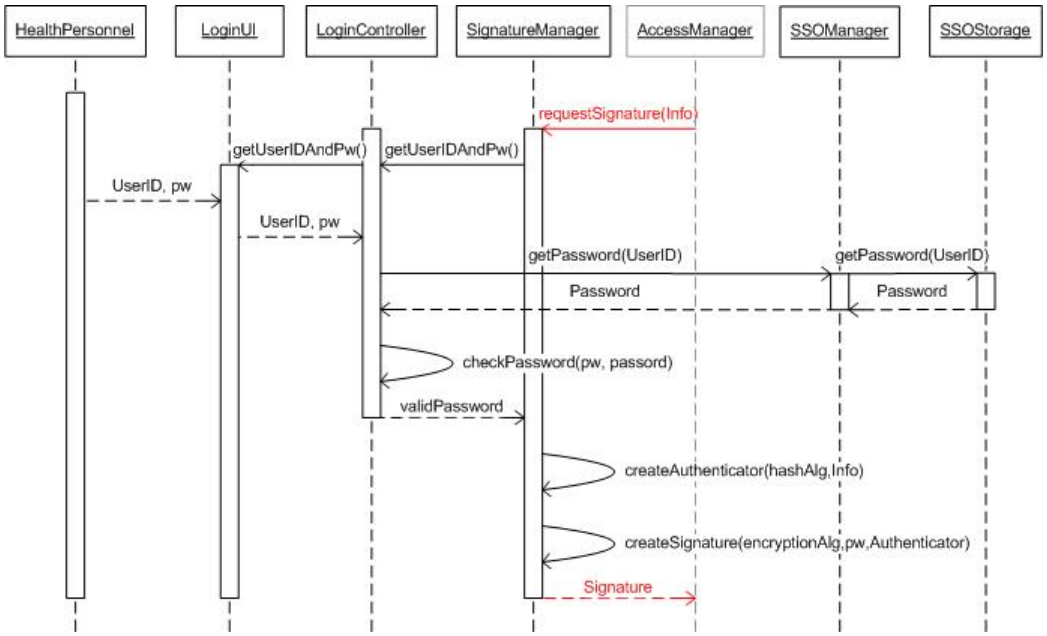


Figure 18.14: Sequence diagram for for digital signatures when authorized information is inserted.

18.2.6 Access Rights Administration

Together with the Access Control subsystem, the Access Rights Administration subsystem shows how the security concern Access Control is preserved. Figure 18.15 illustrates how the Access Rights Administration subsystem can be further decomposed into components.

AdminUI is the user interface in the Access Rights Administration subsystem.

System administrators or other users which have the right to administrate access rights have to log in through the ***AdminTool***. The Single access point pattern is chosen for the Access Rights Administration subsystem, and the AdminTool serves as the single access point. The Single access point pattern is described in Appendix B.

The ***RoleAndUserAdm*** component supports definition of users, roles and access rights. In this component it is possible to attach/detach users to roles and access rights.

The ***RoleAdm*** component supports initiation and administration of roles and access rights. Roles and access rights are retrieved and stored in the RoleAndUserStorage component, which is accessed through its guard GrdRoleAndUserStorage.

The ***UserAdm*** component supports initiation and administration of user accounts. User accounts and their belonging user information are also

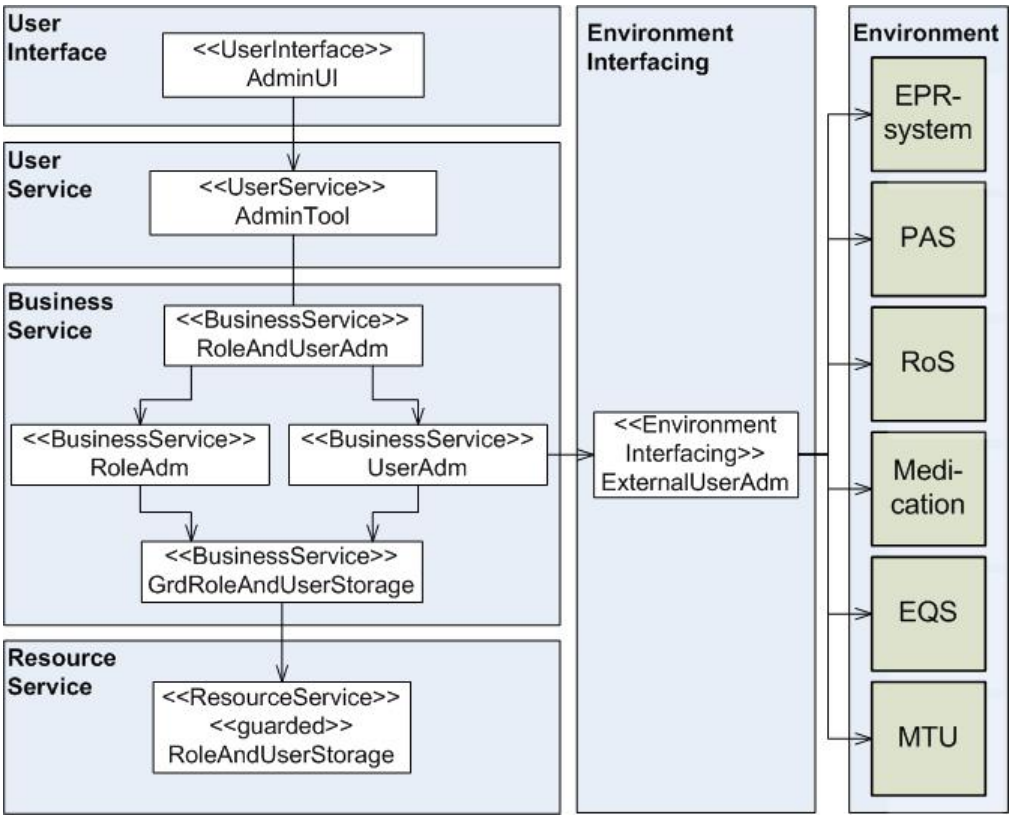


Figure 18.15: System Decomposition Model showing the Access Rights Administration subsystem of the target system.

retrieved and stored in the RoleAndUserStorage component. UserAdm also has the possibility to retrieve external user accounts through the ***ExternalUserAdm*** component. ExternalUserAdm makes it possible to integrate user administration with the environment systems' user databases. ExternalUserAdm is therefore dependent on each of the environment systems.

The ***RoleAndUserStorage*** stores information about users, roles and access rights. This component is labeled with $\langle\langle guarded \rangle\rangle$, and can only be access through its guard ***GrdRoleAndUserStorage***.

Delegation of access rights

Figure 18.16 shows how delegation of access rights is handled by the Access Rights Administration subsystem. Access Rights Administration is dependent on some of the components in the Access Control subsystem. These components, AccessManager and RoleManager, are shaded grey in the figure.

Scenario A shows a sequence for delegation of access rights where one user

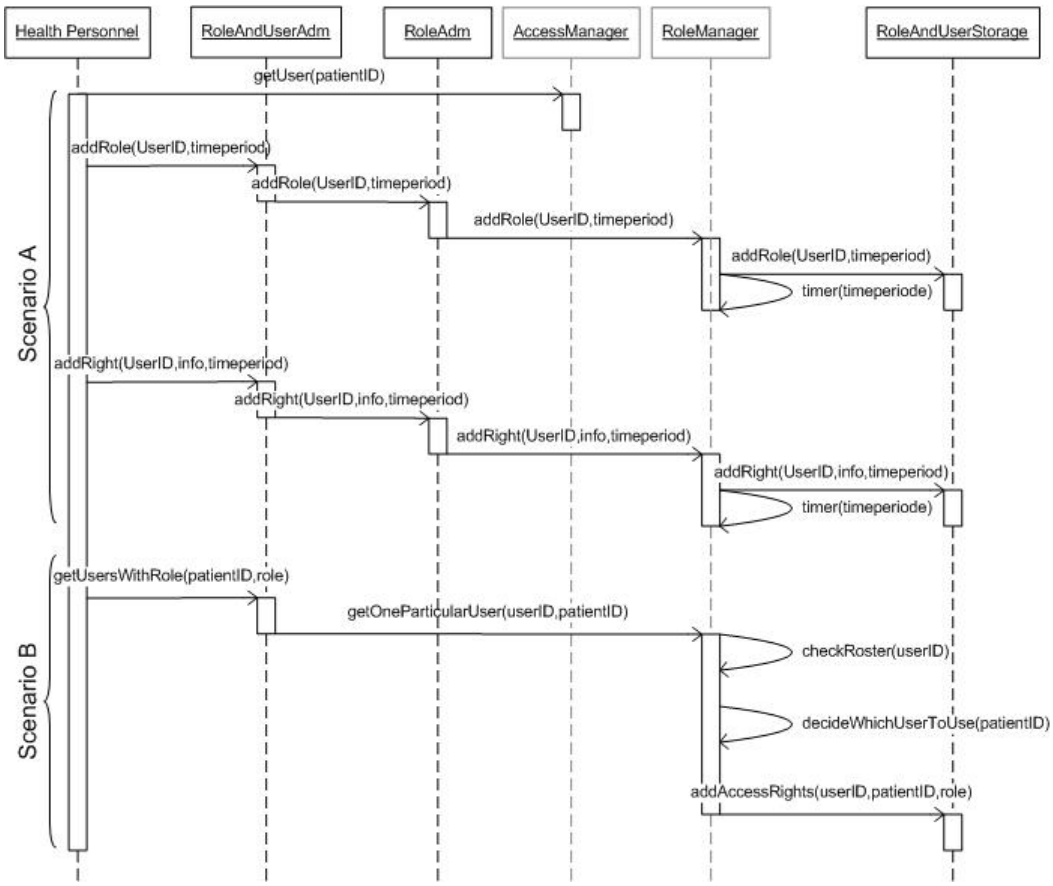


Figure 18.16: Delegation of access rights.

delegates access rights to another. The user's identification is first retrieved based on one particular patient, before rights are delegated by adding a role to the user for a certain time period. The user can also be delegated access rights for specific information resources for a certain period.

In Scenario B, access rights are delegated to a user based on a particular role. For example, if the doctor in charge sends a referral to a specialist, the referral is dependent on the user's role, not the user himself. Therefore, users with the particular role are first retrieved. Then, one particular user is chosen based on rosters and the patient in care. The chosen user is then assigned access rights for specific information resources belonging to the particular patient.

18.2.7 Auditing

The Auditing subsystem shows how the security concern Auditing, which is described in Section 15.1, is preserved. For this purpose, the Auditing subsystem is further decomposed into components, as shown in Figure 18.17.

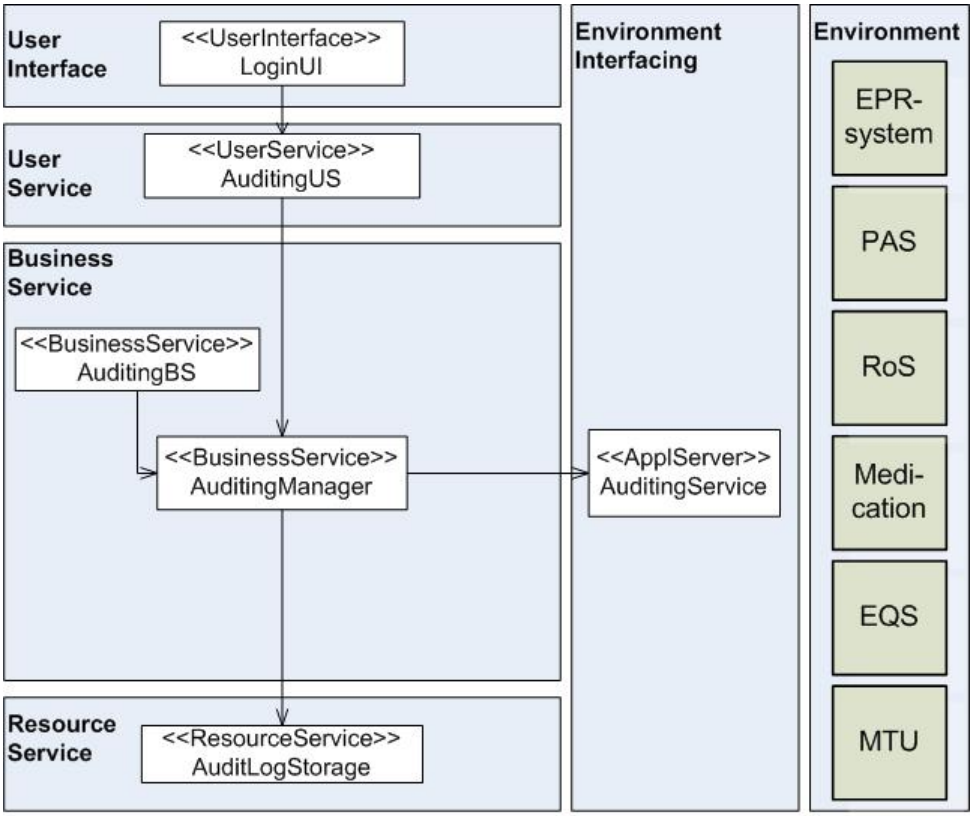


Figure 18.17: System Decomposition Model showing auditing as a subsystem of the target system.

The ***LoginUI*** component is the user interface from the Access Control subsystem, as shown in Figure 18.7.

AuditingUS and ***AuditingBS*** respectively audit service accesses on the user service tier and business service tier, i.e. AuditingUS audits target system logins, while AuditingBS audits target system operations.

The ***AuditingManager*** coordinates both audit logs from AuditingUS, AuditingBS and AuditingService, and stores them in the ***AuditLogStorage***.

The ***AuditingService*** component provides a service for auditing of invoked services provided by environment systems. Refer to the requirement viewpoint where requirement R2.2.1.3.1 states that the service which receives the request is responsible for auditing.

18.3 System Collaboration Model

Model	**System Collaboration Model**
Purpose	Describe the main interactions in the system as a set of collaborating components.
Input	Target System Interface Model (Requirement viewpoint).
Output	UML class diagram, UML activity diagram, UML sequence diagram, UML collaboration diagram, Textual description (area of concern).

Table 18.3: System Collaboration Model as described in the generic MAFIIA.

Figure 18.18 shows a conceptual UML class diagram which represents main concepts in the target system's domain. Concepts are shown as classes, while collaborations between the concepts are shown as associations and operations. Each concept is related to one or more health information systems, shown as colored rectangles behind each concept. The colored rectangles are not UML convention, but they indicate where the particular patient information is received, and therefore which health information systems CARDIAC's EOC-system has to include. The only concepts without any corresponding health information system are ***Referrals*** and ***Epicrisis***. No particular health information system handles these concepts. Referrals and epicrisis are usually messages which are manually handled.

The EOC-system should mainly consist of three parts; ***navigation caremap***, ***patient chart form*** and ***portal***. The EOC-system shall manage information about the ***patient care***, including information about the ***patient***, the patient's ***diagnosis***, ***measurements***, ***observations***, ***requisitions*** and ***responses***, ***referrals*** and ***epicrisis*** and ***medication prescriptions***. Figure 18.18 also shows that ***health personnel*** treats the patient in accordance with certain ***quality procedures***.

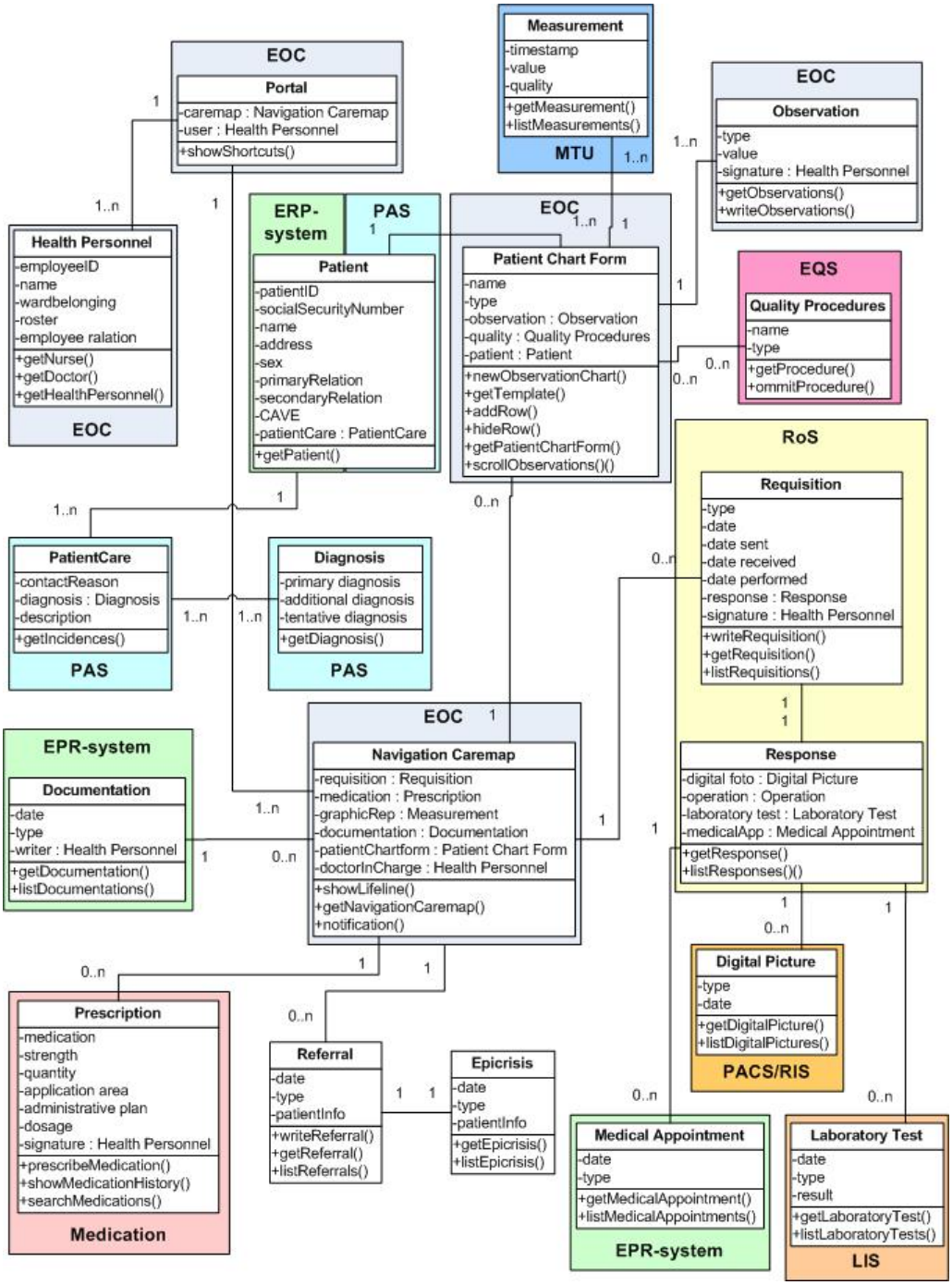


Figure 18.18: System Collaboration Model showing a set of collaborating components.

The System Collaboration Model is further detailed in Figure 18.19 and Figure 18.20, where the sequence diagram in Figure 18.20 is a continuation of the one in Figure 18.19. Because of lack of space the sequence diagram had to be divided in two. Figure 18.19 mainly displays interactions in the navigation caremap part, while Figure 18.20 mainly displays interactions in the patient chart form part.

Health personnel should access the EOC-system through the portal and choose a patient. Information for the particular patient will then be retrieved in the navigation caremap. The retrieval of the navigation caremap will involve:

- retrieving CAVE information from the EPR-system,
- retrieving biographical data, diagnosis and information about patient care incidences from PAS,
- retrieving observations and measurements from the patient chart form,
- retrieving responses to requisitions and referrals,
- retrieving textual documentation about the patient from the EPR-system,
- retrieving information about the patient's medication history from the Medication system,
- showing a lifeline over patient care incidences from birth to the present moment.

After retrieving the navigation caremap for the particular patient, health personnel should be able to zoom the lifeline in and out or scroll the lifeline backward and forward. This is shown in Figure 18.19 with the operations ***showLifeline(time)*** and ***scrollLifeline(time)***. Health personnel should also be allowed to write requisitions, referrals and prescriptions from the navigation caremap.

For each patient, health personnel should be allowed to configure one or several observation chart templates. This is shown in Figure 18.20. Measurements from MTU will either manually or automatically be registered in the observation chart. Health personnel should also be allowed to register other observations such as fluid intake, and to scroll the observation chart backward or forward. As mentioned earlier, health personnel will have to follow some quality procedures during the continuity of patient care.

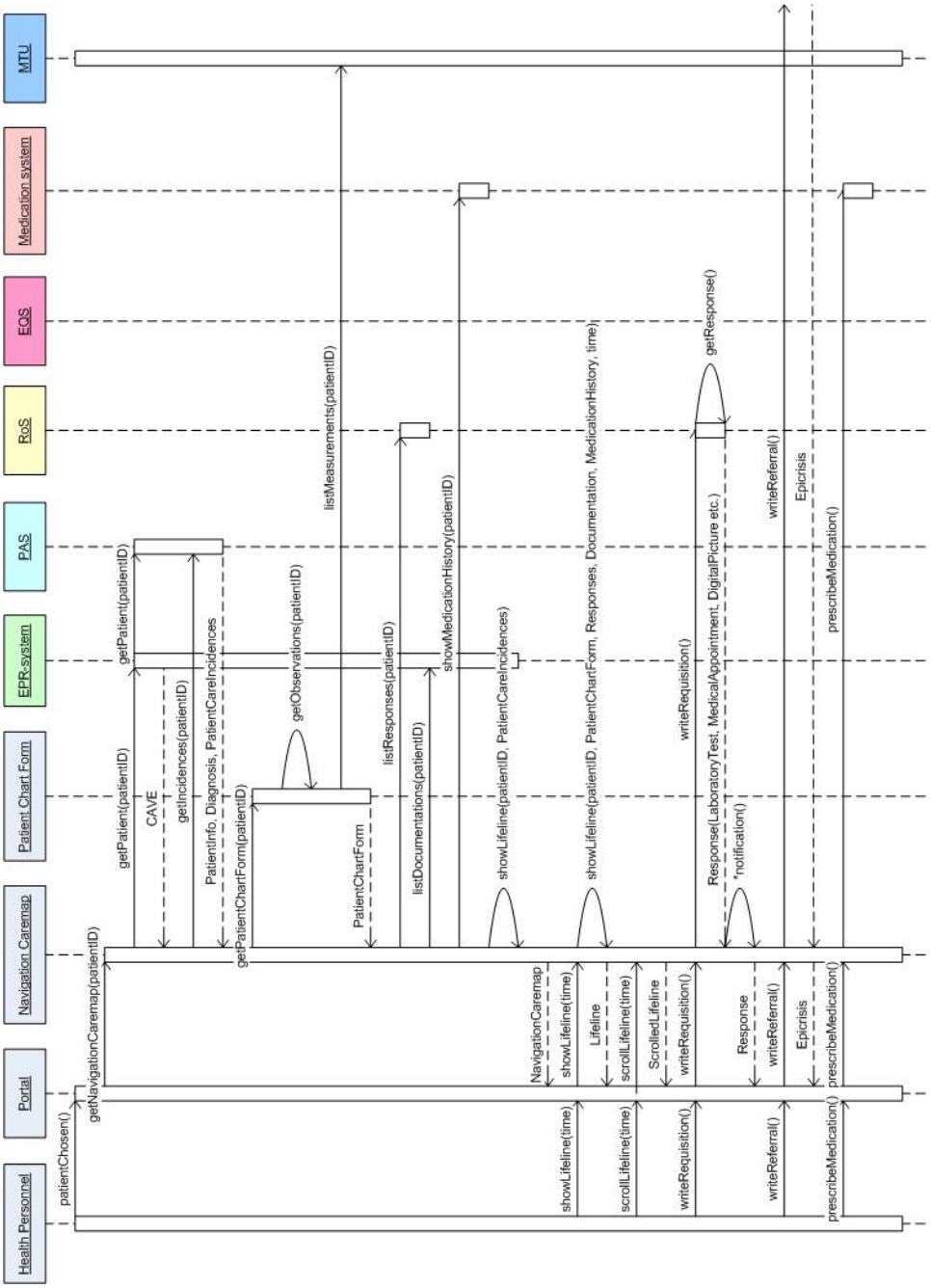


Figure 18.19: System Collaboration Model showing how components collaborate to perform main interactions in the navigation caremap.

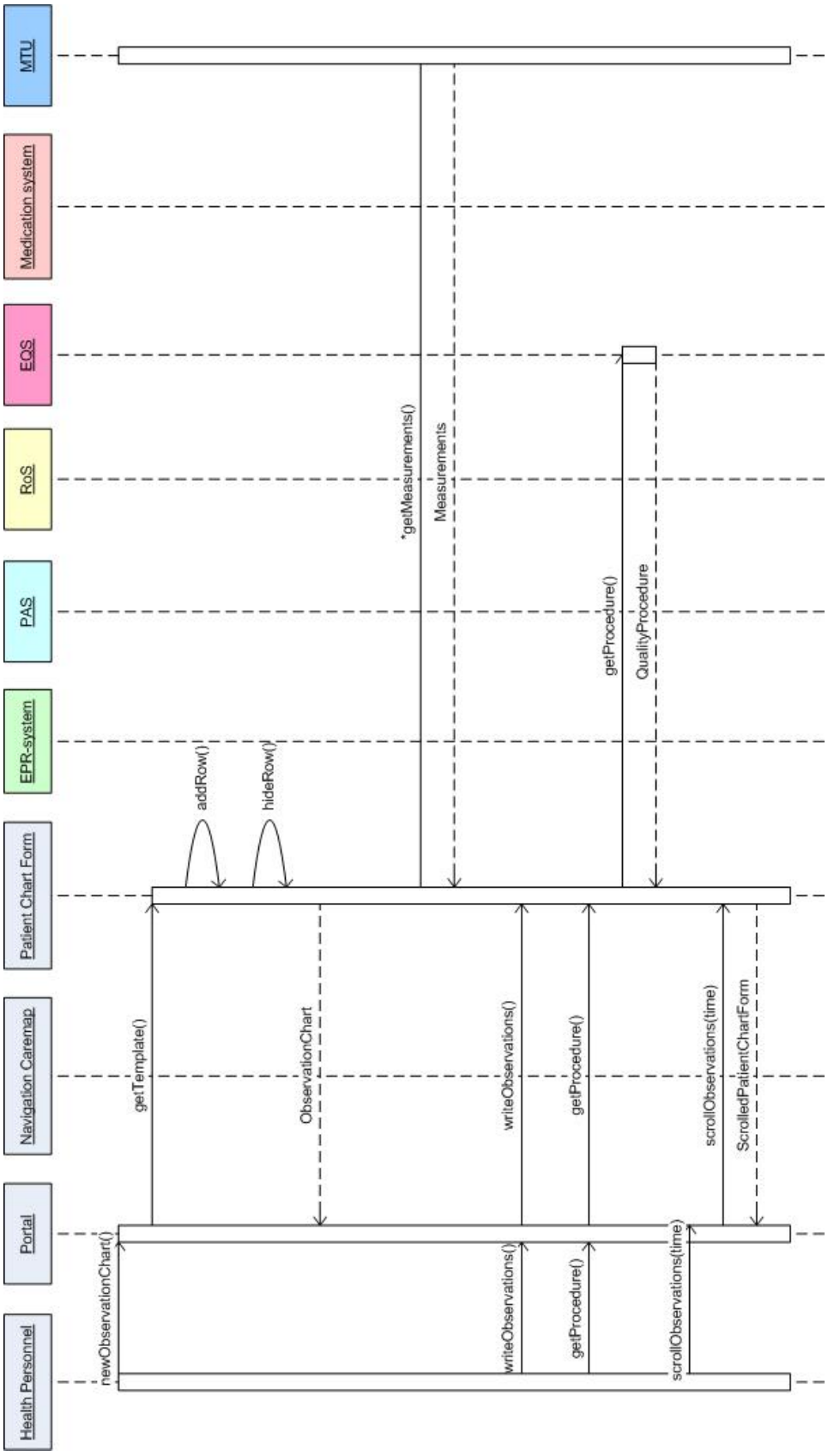


Figure 18.20: System Collaboration Model showing how components collaborate to perform main interactions in the patient chart form.

18.4 Component and Interface Specification Model

Model	**Component and Interface Specification Model**
Purpose	Describe each of the main components and interfaces of the target system, including operation signatures and behavior.
Input	Requirement viewpoint. System Decomposition Model, System Collaboration Model (Component viewpoint).
Output	UML and UMLsec class diagram, UML state chart diagram, Textual description.

Table 18.4: Component and Interface Specification Model as described in the generic MAFIIA and MAFIIA/RBAC.

Component and Interface Specification Model describes the interfaces between the main components in the target system. The diagrams in this model are based on UML 2.0 and the use of composite structures. UML 2.0 presents two concepts for describing interfaces between components, namely ports and interfaces. A short introduction to ports and interfaces is given below, while they are described further in Appendix C in Section C.9.1.

A port is a property which specifies a distinct interaction point between a component and its environment or between the internal ports within one component. A port may specify the services which the component provides/offers and the services which the component expects/requires.

An interface is a declaration of a set of public features and obligations. An interface specifies a contract - any instance that realizes the interface must fulfil that contract. The obligations may be constraints such as preconditions, post-conditions, protocol specifications or others.

Component and Interface Specification Model shows security, service and shortcut interfaces in the target system. The security, service and shortcut interfaces are shown in separate figures. These figures follow the same structure; the left side of the figures illustrates main components of CARDIAC's EOC-system, while the right side illustrates the environment systems.

The components on the left side represent different subsystems, which are already described in the System Decomposition Model in Section 18.2. ***RoleManager***, ***SSOManager*** and ***AccessManager*** components are defined in the Access Control subsystem. The ***AuditingManager*** component is depicted in the Auditing subsystem, while the ***UserAdm*** component is depicted in the Access Rights Administration subsystem. The ***ServiceIntegrator*** component is introduced in both the Navigation Caremap subsystem and the Patient Chart Form subsystem. The ***ShortcutHandler*** component

belongs to the Portal subsystem.

In cases where the interface(s) of the environment systems' classes does/do not match the needed interface, Adapter pattern should be used. This is not shown in any of the following figures because it complicates the figures. Still, the Adapter pattern should be adapted on the components ***Security***, ***Services*** and ***Shortcuts***, respectively shown in Figure 18.21, Figure 18.22 and Figure 18.23. Adapter pattern is described in Appendix B.

Figure 18.21 shows the main components and interfaces regarding security in the EOC-system. Each environment system, shown on the right side of the figure, provides some security services through the interface ***EnvironmentSecurity***. They are required to provide information about their users and access control mechanism.

The ***Security*** component, shown in the middle of the figure, utilizes the EnvironmentSecurity interface. In addition, it offers target system components interfaces for the retrieval of audit logs, ***AuditLog***, and external user information, ***ExternalUsers***, through the ***Security*** port. Single sign-on and access control logic are also provided through the Security port.

The target system components ***RoleManager***, ***SSOManager***, ***AccessManager***, ***AuditingManager*** and ***UserAdm*** utilize interfaces on the Security port.

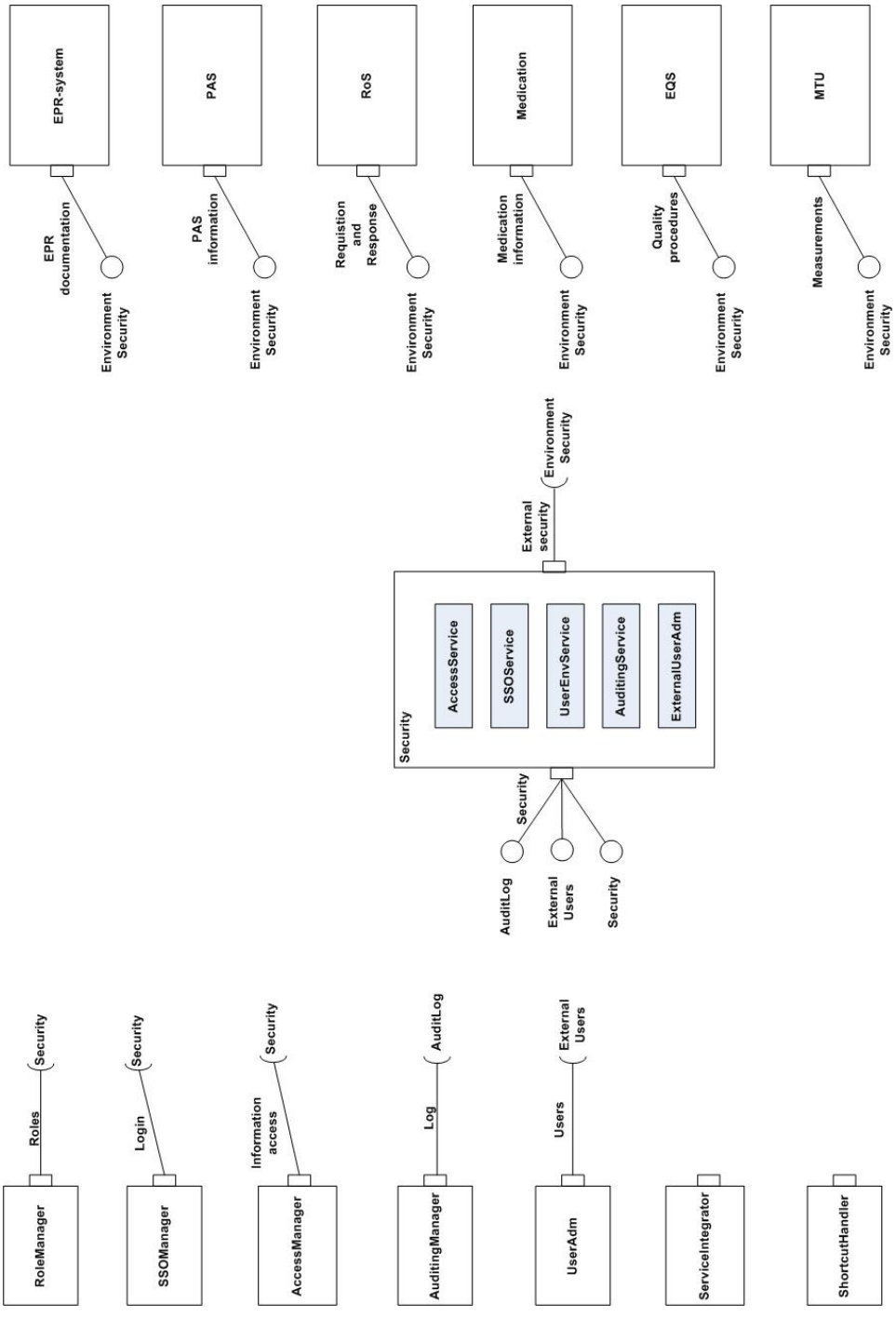


Figure 18.21: Component and Interface Specification Model showing how security is handled in the EOC-system.

Figure 18.22 gives an overview of the main functionality provided in the target system. Information is retrieved from the environment systems, and users are allowed to access RoS and Medication system when writing requisitions and prescribing medication. When information is retrieved or when RoS or Medication system are accessed, single-sign on is required. In addition, the access control within the respective environment system has to be ensured and an audit log has to be generated.

In Figure 18.22, all environment systems provide services for information retrieval through their ***DataRetrieval*** interface. RoS and Medication also provide a service for writing requisitions or for prescribing medication, respectively. These services are provided through the ***UI*** interface.

Services provided by the environment systems are utilized by the ***ServiceIntegrator***, accessed through the ***UIService*** and ***RetrievalService*** interfaces provided by the ***Service*** component.

The Service component also ensures single sign-on, access control and auditing through its ***InternalSecurity*** port. Single sign-on, access control and auditing is respectively provided by the ***SSOManager***, ***AccessManager*** and ***AuditingManager***.

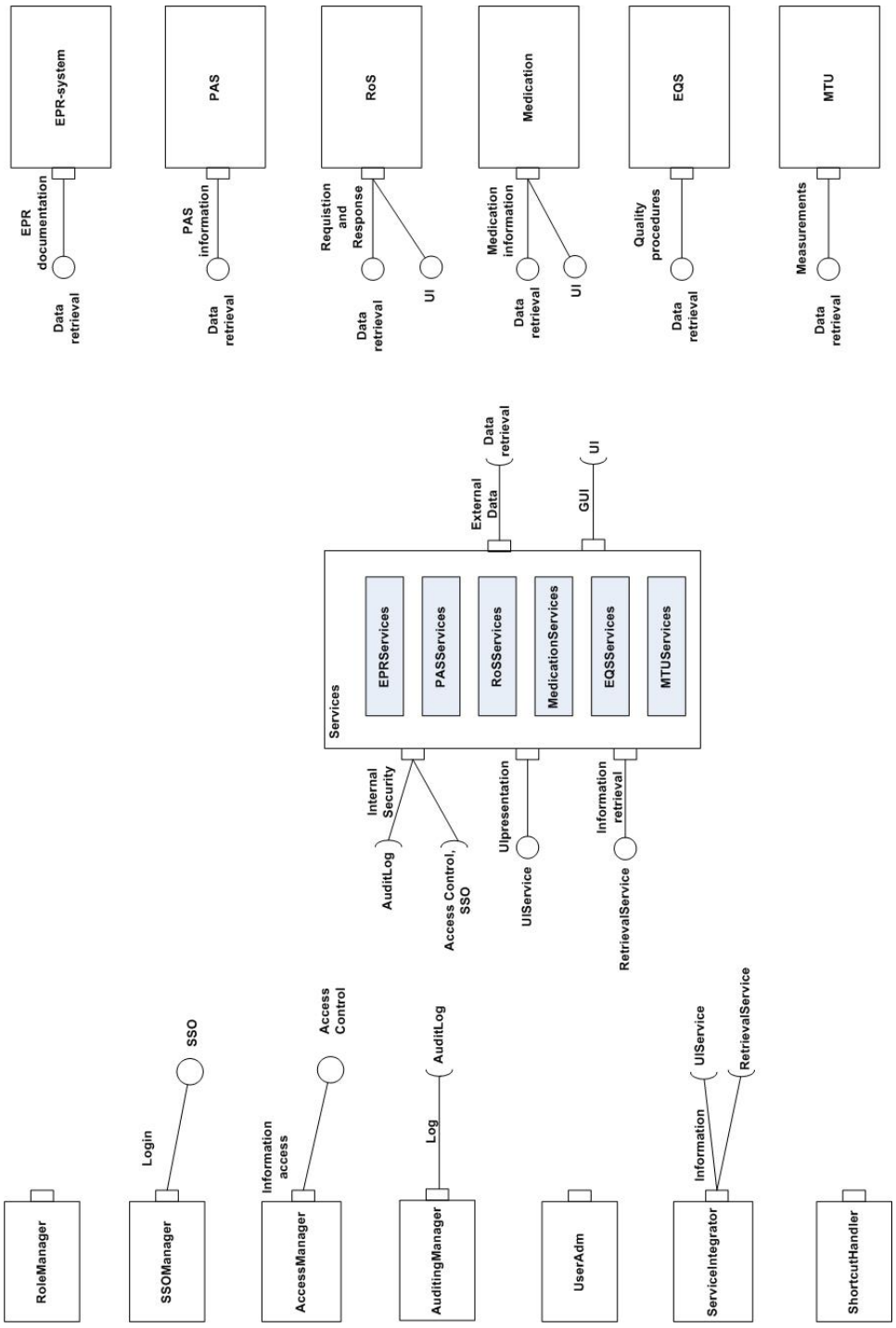


Figure 18.22: Component and Interface Specification Model showing how integration of services is handled in the EOC-system.

As mentioned earlier, it shall be possible to access environment systems through the Portal subsystem, described in Section 18.2.3. The Portal subsystem provides shortcuts to the environment systems through its ***ShortcutHandler*** component. When shortcuts are used, single sign-on to the environment systems is required. Single sign-on is provided by the ***SSOManager*** component in the Access Control subsystem, described in Section 18.2.4.

Figure 18.23 gives an overview of how ShortcutHandler and SSOManager interface the ***Shortcuts*** component in order to access the environment systems. The environment systems have to provide their user interface to the Shortcuts component in order to enable shortcuts.

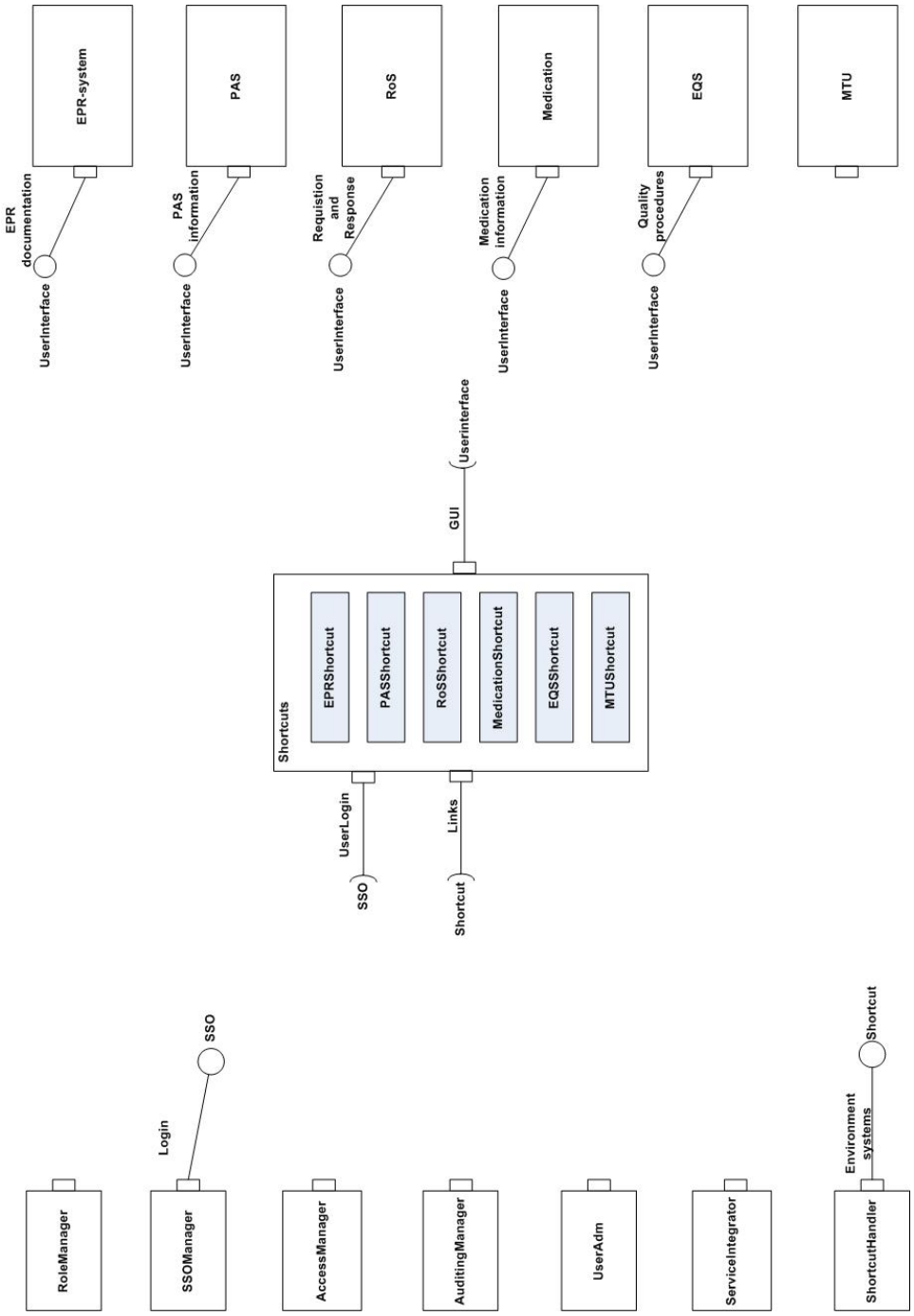


Figure 18.23: Component and Interface Specification Model showing shortcuts to environment systems is the EOC-system.

18.5 System Security Model

Model	**System Security Model**
Purpose	Describe how security concerns are handled by the components.
Input	Requirement viewpoint. System Decomposition Model (Component viewpoint).
Output	UML and UMLsec class diagram, UML sequence diagram, UML state chart diagram, Textual description.

Table 18.5: System Security Model as described in MAFIIA for IIS.

Stereotypes from the reference architecture are UserInterface, UserService, BusinessService, ResourceService and EnvironmentInterfacing. Further generalization of these stereotypes is shown in Figure 18.24.

The ***UserService*** stereotype is a supertype with ***LoginControlled*** and ***AuditingControlledUS*** as subtypes. The key idea with generalization is that everything about the UserService - associations, attributes, operations - is true also for LoginControlled and AuditingControlledUS. Because AuditingControlledUS is a subtype of LoginControlled everything about the LoginControlled is true for AuditingControlledUS.

Similarly, everything about the ***BusinessService*** stereotype is true for ***AccessControlled***, ***AuditingControlledBS*** and ***DigitallySigned***. In addition, everything about AccessControlled is true for AuditingControlledBS and for DigitallySigned, and so on.

The ***EnvironmentInterfacing*** stereotype is a supertype for ***SSOFulfilled***, ***EnvACFulfilled*** and ***EnvAuditing***, which in turn are supertypes for ***EnvSecured***. The EnvironmentInterfacing stereotype is also a supertype for the ***ApplServer*** stereotype.

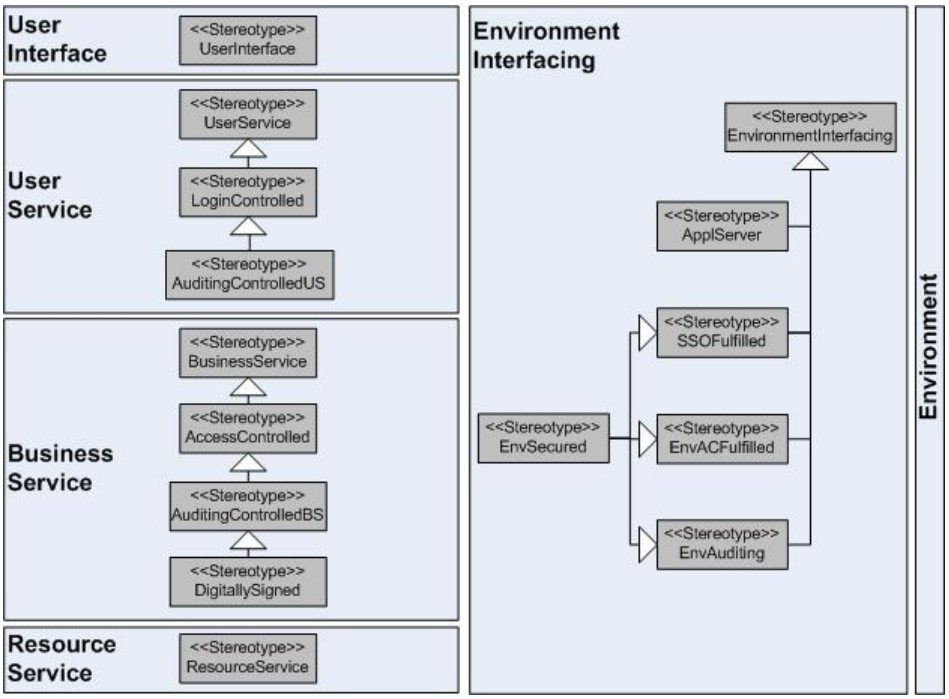


Figure 18.24: Generalization of reference architecture stereotypes.

Stereotypes are used to show how security concerns are handled by the components described in the System Decomposition Model in Section 18.2. The figures in the following sections, Figure 18.25, Figure 18.26 and Figure 18.27, indicate the semantics of the stereotypes introduced in the figure above. After these stereotypes are defined, they are used to show how the subsystems Navigation Caremap, Patient Chart Form, Portal and Access Rights Administration handle security related concerns. This is shown in Figure 18.28 - Figure 18.31.

18.5.1 Access Control

In Figure 18.25 stereotypes for controlling login and access are defined.

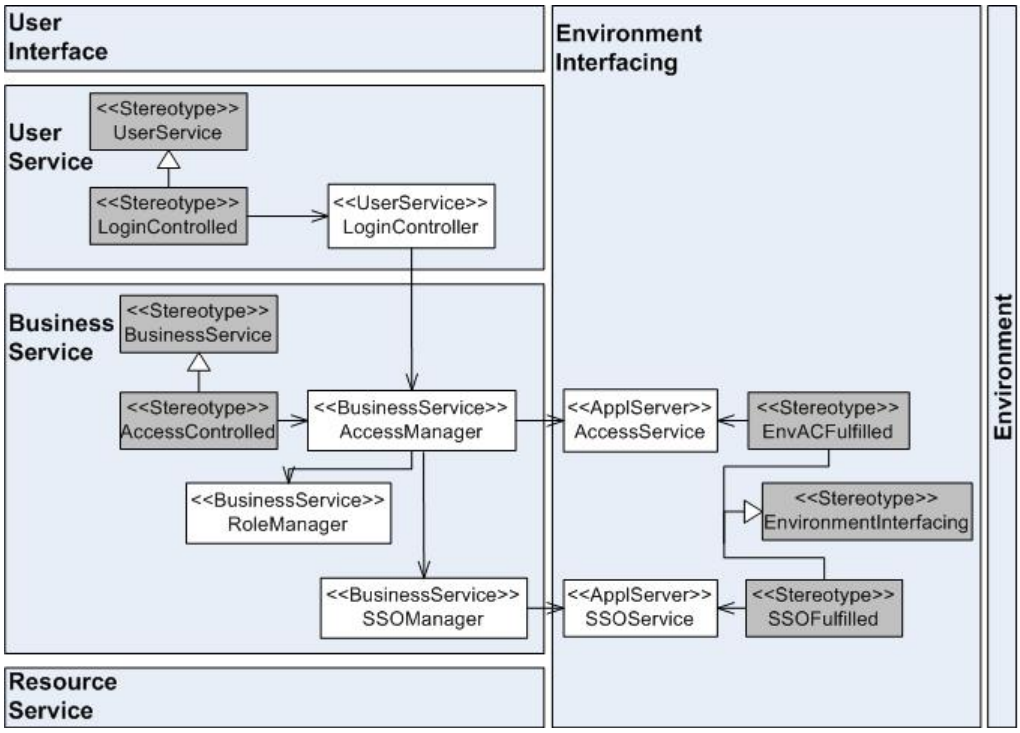


Figure 18.25: Definition of stereotypes for UserService, BusinessService and EnvironmentInterfacing components which employ components in the Access Control subsystem.

LoginControlled defines a stereotype for UserService components which utilizes the login or authentication mechanism in the target system, LoginController. Components stereotyped with LoginControlled provide for controlled and valid logins.

AccessControlled defines a stereotype for BusinessService components which utilizes the access control mechanism in the target system, which in this particular case also includes the access control mechanisms in the environment systems. AccessControlled provides for controlled and legitimate accesses according to the underlying access control mechanisms.

EnvACFulfilled defines a stereotype for EnvironmentInterfacing components which ensures the access control mechanism within the particular environment system.

SSOFulfilled also defines a stereotype for EnvironmentInterfacing components. Components stereotyped with SSOFulfilled provide for single sign-on.

18.5.2 Auditing

In Figure 18.26 stereotypes for auditing are defined.

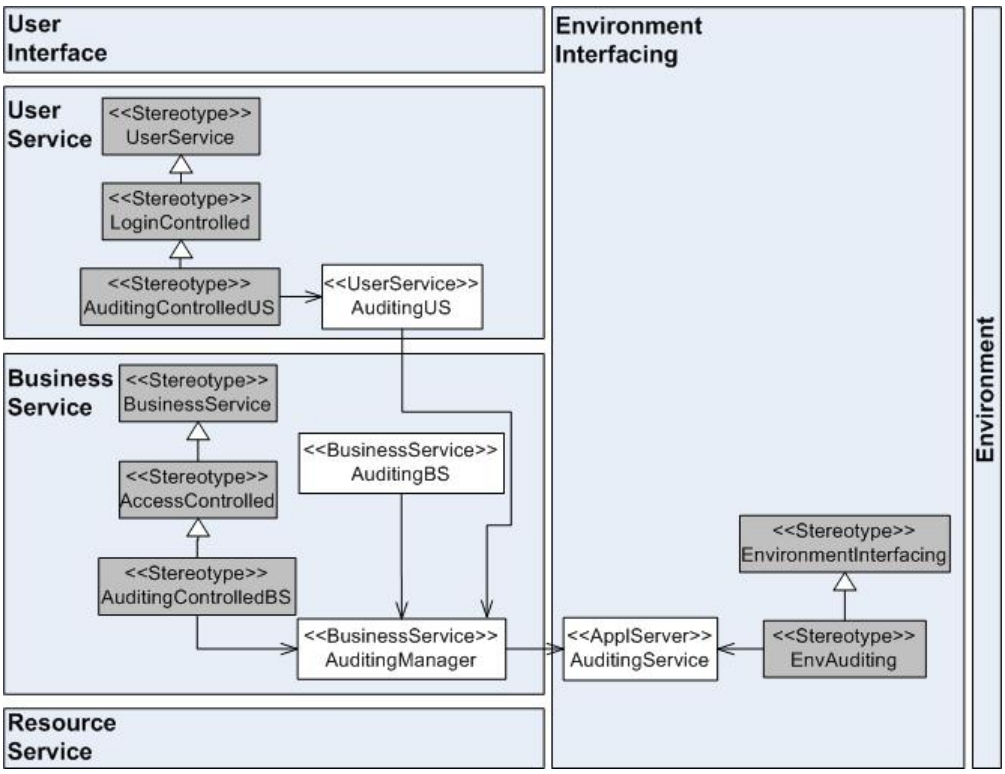


Figure 18.26: Definition of stereotypes for UserService, BusinessService and EnvironmentInterfacing components which employ components in the Auditing subsystem.

AuditingControlledUS defines a stereotype for UserService components which provides for auditing of all logins.

AuditingControlledBS is similar to AuditingControlledUS, but it is defined on the Business Service tier. All BusinessService components which read, update or make insertions in the EOC have to be stereotyped with AuditingControlledBS.

Components stereotyped with ***EnvAuditing*** ensure that auditing is performed by the consumed service.

18.5.3 Digital Signature

In Figure 18.27 a stereotype for digital signing is defined.

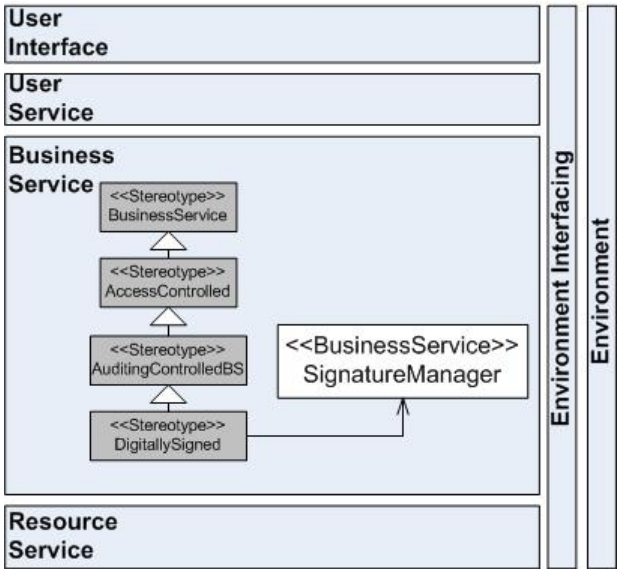


Figure 18.27: Definition of stereotypes for BusinessService components which employ components in the Digital Signature subsystem.

DigitallySigned defines a stereotype for BusinessService components which provide for digital signing when information is edited or registered in the EOC.

18.5.4 Navigation Caremap

Figure 18.28 shows how the Navigation Caremap subsystem handles security related concerns such as single sign-on, access control, auditing and digital signing.

All logins to the Navigation Caremap subsystem have to be audited. This is shown by the AuditingControlledUS stereotype, which is a subtype of LoginControlled, meaning that AuditingControlledUS ensures both controlled login and auditing.

Accesses performed by the InfoIntegrator or the ServiceIntegrator need to be controlled, but not necessarily audited, hence stereotyped with AccessControlled. In addition to ensuring controlled access to the information resources and services the TransformationHandler and the SelectionHandler access, auditing of the operations they perform is needed. Transformations also need to be digitally signed, hence TransformationHandler is stereotyped with DigitallySigned, while SelectionHandler is stereotyped with only AuditingControlledBS.

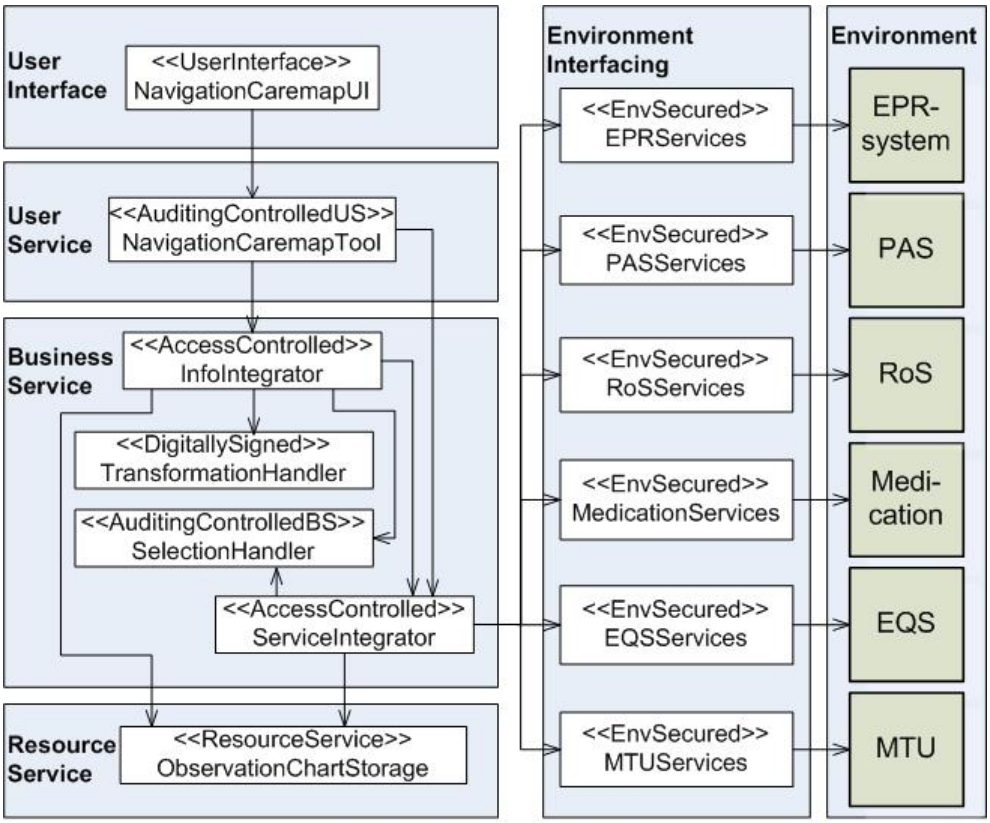


Figure 18.28: The Navigation Caremap subsystem handles security related concerns such as single sign-on, access control, auditing and digital signing.

Single sign-on needs to be fulfilled before any environment system service is accessed. Accessing these services also requires that the access control mechanism within the particular environment system is ensured and that auditing by the accessed service is done. EnvironmentInterfacing components requesting environment system services are therefore of stereotype ***EnvSecured***. EnvSecured is a generalization of the stereotypes SSOFulfilled, EnvACFullfilled and EnvAuditing, meaning that it ensures single sign-on, access control and auditing in the environment systems.

18.5.5 Patient Chart Form

Figure 18.29 shows how the Patient Chart Form subsystem ensures single sign-on, auditing and controlled access to resources and environment systems.

Similar to the Navigation Caremap subsystem, all logins to this subsystem have to be controlled and audited. The components InfoIntegrator and ServiceIntegrator are stereotyped with AccessControlled, while Trans-

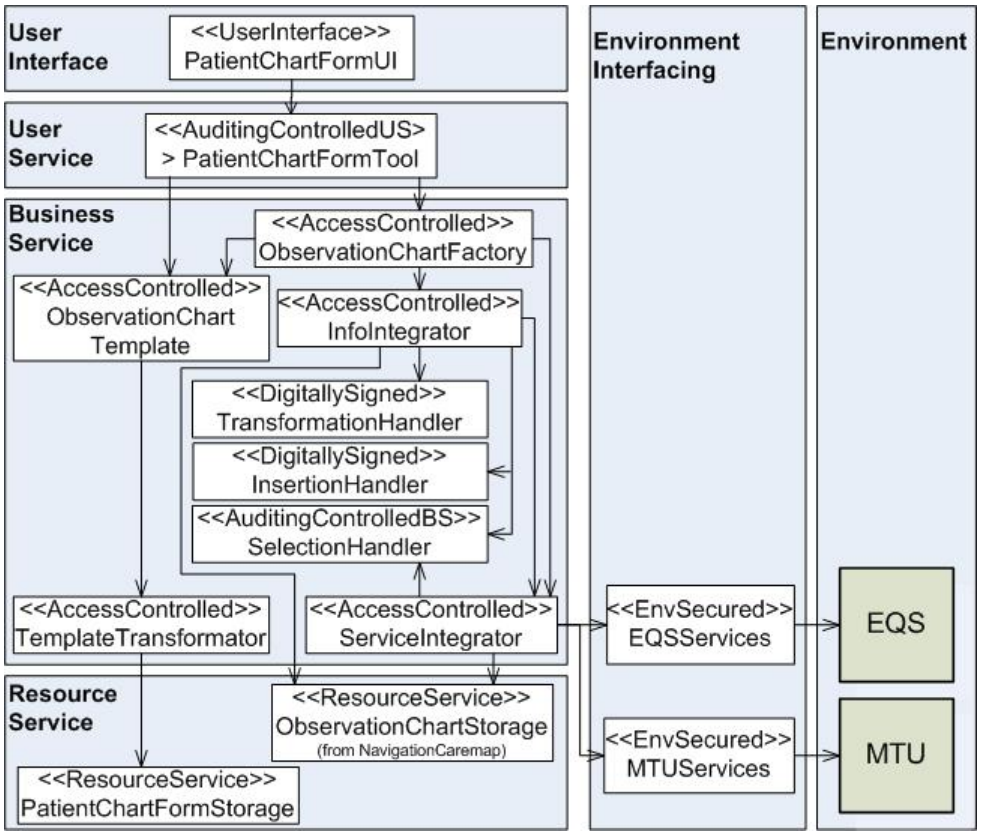


Figure 18.29: The Patient Chart Form subsystem ensures single sign-on, controlled access to resources and environment systems and auditing.

formationHandler is stereotyped with DigitallySigned and SelectionHandler is stereotyped with AuditingControlledBS.

Unlike the Navigation Caremap subsystem, which is read-only, the Patient Chart Form subsystem supports insertions and template configurations. Before any insertions are allowed, the observation chart has to be configured. Access rights have to be controlled before allowing users to do this. Hence, the components ObservationChartFactory, ObservationChartTemplate and TemplateTransformator are stereotyped with AccessControlled. The InsertionHandler is stereotyped with DigitallySigned because insertions have to be authorized and audited, in addition to being digitally signed.

Similar to the Navigation Caremap subsystem, EnvironmentInterfacing components are stereotyped with EnvSecured which ensures single sign-on, the access control mechanism within environment systems and auditing by the invoked service.

18.5.6 Portal

Figure 18.30 shows how the Portal subsystem ensures controlled access to Navigation Caremap and Patient Chart Form and single sign-on to environment systems.

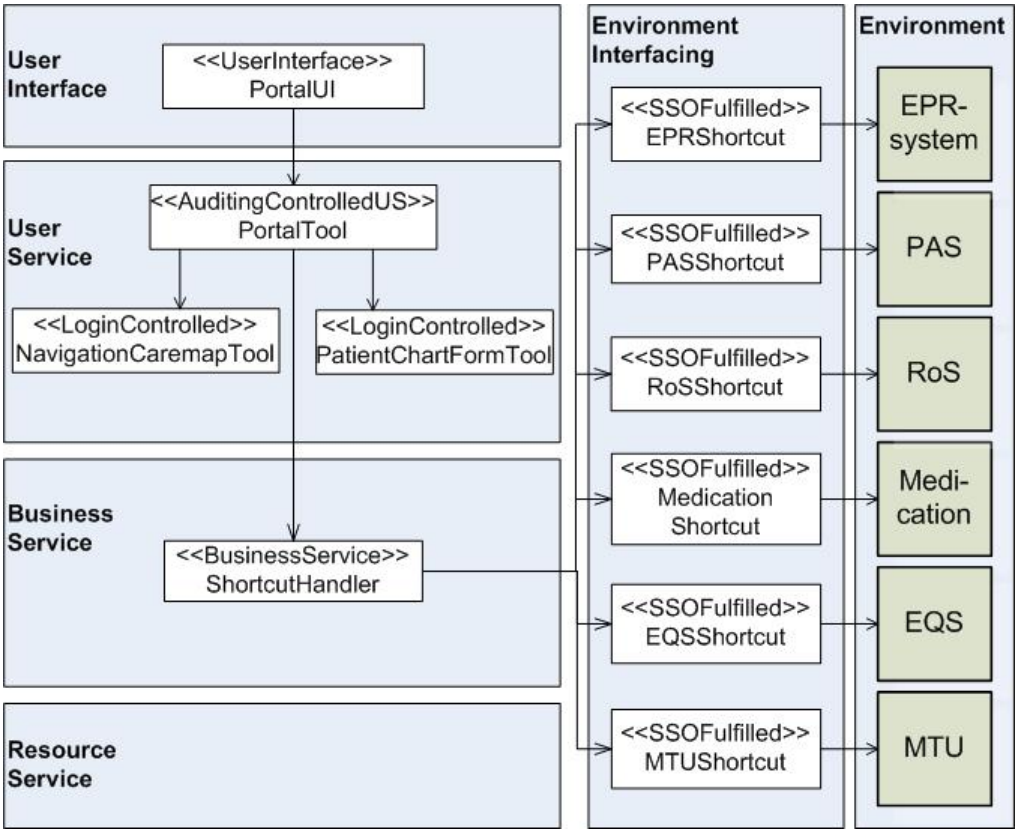


Figure 18.30: The Portal subsystem ensures controlled access and single sign-on to Navigation Caremap, Patient Chart Form and environment systems.

All logins to the Portal subsystem, which is the user interface in the target system, have to be controlled and audited. Logins to the Navigation Caremap subsystem and the Patient Chart Form subsystem need to be controlled, not audited. Further access control and auditing are handled within each of these subsystems, as described in the two previous sections.

The Portal provides shortcuts to environment systems, and these shortcuts provide automatic logins. Hence, the EnvironmentInterfacing components are stereotyped with SSOFulfilled. Access control and auditing are then handled by each of the environment systems.

18.5.7 Access Rights Administration

Figure 18.31 shows how the Access Rights Administration subsystem ensures controlled login and access.

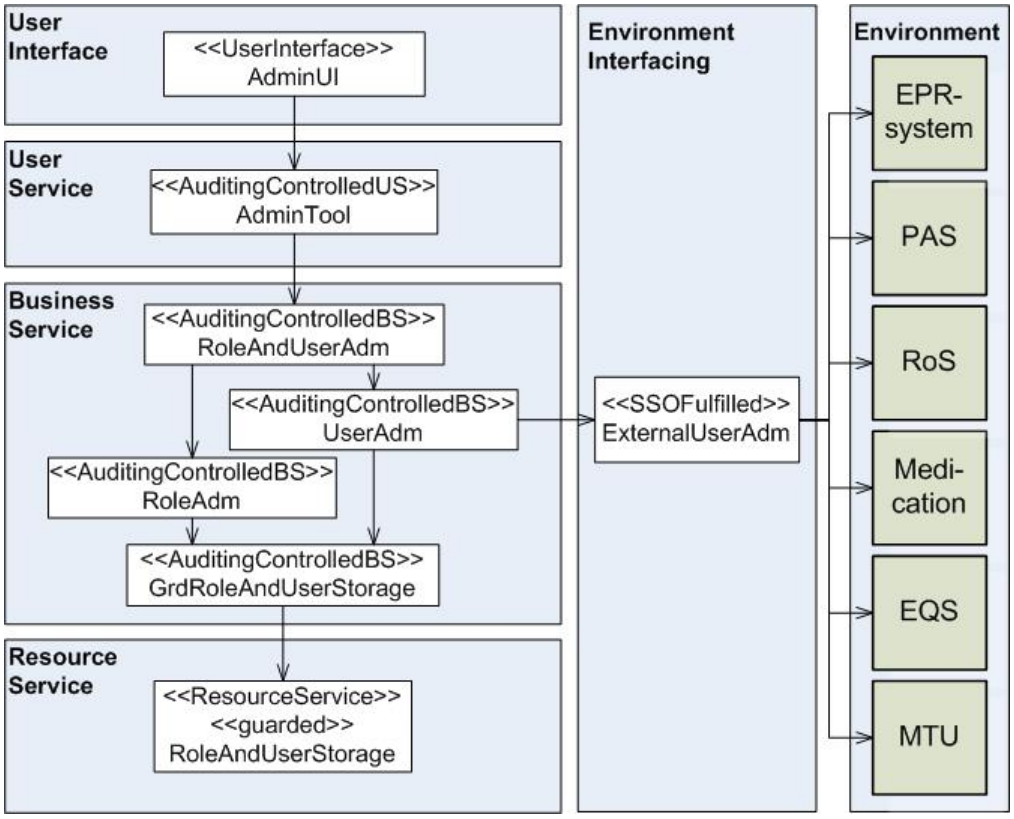


Figure 18.31: The Access Rights Administration subsystem ensures controlled login and access.

In the Access Rights Administration subsystem, controlled login and access are required in addition to auditing of all insertions, updates and deletions. Hence, UserService components and Business Service components are stereotyped with AuditingControlledUS and AuditingControlledBS, respectively. The EnvironmentInterfacing component ExternalUserAdm requires single sign-on to environment systems. This component is therefore stereotyped with SSOFulfilled.

18.6 System Access Control Model

Model	**System Access Control Model**
Purpose	Describe the relationship and properties of components in the system that form the access control-subsystem and how they enforce the security policy.
Input	Business Aspects Model, Requirement viewpoint, System Decomposition Model, System Collaboration Model.
Output	UMLsec activity diagrams, UMLsec use case diagram, UML state chart, Textual description.

Table 18.6: System Access Control Model as described in MAFIIA/RBAC.

MAFIIA/RBAC defines a model with special concern on access control. Since this thesis also has focus on other security mechanisms in addition to access control, this model will not be taken into account here. Instead, access control is described in the System Decomposition Model through the subsystems Access Control and Access Rights Administration respectively described in Section 18.2.4 and Section 18.2.6.

18.7 Summary

In this viewpoint, the target system has been decomposed into subsystems and components. Each subsystem or component is described in relation to their function, collaboration with other subsystems or components and to the security mechanisms they ensure.

The components identified in this viewpoint will, after further preparations in the distribution and realization viewpoint, be the basis for the implementation of the architectural description.

Chapter 19

Distribution Viewpoint

The distribution viewpoint describes the logical distribution of system components. This viewpoint also documents which components must be separated and which cannot.

The following models are part of the distribution viewpoint:

- System Distribution Model describes the logical distribution of all components that are a part of the target system.
- Role Distribution Model documents the distribution of stakeholders.
- System Security Model discusses the distribution of security related components.

19.1 System Distribution Model

Model	**System Distribution Model**
Purpose	Shall describe logical units or components that must be distributed and deployed together.
Input	System Decomposition Model, System Collaboration Model (Component viewpoint).
Output	UML and UMLsec deployment diagram, Textual description (rationale).

Table 19.1: System Distribution Model as described in the generic MAFIIA and MAFIIA/RBAC.

Figure 19.1 is a UML deployment diagram showing the logical distribution of components in the target system, independent of the tiers in the reference architecture.

User interface components and some user logic are hosted on the ***Client***. ***PortalUI*** is dependent on both ***NavigationCaremapUI***, ***PatientChartFormUI*** and ***LoginUI*** to form the user interface of the EOC-system. PortalUI is also dependent on ***PortalTool*** for the purpose of user logic behind the user interfaces.

Figure 19.1 shows that the Single access point pattern is chosen to create only one way to get into the EOC-system. The PortalTool serves as the single access point, and this component verifies the information collected in LoginUI in a check point. PortalTool is dependent on the ***LoginController*** for authentication verification. The Check point pattern is therefore used with LoginController as authentication check point. The Single access point pattern and the Check point pattern are described in Appendix B.

The Client, or more precisely the ***PortalTool*** component, communicates with the ***Application Server*** through its components ***NavigationCaremapTool*** and ***PatientChartFormTool***. These two components are dependent on the ***AccessManager*** for controlled access to resources. The AccessManager can be seen as a check point for authorization, indicating that the Check point pattern is used here as well.

The Application Server hosts all system logic. The components ***InsertionHandler***, ***TransformationHandler*** and ***SelectionHandler*** are dependent on ***AuditingBS*** because auditing should be performed during information registration, editing and retrieval. Other dependencies within the Application Server are shown in the System Decomposition Model in Section 18.2.

The Application Server communicates with the Security Server for the purpose of access control, auditing and digital signing. It also communicates with its database, ***SystemDB*** and environment systems for the purpose of information retrieval.

The ***Environment System*** node is different from the rest of the nodes in Figure 19.1, because it includes all environment systems and environment system components. Similar to the target system, each environment system also has its own internal distribution.

The ***Security Server*** hosts all security logic which includes access control, auditing, digital signing and access rights administration.

SecurityDB and ***AuditingDB*** are depicted in Figure 19.1. Components within the SecurityDB node, belong to single sign-on, access control, access rights administration and digital signature components hosted in the Security Server, while the component within the AuditingDB belongs to auditing components hosted in the Security Server.

The ***AdminClient*** hosts user interface logic for the administration of users and access rights. It communicates with the Security Server for the purpose of accessing the ***RoleAndUserStorage*** through ***RoleAndUserAdm***, ***UserAdm*** or ***RoleAdm***.

Links between the different nodes indicate communicating nodes.

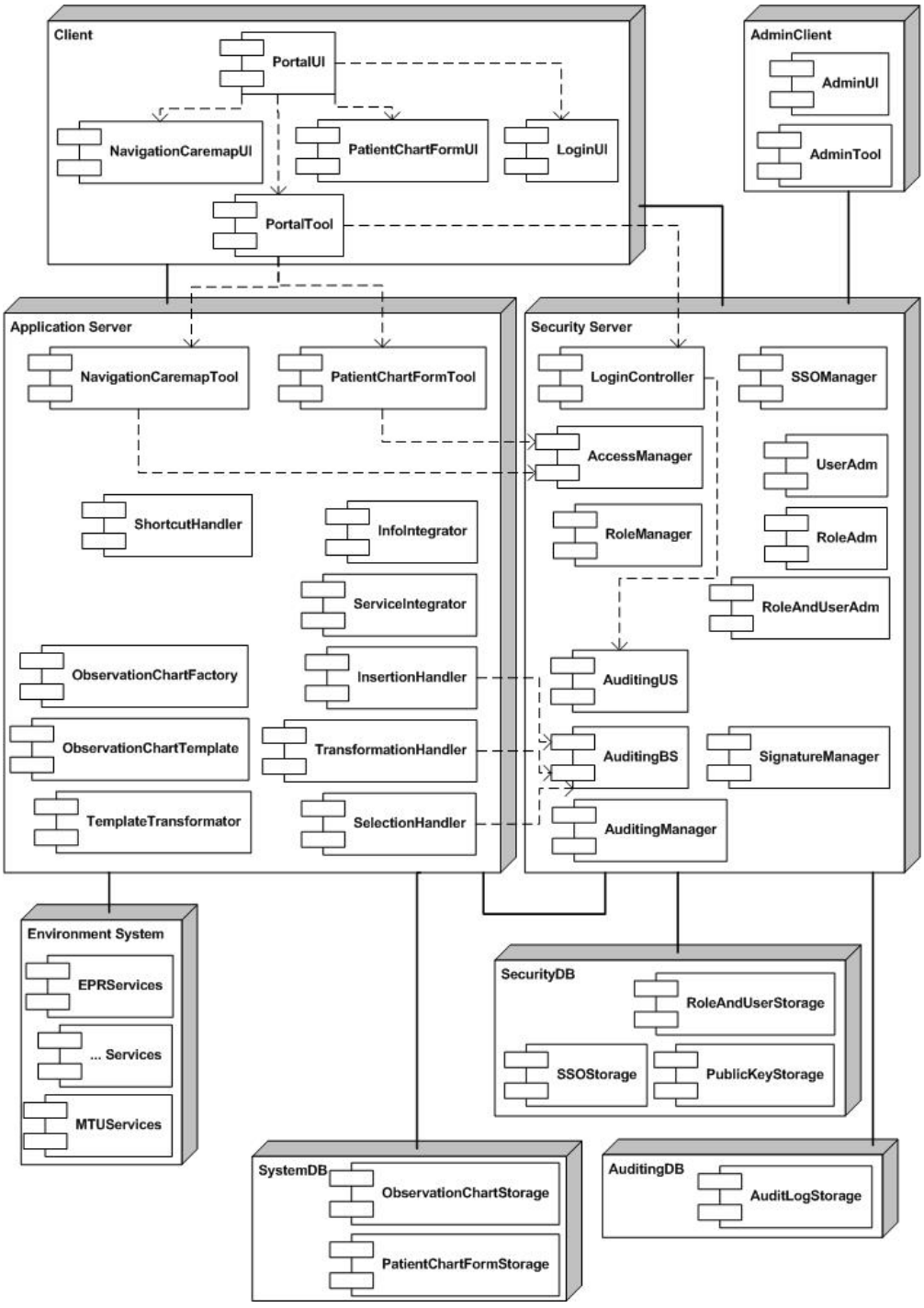


Figure 19.1: System Distribution Model showing the logical distribution of components.

19.2 Role Distribution Model

Model	**Role Distribution Model**
Purpose	Describe the distribution of the different roles that are part of the target system.
Input	Business Aspects Model (Context viewpoint), System Distribution Model (Distribution viewpoint).
Output	UML deployment diagram, Textual description.

Table 19.2: Role Distribution Model as described in the generic MAFIIA.

For the Role Distribution Model, three roles are identified:

- System user
- Owner of the EOC-system
- Administrator of the EOC-system

System users are health personnel in one particular hospital within the Health Region for Central Norway. System users shall not have access to the EOC-system outside the hospital network. Hence, system users are only distributed inside a hospital.

The owner and administrator of the EOC-system is probably HEMIT.

How the administration of the environment systems is organized is outside the scope of this thesis. However, the environment systems are most likely owned by HEMIT.

19.3 System Security Model

Model	**System Security Model**
Purpose	Describe the effects of the security concern on the other models defined in the system distribution viewpoint.
Input	System Security Model (Component viewpoint)
Output	UML use case diagram, UML collaboration diagram, Textual description.

Table 19.3: System Security Model as described in MAFIIA for IIS.

System Security Model gives a suggestion for how components identified in the System Distribution Model can be distributed in order to ensure the

security within the target system. For better understanding of the target system and to get an overview of its components, the security components are separated from other system components in the System Distribution Model, shown in Figure 19.1. Security logic components are grouped into one logical node, while security relevant information and audit logs are logically separated onto two different nodes.

The Security Server should be separated from the Application Server primarily because of reliability requirements. This separation will also provide for compartmentalization, i.e. the possibility to limit access on different levels between the Application Server and the Security Server. Thus, the Security Server comes behind another barrier in relation to the Application Server, and defense in depth is practiced. The practice of defense in depth also requests protection through firewalls and encryption. In this particular case, a corporate wide firewall should keep intruders out of the hospital network, while encryption should protect information that travels across this network.

Distribution of components into several nodes also prevents that the entire system is compromised if only parts of the system are compromised. Separating security components from other system logic components, also ensures special control on security.

For best possible protection of the system, the physical distribution of security components is essential. A suggestion for how the security components can be distributed physically is shown in Figure 20.1 in the realization viewpoint.

19.4 Summary

The models in the distribution viewpoint describe how the system components are logically distributed, independent of existing infrastructure, which is considered in the realization viewpoint in the next chapter.

Chapter 20

Realization Viewpoint

The realization viewpoint describes how the target system should be implemented and deployed into its environment. Although the distribution viewpoint is the input for this viewpoint, there is not necessarily a direct mapping between these two viewpoints. The distribution viewpoint describes logical distribution of components, while the realization viewpoint describes physical distribution of components. For example, two logically distributed components can be deployed at the same physical node. However, two components not logically distributed should be deployed at the same node.

The realization viewpoint requires three different models to describe the realization of the target system:

- System Deployment Model shows physical relationships among software and hardware components in the target system.
- Technology Mapping Model specifies how components are related to or implemented by technology solutions.
- System Integration Test Model describes a set of test scenarios.

Only a brief description of the System Deployment Model and Technology Mapping Model will be given in the subsequent sections. Realization of the target system is not the main focus in this thesis. The aim is to create a security focused integration architecture for CARDIAC's EOC-system.

For a proper description of the realization viewpoint, there is a need for extensive system documentation on the health information systems which shall be integrated with the EOC-system. This information has not been available for this project.

20.1 System Deployment Model

Model	**System Deployment Model**
Purpose	The purpose of this model is to describe the set of system deployment configurations.
Input	System Distribution Model (Distribution viewpoint), Requirement Model (Requirement viewpoint).
Output	UML deployment diagram, Textual description.

Table 20.1: System Deployment Model as described in the generic MAFIIA.

Input to System Deployment Model is System Distribution Model, and the nodes identified in this model are physically distributed in the System Deployment Model. Figure 20.1 shows how the target system is distributed onto physically nodes. Even if the System Distribution Model shows that the Client and ClientAdmin are distributed onto separate nodes, the realization of the target system allows them to be deployed on one and the same node, the ***Web Server***.

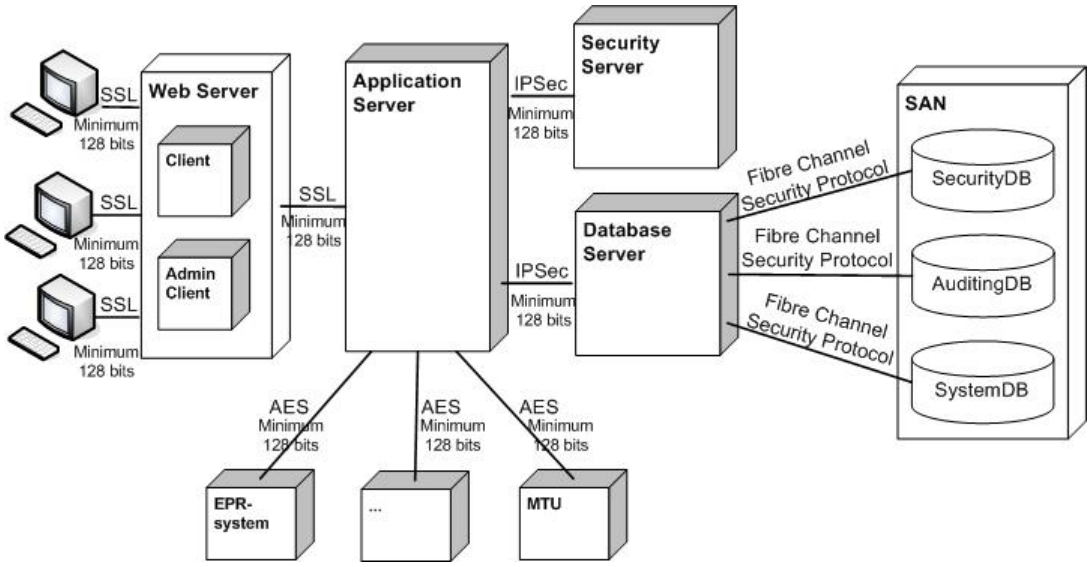


Figure 20.1: System Deployment Model showing how the target system is physically distributed onto nodes.

Since today's database solutions provide separation of databases within the same machine, all database nodes are physically distributed onto one node, the ***SAN***, in Figure 20.1. A database solution for the EOC-system will most likely be a Storage Area Network (SAN), meaning that the actual

information stored in SecurityDB, AuditingDB and SystemDB will be localized on the same storage solution (SAN) with different zones so that they are separated.

The ***Database Server***, most likely Oracle in a system with this calibre, contains the database software for processing information stored in the SAN.

The ***Application Server*** fetches information from the Database Server, processes the information and presents it to the ***Web Server*** where it becomes available for the users.

As described in the System Security Model in Section 19.3, defense in dept through encryption should be practiced. In addition, requirement R2.2.1.2.3 in the Requirement Model requests that the EOC-system provides secure transmission of information, i.e. encryption, between entities participating in the communication.

This may be solved by using Secure Socket Layer (SSL) for encryption of communication between clients and the Web Server and between the Web Server and the Application Server in the network. Encryption between the Application Server and the Security Server, and encryption between the Application Server and the Database Server may be solved by IPSec. Encryption of XML data between the Application Server and the environment systems may be solved by the encryption algorithm Advanced Encryption Standard (AES). Since SANs most often use fibre channel technologies, the Fibre Channel Security Protocol is proposed for secure transmission of information between the Database Server and the SAN. Communication links are therefore labeled with either *SSL*, *IPSec*, *AES* or *Fibre Channel Security Protocol* in Figure 20.1.

The Data Inspectorate recommends an encryption algorithm with minimum 128 bits key or stronger when personal health data is transferred outside the data controller's control [56]. Thus, communication links are also labeled with *Minimum 128 bits* in the figure.

20.2 Technology Mapping Model

Model	**Technology Mapping Model**
Purpose	Shall describe how system components map to technological solutions, concepts and mechanisms.
Input	Requirement Model (Requirement viewpoint), Component and Interface Specification Model (Component viewpoint), System Distribution Model (Distribution viewpoint), System Deployment Model (Realization viewpoint).
Output	UML component diagram, UML deployment diagram, Textual description (rationale).

Table 20.2: Technology Mapping Model as described in the generic MAFIIA.

The Technology Mapping Model is not described in detail in this architectural description because implementation of CARDIAC's EOC-system is outside the scope of this thesis. In order to specify the mapping of components to technological solutions one also needs to have detailed knowledge of the environment systems and the technology platforms used in these systems. Still, some technology recommendations were given in the System Deployment Model. In addition, some technology solutions are certain.

CARDIAC aims to implement the EOC-system within the Health Region for Central Norway. This health region requests a service-oriented architecture where all health information systems offer functionality through Web services. The chosen technology is Microsoft .NET. Microsoft .NET is a Web-based middleware solution developed by Microsoft. It is described as a prefabricated infrastructure connecting information, people, systems and devices [63].

The target system should also be based on the IMATIS Platform, which is a middleware platform developed by CARDIAC. The IMATIS Platform was described in Section 6.6.

20.3 System Integration Test Model

Model	**System Integration Test Model**
Purpose	Shall describe a set of test scenarios to be conducted during system deployment (subsystem integration).
Input	Requirement Model (Requirement viewpoint), System Deployment Model (Realization viewpoint).
Output	Textual descriptions.

Table 20.3: System Integration Test Model as described in the generic MAFIIA.

System Integration Test Model is outside the scope of this thesis because it is too implementation specific.

20.4 Summary

The realization viewpoint is partly outside the scope of this thesis because it shall describe the realization of the target system. Still, a suggestion of the physical distribution of components is given. Additional, some of the required technology solutions are mentioned.

Chapter 21

Summary

Security and integration are highlighted throughout this architectural description by decomposing the EOC-system into subsystems and components. The EOC-system is divided into seven subsystems. Each of them is described throughly in the component viewpoint.

For example, one of the security related subsystems, the Access Control subsystem, contains business logic for the management of role-based and context-based access control within the EOC-system. This is actually handled by the components AccessManager and RoleManager within the Access Control subsystem. Integration is mainly handled by the Navigation Caremap subsystem and the Patient Chart Form subsystem through the components InfoIntegrator and ServiceIntegrator. Additionally, the environment systems have to provide certain interfaces and services for making the integration possible.

The context and functionality of the EOC-system are also important aspects of the architectural description. The EOC-system shall run within a somatic hospital, primarily within the Health Region for Central Norway. It is an integration between several other health information systems, and it shall support read, write and edit actions.

A discussion of the choices made for the architectural description and the experiences gained during this work is given in the next part.

Part V

Discussion, Conclusion and Further Work

Chapter 22

Discussion

This chapter contains a discussion of how the goals of this thesis are met, what experiences are gained during the work with this thesis and finally what choices and decisions are made in the architectural description.

22.1 Goals

The overall goal of this thesis was to create an architectural description for CARDIAC's EOC-system. This architectural description was also supposed to show how information security could be ensured in the integration.

The overall goals were achieved, and the result of this thesis is a security focused architectural description for CARDIAC's EOC-system. The architectural description has identified important concepts which must be taken into account when integrating health information systems. In addition, relevant legal issues and important security aspects concerning the EOC-system were identified. When implementing the EOC-system within the Health Region for Central Norway, this architectural description will hopefully contribute to the implementation process for CARDIAC.

A subgoal of this thesis was to try out the architectural description framework MAFIIA/RBAC on a specific case. MAFIIA/RBAC has given some useful suggestions for access control. MAFIIA/RBAC extends the generic MAFIIA with two new models, Target Organization Security Policy Model and System Access Control Model. In this architectural description, the Target Organization Security Policy Model has not been adequate, because little was known about the target organization's security policy. The System Access Control Model became superfluous in this architectural description because we chose to document access control together with the other security mechanisms in the System Decomposition Model and System Security Model in the component viewpoint. Although the two new models in MAFIIA/RBAC turned out to be less important, other concepts introduced in MAFIIA/RBAC were maintained. Among other things role-cards, UMLsec

and security patterns were utilized. The overall impression of MAFIIA/R-BAC was that it was far too general for an integration case as complex as CARDIAC's EOC-system. MAFIIA for IIS, which is specially targeted towards system integration, was a much better support during the creation of this architectural description.

In addition to being useful for CARDIAC, this architectural description may be advantageous for other parties. SINTEF, who originally created the MAFIIA architectural framework, may have interest in the experiences gained by using MAFIIA for IIS and MAFIIA/RBAC. Neither MAFIIA for IIS nor MAFIIA/RBAC have been tried out on an integration specific case before, and the results of this thesis may therefore influence further use of MAFIIA in architectural descriptions for integrated systems.

22.2 Experiences

When integrating several systems, regardless of being health information systems or not, it is important to have certain information about each system. It should be clarified whether or not the system supports integration. If it does, information about what kind of integration architecture the system supports should be available.

It has been difficult to obtain enough information about the health information systems that were chosen for integration with CARDIAC's EOC-system. We have experienced that such documentation is often kept secret and unavailable for student projects. Because of this difficulty, we were forced to assume that the health information systems chosen for integration provided their functionality as services. Present technology development and the strategies explored make this assumption reasonable. When assuming that system functionality was provided as services, lack of information was no longer such a problem.

Service-oriented system integration requires that most, if not all, systems are changed to provide their functionality as services. Clearly, this is a drawback because changing systems is a very costly proposition. However, service-oriented system integration was chosen as integration architecture because sharing services represents a tremendous benefit. Web services are both platform and language independent; it is possible to deploy it on any platform. Web services can also be accessed by any other information system, regardless of either's platform or language.

A combination of a service-oriented and portal-oriented integration architecture was chosen because CARDIAC aims to use their IMATIS Platform for the implementation of the EOC-system. HEMIT's two strategies also recommend a service-oriented architecture where services are presented through a portal. These strategies are followed throughout this architectural description because CARDIAC's first implementation of the EOC-system is

supposed to be within the Health Region for Central Norway where these strategies prevail.

Nevertheless, it should be discussed if a service-oriented and portal-oriented integration architecture is the optimal choice. Strictly speaking, the architectural description should have had the possibility to integrate health information systems supporting other integration architectures than service-oriented. Information-oriented system integration has been the most common and widely used integration architecture, and some health information systems still support this kind of integration only.

Probably the best way to integrate health information systems would be business process oriented system integration. This integration architecture is complimentary to both information-oriented, service-oriented and even portal-oriented system integration, meaning that system services and system information are encapsulated into a single controlling business process model and presented through a portal. Business process oriented system integration therefore seems like the most optimal integration architecture. When using business process oriented integration architecture on this particular case, business processes within several wards or hospitals would be supported and the flow of information would most probably be improved. Another advantage with this integration architecture is that there is no need to change the participating systems when a change in a process flow or logic is required, only the business process model needs to be altered. Still, a great disadvantage of using this integration architecture in the health domain is the creation of the business process model. Defining the business processes within a hospital is a rather complex and time-consuming task. In addition, business process oriented system integration seems to be difficult or almost impossible to implement. No examples for use of this integration architecture in practice are found.

When it comes to standards, strategies and legal issues, we have experienced that it is quite difficult to show how these directly have affected the architectural description. Therefore, many standards and strategies are presented, but only HEMIT's two strategies are used extensively throughout the architectural description. When creating an architectural description of a health information system it is important to identify laws and regulations relevant for the system. The identified laws and regulations were used for the selection of security mechanisms which are ensured in the integration architecture. Several other security mechanisms such as database security and concurrency control could have been covered in the integration architecture, but because of the large extent of this thesis and limited amount of time only the most relevant security mechanisms were chosen.

We have experienced that security in an integration of several health information systems is a complex issue. Specially the access control mechanism within the integrated system becomes complex. The reason for this is that the number of users increases when integrating several health informa-

tion systems. In spite of the large number of users, information relevant for proper patient care should still be available for those users who have a need for it. In addition, new aspects such as single sign-on should be regarded when several systems are integrated.

In order to reduce the complexity of security in an integration of several health information systems, only one security mechanism could have been chosen. Still, an advantage of documenting several security mechanisms in the architectural description is that security is seen from a broader perspective already from the beginning of the software development process.

22.3 Choices

As mentioned above, it was difficult to obtain information about any of the health information systems relevant for integration with CARDIAC's EOC-system. Which environment systems we chose for the integration had little impact on the creation of the architectural description. We only assumed that the environment systems provided their functionality as services. Still, six environment systems were identified for integration in the Environment Systems Model; an EPR-system, PAS, RoS, Medication system, EQS and MTU. These environment systems were chosen because CARDIAC thought these were the most relevant ones.

To create a security focused integration architecture, we chose to divide the EOC-system into seven different subsystems where four of them were concerning security. The security subsystems covered the identified security concerns; digital signing, access control and auditing. Security could have been shown in one subsystem or as a component within any of the other subsystems, e.g. the Portal subsystem, but this would typically have shown security as a black box. The decomposition of the subsystems became easier when the EOC-system was divided into both functionality and security concerned subsystems.

Each of the subsystems was decomposed further into components, where the number of components reflected the complexity of the subsystem. Complex subsystems were decomposed into several components, each regarding different kind of functionality. For example, the Access Control subsystem was decomposed into an AccessManager component, a RoleManager component and a SSOManager component respectively handling the mapping of users towards resources, the mapping of access rights towards roles and the mapping of userIDs for single sign-on.

Chapter 23

Conclusion

During this master's thesis project, we have explored some issues which were used as guidance while creating the architectural description. Exploring laws and regulations has resulted in restrictions on processing and storing sensitive information and on information security in health information systems. On the basis of these restrictions, secure communication, digital signing, auditing and access control were chosen as security mechanisms covered in the architectural description.

When investigating prevailing strategies, we discovered that there are several ways to integrate different health information systems. In integrating health information systems it is important to adapt the integration architecture to the existing systems, but still be future-oriented and create an architecture which may be able to evolve with changes in the environment. This resulted in the choice of a service-oriented and portal-oriented integration architecture for the EOC-system.

The result of this thesis is a security focused integration architecture of CARDIAC's EOC-system. During the creation of the architectural description, we have identified useful requirements for the EOC-system. For emphasizing the security within the integration architecture, we chose to divide the EOC-system into several security subsystems which were further decomposed into security components. The systems chosen for integration with CARDIAC's EOC-system have less significance for the architectural description since we have assumed that each integrated system provides its functionality as services.

During the design and implementation of the EOC-system, this architectural description will be a useful contribution for CARDIAC.

Chapter 24

Further work

CARDIAC implementing the EOC-system based on the proposed architectural description, is a future goal for this thesis. But, before implementing the EOC-system, the realization viewpoint has to be emphasized and further detailed. For example, documentation of the System Integration Test Model has been omitted in the proposed architectural description because implementation of the EOC-system is outside the scope of this thesis. Thus, this model has to be described for the implementation purpose. In addition, the six health information systems proposed for integration have to be further examined for making an integration possible.

Further work on the architectural description also includes some other extensions. First of all, the architectural description should be extended to cover both read and write access to the environment systems. For example, users should be able to add new patient information such as diagnosis in PAS through the EOC-system.

Another extension is to integrate all EOC-relevant health information systems into the EOC-system. This thesis only covers an integration of six relevant health information systems.

The architectural description should also be extended to cover an integration across several hospitals within the Health Region for Central Norway. Users outside the hospital where the EOC-system runs, should be able to access the EOC-system. This means that users not connected to the hospital network should have permission to access the EOC-system over a regional health network. Then, an extension across regional boarders could be considered.

Part VI

Bibliography

Bibliography

Books

[1] Gustavo Alonso et al. *Web Services Concepts, Architectures and Applications*. Springer, 2004.

[2] Len Bass et al. *Software Architecture in Practice*. Addison-Wesley, second edition, 2003.

[3] Frank Buschmann et al. *Pattern-Oriented Software Architecture. A system of Patterns*. John Wiley & Sons, Ltd., 2000.

[4] Frank Buschmann et al. *Pattern-Oriented Software Architecture. Patterns for Concurrent and Networked Objects*. John Wiley & Sons, Ltd., 2000.

[5] D.R. Clark and D.R. Wilson. *A comparison of commercial and military computer security policies*. Proceedings of the 1987 IEEE Symposium on Security and Privacy, 1987.

[6] Ramaswamy Chandramouli David F. Ferraiolo, D. Richard Kuhn. *Role-Based Access Control*. Artech House, Inc., 2003.

[7] Torgeir Daler et al. *Håndbok i datasikkerhet - informasjonsteknologi og risikostyring*. Tapir akademisk forlag, 2002.

[8] Martin Fowler and Kendall Scott. *UML Distilled*. Addison-Wesley, second edition, 2000.

[9] Erich Gamma et al. *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison Wesley, 1995.

[10] Dieter Gollmann. *Computer Security*. John Wiley and Sons, Inc., 1999.

[11] Statens helsetilsyn. *Pasientjournalen. Innhold, gruppering og arkivering av pasientdokumentasjon i somatiske sykehus.* Statens helsetilsyn utredningsserie 3-94, 1994.

[12] Jan Jürjens. *Secure Systems Development with UML (unpublished).* Springer-Verlag, 2004. Downloaded 10/3-05 from `http://www4.in.tum.de/lehre/seminare/hs/WS0405/uml/umlsec.pdf`.

[13] David S. Linthicum. *Next Generation Application Integration.* Addison-Wesley, 2004.

[14] William Stalling. *Network Security Essentials.* Prentice Hall, second edition, 2003.

[15] R. C. Summers. *Secure Computing: Threats and Safeguards.* McGraw-Hill, 1997.

[16] SINTEF Telecom and Informatics. *The MAFIIA Handbook - An Architectural Description Framework for Information Integration Systems.* SINTEF Telecom and Informatics, 2003.

[17] Erlend Stav Ulrik Johansen and Ståle Walderhaug. *MAFIIA/H Overview and guide to use the MAFIIA "healthcare" framework.* SINTEF Telecom and Informatics, 2003.

[18] John Viega and Gary McGraw. *Building Secure Software - How to Avoid Security Problems the Right Way.* Addison-Wesley Professional Computing Series, 2001.

Articles

[19] Allen Brown et al. *Web Services Architecture Requirements.* W3C, 2002. Downloaded 13/3-05 from `http://www.w3.org/TR/wsa-reqs/`.

[20] Joe Clabby. *Book Excerpt: What Are Web Services?* Computerworld, 2004. Downloaded 14/4-05 from `http://web.uccs.edu/dowens/IS440/week6/Web_Services_book_excerpt.htm`.

[21] Inger Dybdahl Sørby et al. *Characterising Cooperation In The Ward: A Framework for Producing Requirements to Mobile Electronic Healthcare Records*, Downloaded 6/3-2005 from `http://www.idi.ntnu.no/~ingerdyb/Hof-paper-final-ids.pdf`.

[22] E. B. Fernandez and Rouyi Pan. *A pattern language for security models. PLoP 2001*, 2001. Downloaded 8/4-2005 from `http://hillside.net/`

plop/plop2001/accepted_submissions/PLoP2001/ebfernandezandrpan0/PLoP2001_ebfernandezandrpan0_1.pdf.

[23] David F. Ferraiolo and D. Richard Kuhn. *Role-Based Access Control. National Institute of Standards and Technology*, 1992. Downloaded 10/3-2005 from http://csrc.nist.gov/rbac/ferraiolo-kuhn-92.pdf.

[24] COTS journal. *UML 2.0 Automates Code Generation from Architecture*, Downloaded 10/6-2005 from http://www.cotsjournalonline.com/home/printthis.php?id=100113.

[25] Razvan Peteanu. *Design Patterns in Security. SecurityPortal.com*, 2001. Downloaded 1/10 2004 from http://members.rogers.com/razvan.peteanu/designpatterns20010611.html.

[26] Jim Reynolds and R. Chandramouli. *Role-Based Access Control (RBAC) Protection Profile. Common Criteria Protection Profile*, 1998. Downloaded 20/3-05 from http://www.commoncriteriaportal.org/public/files/ppfiles/RBAC_987.pdf.

[27] Konstantin Rozinov. *Secure Programming. SFS Meeting - Secure programming*, 2003. Downloaded 1/4-05 from http://konstantin.sfs.poly.edu/presentations/security_patterns.pdf.

[28] Guttorm Sindre and Andreas L. Opdahl. *Templates for Misuse Case Description.* Downloaded 15/6-2005 from http://www.ifi.uib.no/conf/refsq2001/papers/p25.pdf.

[29] Ronald Wassermann and Betty H.C. Cheng. *Security Patterns. Sens meeting, Michigan State University*, 2003. Downloaded 14/2-2005 from http://www.cse.msu.edu/sens/sensTalks/Sp2003/docs/2003-04-11-11-00.slides.ppt.

[30] Joseph Yoder and Jeffrey Barcalow. *Architectural Patterns for Enabling Application Security. Washington University (wucs-97-34)*, 1997. Downloaded 1/5-2004 from http://www.joeyoder.com/papers/patterns/Security/appsec.pdf.

Reports

[31] A. Shapiro D. Braun, J. Sivils and J. Versteegh. *UML tutorial.* Kennesaw State University, 2001. Downloaded 10/6-05 from http://pigseye.kennesaw.edu/~dbraun/csis4650/A\&D/UML_tutorial/index.htm.

[32] Mirela Divic and Andreas Grytting Furuseth. *MAFIIA/RBAC: Describing access control as part of a system architectural framework.* Norwegian University of Science and Technology, Department of Computer and Information Science, 2004.

[33] Harald Strøm et al. *IT på Japanske sykehus - en kilde til inspirasjon og læring for norsk helsevesen? KITH Rapport 11/03*, 2003.

[34] Directorate for Health and Social Affairs. *Te@mwork 2007 Electronic Cooperation in the Health and Social Sector.* Ministry of Health and Ministry of Social Affairs, 2004. Downloaded 10/3-05 from `http://www.shdir.no/assets/13163/Te@mwork%202007.pdf`.

[35] Helsedepartementet. *Styringsdokumentene for de regionale helseforetakene 2004.* Helsedepartementet, 2003.
Downloaded 10/3-05 from `http://www.dep.no/shd/sykehusreformen/brev/042031-990041/index-dok000-b-n-a.html`.

Laws

[36] Ministry of Health and Care Services. *The Health Personnel Act.* Lovdata, 2005. Downloaded 16/2-2005 from `http://odin.dep.no/hod/engelsk/regelverk/p20042245/042051-200005/index-dok000-b-n-a.html`.

[37] Ministry of Health and Care Services. *Personal Health Data Filing System Act.* Lovdata, 2005. Downloaded 25/2-2005 from `http://www.ub.uio.no/ujur/ulovdata/lov-20010518-024-eng.pdf`.

[38] Ministry of Health and Care Services. *Patients' Rights Act.* Lovdata, 2005. Downloaded 7/3-2005 from `http://www.ub.uio.no/ujur/ulovdata/lov-19990702-063-eng.pdf`.

[39] Ministry of Justice and the Police. *Personal Data Regulations.* The Data Inspectorate, 2005. Downloaded 25/2-2005 from `http://www.ub.uio.no/ujur/ulovdata/for-20001215-1265-eng.pdf`.

[40] Ministry of Justice and the Police. *Personal Data Act.* Lovdata, 2005. Downloaded 28/2-2005 from `http://www.ub.uio.no/ujur/ulovdata/lov-20000414-031-eng.pdf`.

[41] Sosial og helsedepartementet. *Lovforslaget til sykehusreformen, Ot.prp. nr. 66 (2000-2001) Om lov om helseforetak m.m. (helseforetaksloven)*, Downloaded 28/1-05 from `http://odin.dep.no/hod/norsk/publ/otprp/030001-050012/dok-bn.html`.

[42] Kultur– og kirkedepartementet. *Lov om arkiv. Lovdata*, 2005. Downloaded 7/3-2005 from `http://www.lovdata.no/all/nl-19921204-126.html`.

[43] Helse– og omsorgsdepartementet. *Forskrift for pasientjournal.* Lovdata, 2005. Downloaded 28/2-2005 from `http://www.lovdata.no/cgi-wift/wiftldles?doc=/usr/www/lovdata/for/sf/ho/ho-20001221-1385.html\&dep=alle\&titt=pasientjournal\&`.

Standards and Strategies

[44] *Health Level 7*, Downloaded 11/5-05 from `http://www.hl7.org`.

[45] Health Informatics Committee IT-014. *Health Informatics-Requirements for an electronic health record architecture* (iso/ts 18308:2004, mod). *Sai Global - Distributor of Australian Standards*, 19/1-2005.

[46] Anthony Nadalin et al. *Web Services Security: SOAP Message Security 1.0 (WS-Security 2004) OASIS Standard 200401.* OASIS Open 2002-2004, 2004. Downloaded 18/4-05 from: `http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf`.

[47] HEMIT. *IT arkitekturstrategi. Helse Midt-Norge*, 28/7-2004.

[48] CEN/TC 251/WGII Secretariat: SIS Swedish Standards Institute. *Health informatics - System of concepts to support Continuity of care - Part 1: Basic concepts. Terminology and Knowledge bases*, 2004.

[49] KITH. *Elektronisk pasientjournal standard; Arkitektur, arkivering og tilgangsstyring Del 1. KITH*, 2001.

[50] Sosial– og helsedirektoratet. *Norm for informasjonssikkerhet i helsesektoren - høringsutkast*, Downloaded 3/6-2005 from `http://www.shdir.no/vp/multimedia/archive/00001/Norm_for_informasjons_1280a.doc`.

[51] Joar Øyen et al. *Integrasjonsstrategi. Helse Midt-Norge og St.Olavs Hospital*, August 2003.

Web pages

[52] *The GALEN Project*, Downloaded 11/5-05 from http://www.cs.man.ac.uk/mig/projects/old/galen/.

[53] *Unified Medical Language System*, Downloaded 11/5-05 from http://www.nlm.nih.gov/research/umls/.

[54] *SNOMED International*, Downloaded 11/5-05 from http://www.snomed.org.

[55] *The WHO Family of International Classifications*, Downloaded 11/5-05 from http://www.who.int/classifications/en/.

[56] *Data Inspectorate*, Downloaded 15/4-05 from http://www.datatilsynet.no/.

[57] *The Norwegian Board of Health*, Downloaded 15/4-05 from http://www.helsetilsynet.no/.

[58] *HEMIT*, Downloaded 15/4-05 from http://www.hemit.no/ordoguttrykk/.

[59] *Directorate of Health and Social Affairs*, Downloaded 15/4-05 from http://www.shdir.no/.

[60] *AMA - American Medical Association*, Downloaded 25/2-05 from http://www.ama-assn.org/ama/pub/category/4610.html.

[61] Alistair Cockburn. *Using CRC cards*. Downloaded 15/9-2004 from http://alistair.cockburn.us/crystal/articles/ucrcc/usingcrccards.html.

[62] Jupitermedia Corporation. *Webopedia: Online Dictionary for Computer and Internet Terms*. Downloaded 27/4-05 from http://www.webopedia.com.

[63] Microsoft Corporation. *Technology Overview*. Microsoft .NET Framework Developer Center, Downloaded 17/4-05 from http://msdn.microsoft.com/netframework/technologyinfo/Overview/default.aspx.

[64] D.Eastlake and J. Reagle Jr. *XML Signature WG*. W3C, 2003. Downloaded 28/4-05 from http://www.w3.org/Signature/.

[65] Senter for informasjonssikring (SIS). *PKI - Public Key Infrastructure*, 2005. Downloaded 15/3-05 from http://www.norsis.no/details.php?type=veiledninger\&id=292.

[66] Ringholm GmbH. *Useful Links*, 2005. Downloaded 3/5-05 from `http://www.ringholm.de/en/links.htm`.

[67] Helsedepartementet. *Organisering av den sentrale helseforvaltningen*, Downloaded 20/3-05 from `http://odin.dep.no/hd/norsk/sykehus/organisering/042031-990086/dok-bn.html`.

[68] Joseph Reagle Jr. *XML Encryption WG*. W3C, 2003. Downloaded 28/4-05 from `http://www.w3.org/Encryption/2001/`.

[69] *Ministry of Health and Care Services*, 2005. Downloaded 15/4-05 from `http://www.odin.no/hod/norsk/bn.html`.

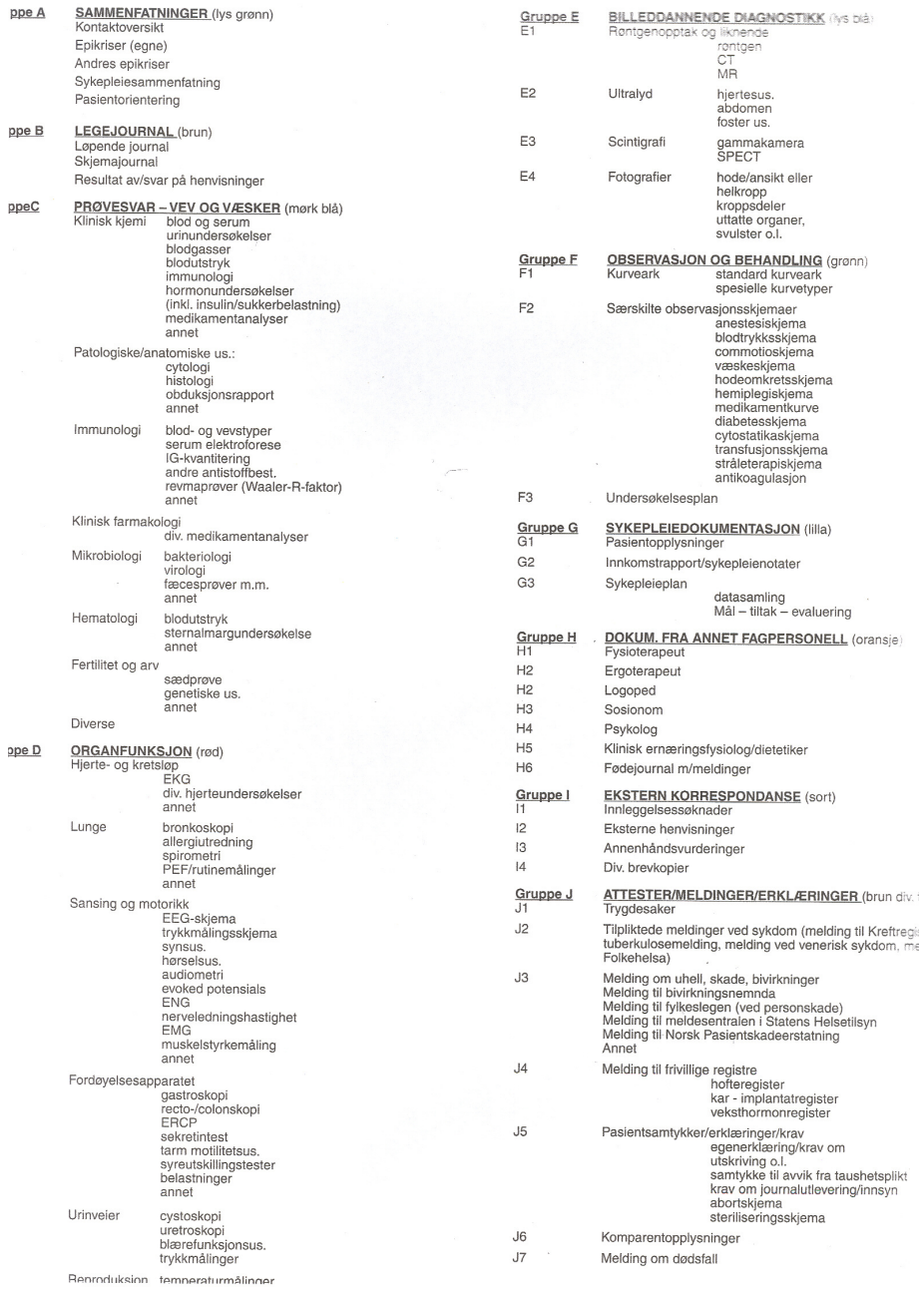
[70] California Academy of Physician Assistants. *The Supervising Physician*, Nov 2004. Downloaded from `http://www.capanet.org/pasuper_phy.cfm`.

Part VII

Appendix

Appendix A

Paper-based patient record



ppe A **SAMMENFATNINGER** (lys grønn)
Kontaktoversikt
Epikriser (egne)
Andres epikriser
Sykepleiesammenfatning
Pasientorientering

ppe B **LEGEJOURNAL** (brun)
Løpende journal
Skjemajournal
Resultat av/svar på henvisninger

ppeC **PRØVESVAR – VEV OG VÆSKER** (mørk blå)
Klinisk kjemi: blod og serum, urinundersøkelser, blodgasser, blodutstryk, immunologi, hormonundersøkelser (inkl. insulin/sukkerbelastning), medikamentanalyser, annet
Patologiske/anatomiske us.: cytologi, histologi, obduksjonsrapport, annet
Immunologi: blod- og vevstyper, serum elektroforese, IG-kvantitering, andre antistoffbest., revmaprøver (Waaler-R-faktor), annet
Klinisk farmakologi: div. medikamentanalyser
Mikrobiologi: bakteriologi, virologi, fæcesprøver m.m., annet
Hematologi: blodutstryk, sternalmargundersøkelse, annet
Fertilitet og arv: sædprøve, genetiske us., annet
Diverse

ppe D **ORGANFUNKSJON** (rød)
Hjerte- og kretsløp: EKG, div. hjerteundersøkelser, annet
Lunge: bronkoskopi, allergiutredning, spirometri, PEF/rutinemålinger, annet
Sansing og motorikk: EEG-skjema, trykkmålingsskjema, synsus., hørselsus., audiometri, evoked potensials, ENG, nerveledningshastighet, EMG, muskelstyrkemåling, annet
Fordøyelsesapparatet: gastroskopi, recto-/colonskopi, ERCP, sekretintest, tarm motilitetsus., syreutskillingstester, belastninger, annet
Urinveier: cystoskopi, uretroskopi, blærefunksjonsus., trykkmålinger
Reproduksjon: temperaturmålinger

Gruppe E **BILLEDDANNENDE DIAGNOSTIKK** (lys blå)
E1 Røntgenopptak og liknende: røntgen, CT, MR
E2 Ultralyd: hjertesus., abdomen, foster us.
E3 Scintigrafi: gammakamera, SPECT
E4 Fotografier: hode/ansikt eller helkropp, kroppsdeler, uttatte organer, svulster o.l.

Gruppe F **OBSERVASJON OG BEHANDLING** (grønn)
F1 Kurveark: standard kurveark, spesielle kurvetyper
F2 Særskilte observasjonsskjemaer: anestesiskjema, blodtrykksskjema, commotioskjema, væskeskjema, hodeomkretsskjema, hemiplegiskjema, medikamentkurve, diabetesskjema, cytostatikaskjema, transfusjonsskjema, stråleterapiskjema, antikoagulasjon
F3 Undersøkelsesplan

Gruppe G **SYKEPLEIEDOKUMENTASJON** (lilla)
G1 Pasientopplysninger
G2 Innkomstrapport/sykepleienotater
G3 Sykepleieplan: datasamling, Mål – tiltak – evaluering

Gruppe H **DOKUM. FRA ANNET FAGPERSONELL** (oransje)
H1 Fysioterapeut
H2 Ergoterapeut
H2 Logoped
H3 Sosionom
H4 Psykolog
H5 Klinisk ernæringsfysiolog/dietetiker
H6 Fødejournal m/meldinger

Gruppe I **EKSTERN KORRESPONDANSE** (sort)
I1 Innleggelsessøknader
I2 Eksterne henvisninger
I3 Annenhåndsvurderinger
I4 Div. brevkopier

Gruppe J **ATTESTER/MELDINGER/ERKLÆRINGER** (brun div.
J1 Trygdesaker
J2 Tilpliktede meldinger ved sykdom (melding til Kreftregi tuberkulosemelding, melding ved venerisk sykdom, me Folkehelsa)
J3 Melding om uhell, skade, bivirkninger
Melding til bivirkningsnemnda
Melding til fylkeslegen (ved personskade)
Melding til meldesentralen i Statens Helsetilsyn
Melding til Norsk Pasientskadeerstatning
Annet
J4 Melding til frivillige registre: hofteregister, kar - implantatregister, veksthormonregister
J5 Pasientsamtykker/erklæringer/krav: egenerklæring/krav om utskriving o.l., samtykke til avvik fra taushetsplikt, krav om journalutlevering/innsyn, abortskjema, steriliseringsskjema
J6 Komparentopplysninger
J7 Melding om dødsfall

Figure A.1: Contents of a paper-based patient record.

Appendix B

Patterns

This appendix gives a representation of the patterns recommended for use when creating the architectural description for CARDIAC's EOC-system.

B.1 Adapter

The description of the Adapter pattern is mainly from the overview over relevant patterns in MAFIIA for IIS [16].

B.1.1 Intent

The intention of Adapter pattern is to convert the interface of a class into another interface clients expects. Adapter lets classes work together that could not otherwise because of incompatible interfaces [9].

B.1.2 Also known as

- Wrapper

B.1.3 Applicability

The adapter pattern is applicable when there is a need to use one or more existing resources, but the interface(s) of their classes do not match the needed interface. To solve the problem one or more adapter classes are created that implement the required interface, and map this interface to the classes of the existing resources.

B.1.4 Structure

Figure B.1 shows the structure of the Adapter pattern.

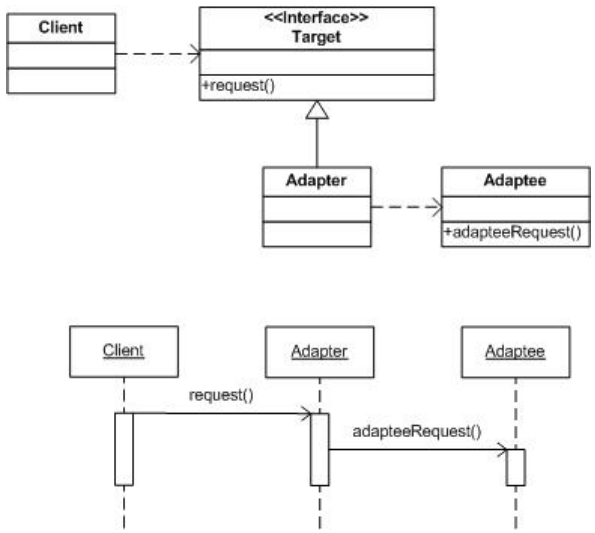


Figure B.1: Structure of the Adapter pattern [16].

B.1.5 Consequences

The implementation of an adapter is often straightforward, e.g. when the target interface and the adaptee interface only differs in operation names and ordering of parameters. The complexity is increased if the interfaces differ more fundamentally, or if the adapter is made more generic to allow configuration to work with different adaptee classes.

B.2 Façade

The description of the Façade pattern is mainly based on the overview of relevant patterns in MAFIIA for IIS [16].

B.2.1 Intent

The intention of Façade pattern is to provide a unified interface to a set of interfaces in a subsystem. Façade defines a higher-level interface that makes the subsystem easier to use [9].

B.2.2 Applicability

The Façade pattern is applicable when a subsystem originally consists of many interfaces, and are complicated to use. When the pattern is applied, a façade interface is constructed. The clients access the subsystem only through the façade, and the façade uses the original interfaces of the subsystem in its implementation.

B.2.3 Structure

Figure B.2 shows the structure of the Façade pattern.

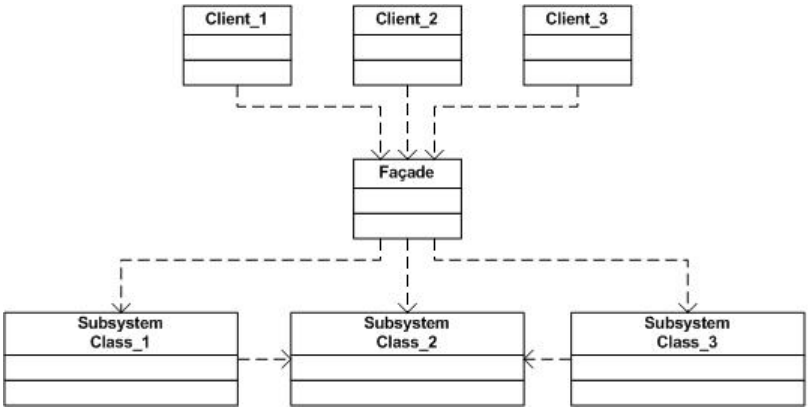


Figure B.2: Structure of the Façade pattern [16].

B.2.4 Consequences

When the façade pattern is used, direct dependencies on the internal, complex parts of the subsystem from the clients are avoided. In addition to making the subsystem easier to use for the client, it also makes it easier to do internal changes in the subsystem without affecting the clients.

B.3 Single access point

B.3.1 Intent

The intention of Single access point pattern is to create only one way to get into the system. This is done by creating a login screen which collects information about the user and verifies it in a check point. The result of this action is a session which keeps track of the user's privileges and his interactions with the system [30].

B.3.2 Also known as

- Login Window
- One Way In
- Guard Door
- Validation Screen

B.3.3 Motivation

Most computer systems today are connected to other systems. This means that one system may be accessible for other systems through several different entry points. Just like it is difficult for one person to guard 4 different doors in a castle, it is very difficult for a security system to guard many entry points [30].

Single access point handles this problem by providing only one way to get into the system and creates a secure place for user validation.

B.3.4 Applicability

This pattern is applicable to most applications, or complete systems, which need to communicate with external entities [29]. It is used in login screens by secure Web servers. The Internet connection of most organizations reflects this pattern [25].

B.3.5 Structure

Figure B.3 shows the structure of the Single access point pattern.

B.3.6 Consequences

The Single access point pattern works a bit like a double-edged sword. A single access point is easy to control, but it is also easy to attack [27]. This pattern may also result in a bottleneck since it does not allow multiple entry points to simplify the access to an application [30].

The advantage of this pattern is that it guarantees that all values are initialized in a correct way. It also simplifies control flow since everything goes through a single point of responsibility [30].

B.3.7 Related Patterns

The Single access point pattern is related to the Check point pattern which is used to initialize user's role and session [30].

B.4 Check point

B.4.1 Intent

The intention of the Check point pattern is to give a way to handle and organize security checks [27], incoming requests and to handle violations [29]. This pattern verifies the information that comes in through single access point.

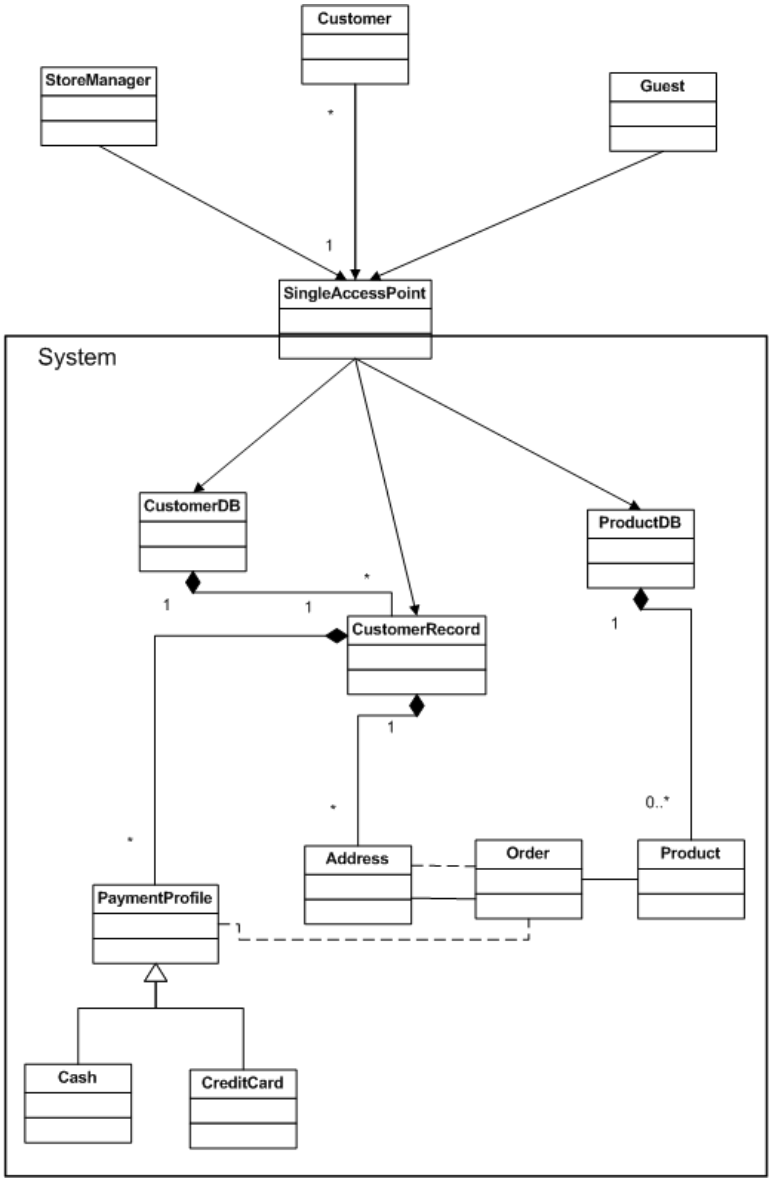


Figure B.3: Structure of Single access point pattern. Adapted from the paper *Security Patterns* [29].

B.4.2 Also known as

- Access Verification
- Authentication and Authorization
- Holding off hackers
- Validation and Penalization
- Make the punishment fit the crime

B.4.3 Motivation

Single access point pattern is useful for keeping unwanted visitors from gaining access to parts of the system which they are not authorized for. But the checks that are done in single access point can sometimes cause more trouble then necessary for regular non-malicious users. Some users make mistakes which the system may interpret as an intent to attack the system (e.g. mistyped password). The system has to allow users to sometimes make these mistakes. Check point provides a structure for all the checks and how to handle requests and violations [30].

B.4.4 Applicability

This pattern may be applied to any system that needs to monitor communication. A security policy is needed for performing the checks [29].

B.4.5 Structure

Figure B.4 show the structure of the Check point pattern.

B.4.6 Consequences

Not all security checks can be done at startup, so check point needs a secondary interface for those parts of the application that need this kind of checks. Although check point might be a complex algorithm, it is isolated in one location so it makes the security algorithm easier to change [30]

B.4.7 Related Patterns

The following patterns are related to the Check point pattern [29]:

- Single access point
- Role-based access control

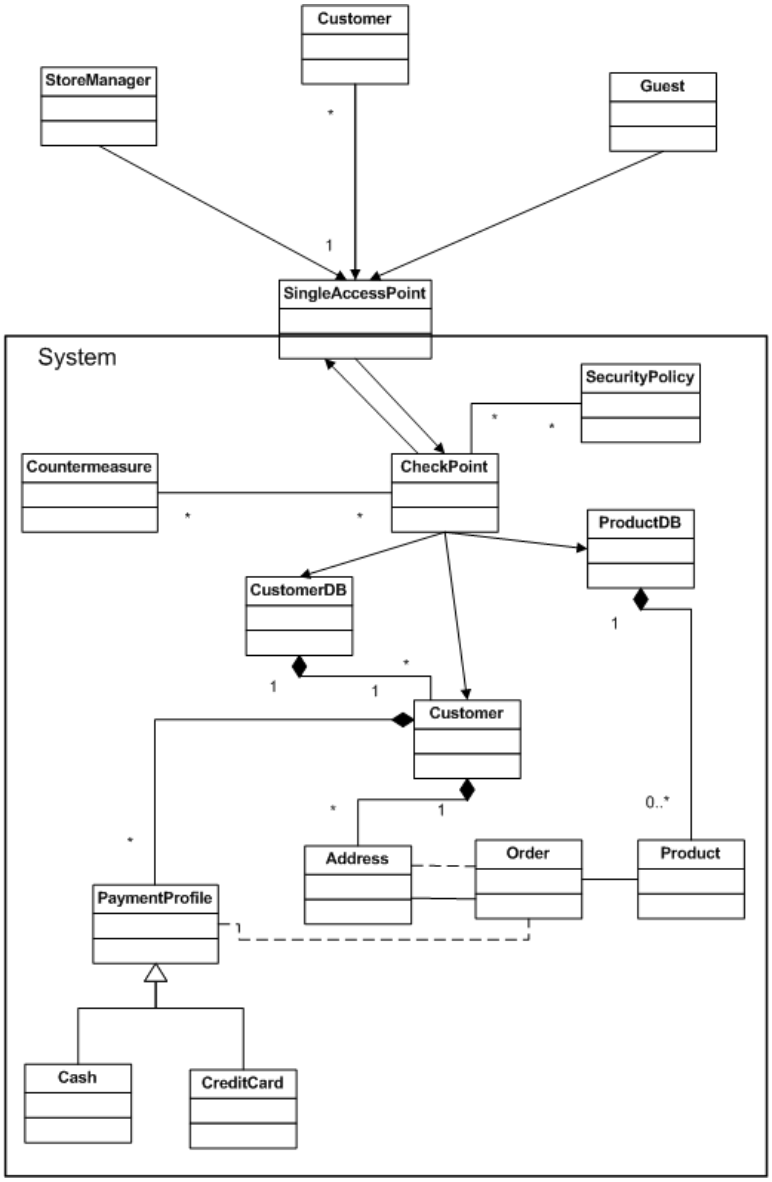


Figure B.4: Structure of Check point pattern. Adapted from the paper *Security Patterns* [29].

B.5 Role-Based Access Control

B.5.1 Intent

The intention of this pattern is to assign rights to users according to their roles in the organization. For security reasons, the users should only get as much information as they need to perform their tasks, neither more nor less. This security policy is also known as "need-to-know principle", explained in the book *Secure Computing: Threats and Safeguards* [15].

B.5.2 Also known as

- Roles
- Actors
- Groups
- Projects
- Profiles
- Jobs
- User types

B.5.3 Motivation

Most organizations have different job functions which require different skills and responsibilities. Users should get rights according to their responsibilities and the job functions they are supposed to perform. Job functions may be implemented as roles that people play while performing their tasks [22].

Granting rights to individual users is too time consuming and requires storage of many authorization rules. The solution lies in grouping the users in to groups which are assigned roles. One user may have more than one role, and role hierarchies, with inheritance of rights, are also allowed.

B.5.4 Applicability

This pattern is applicable to any application which has different users who need different access to resources. Web-based systems often have a huge variety of users (e.g. company employees, customers, partners, search engines) [22]. Health information systems also have several kinds of users and require a strict security policy for giving the users access to information.

B.5.5 Structure

Figure B.5 shows the structure of the Role-based access control pattern.

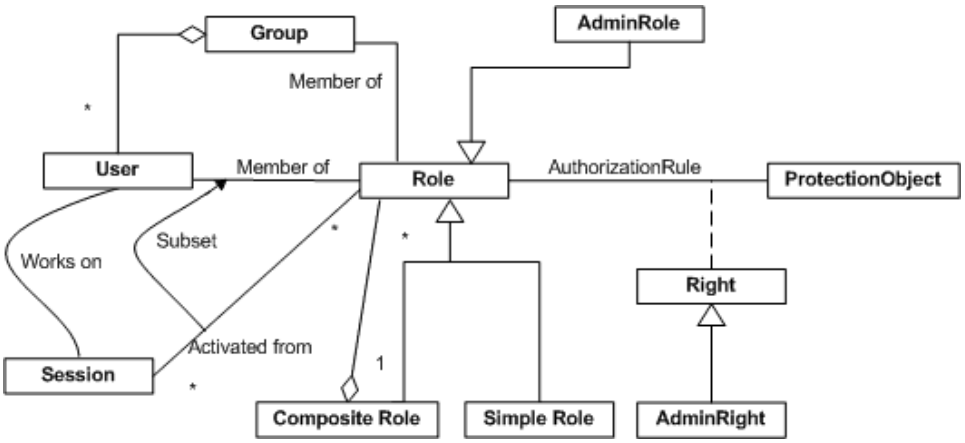


Figure B.5: A pattern for role-based access control. Adapted from the paper *A pattern language for security models* [22].

B.5.6 Participants

The following elements are in [29] identified as participants in the RBAC-pattern:

- Protection Object
- Right
- Role
- Roles
- User

B.5.7 Collaborations

Roles are associated to a set of objects. These objects define properties of each relationship. Each user is associated with a role which determines his privileges. Information about access privileges can be queried by other system components [29].

B.5.8 Consequences

One of the greatest advantages of Role-based access control pattern is that it reduces complexity of security. Roles may directly reflect the organization in a company. Users may have several roles at a time, but some roles can also be mutually exclusive, which means that one user cannot be assigned to two different roles in the same set of roles [22].

One disadvantage that might be seen with this pattern is that it adds additional complexity to the system.

B.5.9 Related Patterns

The Check point pattern is relevant to this pattern.

Appendix C

UML

The Unified Modeling Language (UML) is a standard for specifying object-oriented or component-oriented software systems. It is a modeling language that may be used to specify architectural and behavioral aspects of software.

UML diagrams describe various views on different parts of a system design. There are several kinds of UML diagrams, each describing different aspects of a system at a different level of abstraction. The following sections describe some of the most common UML diagram types.

C.1 Use case diagrams

Use case diagrams are diagrams which describe a set of scenarios which are tied together by a common user goal [12]. A scenario is a sequence of steps describing the interaction between a user and the system. An example of a use case diagram is shown in Figure C.1.

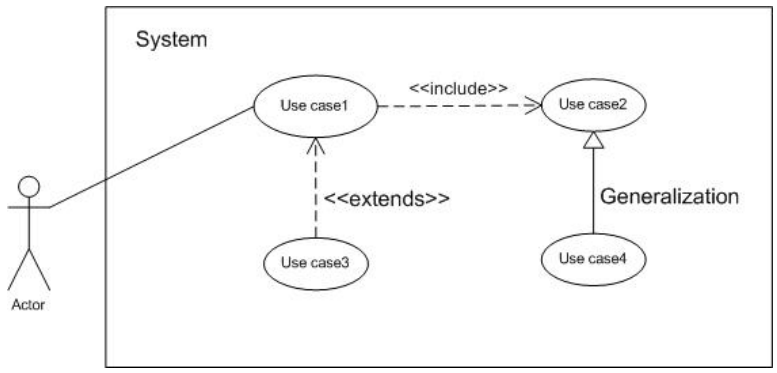


Figure C.1: Example of a use case diagram following the UML convention.

An ***actor*** is a role that a user plays with respect to the system. The actor does not necessarily have to be a human being - it may as well be a system or a system component. The $\langle\langle include \rangle\rangle$ relationship is used when

several use cases have similar behavior. The ⟨⟨*generalization*⟩⟩ relationship is used when one use case has similar behavior as another use case, but is still does a bit more. The ⟨⟨*extends*⟩⟩ relationship is essentially similar to generalization, but it adds more rules to it [8].

C.2 Misuse case diagrams

A misuse case is the inverse of a use case. A misuse case often shows functions that the system should not allow. In other words, a misuse case may be defined to be a completed sequence of actions which result in loss for the organization or some specific stakeholder [28]. Misuse case diagrams introduce a special type of actor, namely the bad guy or the ***crook***, who performs the malicious operations towards the system. In addition to the ⟨⟨*extend*⟩⟩ and ⟨⟨*include*⟩⟩, misuse cases also define ⟨⟨*prevents*⟩⟩ and ⟨⟨*detects*⟩⟩ relations as shown in Figure C.2.

The ⟨⟨*prevents*⟩⟩ relation shows which use cases prevent certain threats to the system. Equivalently, the ⟨⟨*detects*⟩⟩ relation is used to emphasize the use cases which detect certain threats.

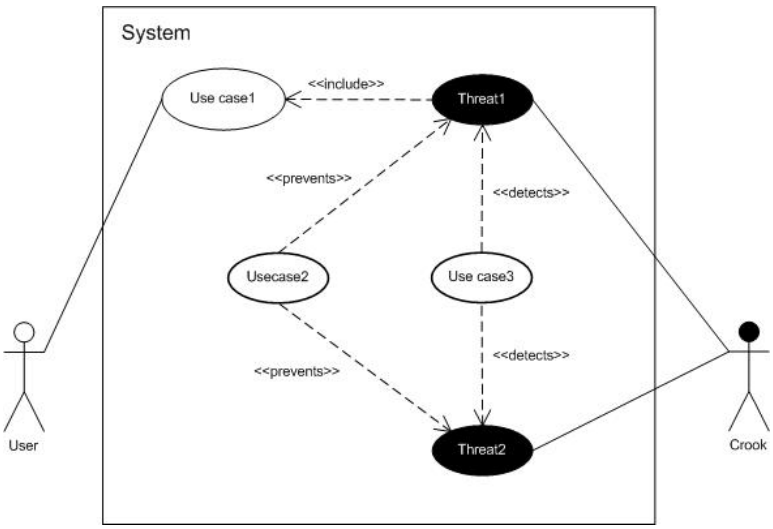


Figure C.2: Example of a misuse case diagram.

C.3 Class diagrams

Class diagrams define the class structure of a system. They show attributes, operations, signals and relationships between classes [12]. An example of a class diagram is shown in Figure C.3.

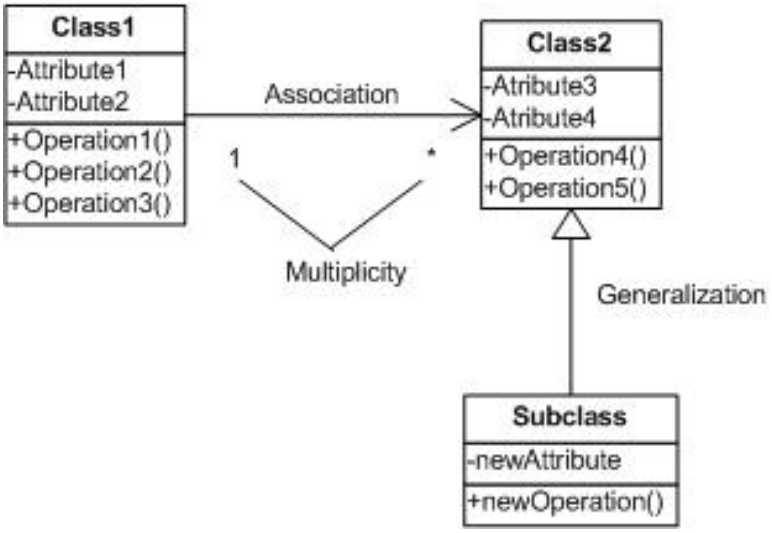


Figure C.3: Example of a class diagram following the UML convention [31].

Attributes are usually characteristics of a class object. ***Operations*** are the processes that a class can carry out. Operations usually have a visibility type, a return type and a name. ***Associations*** represent relationships between the instances of classes. Each association end has ***multiplicity***, which indicates how many objects participate in the given relationship. The arrows on the association lines indicate ***navigability***. ***Generalization*** is used when several classes have similarities. One can then place all the similarities in one superclass and leave the other functionality in the subclasses.

C.4 Sequence diagrams

Sequence diagrams describe interactions between objects or system components through message exchange [8]. Figure C.4 shows an example of a UML sequence diagram. The vertical line below each object is the object's ***activation line***.

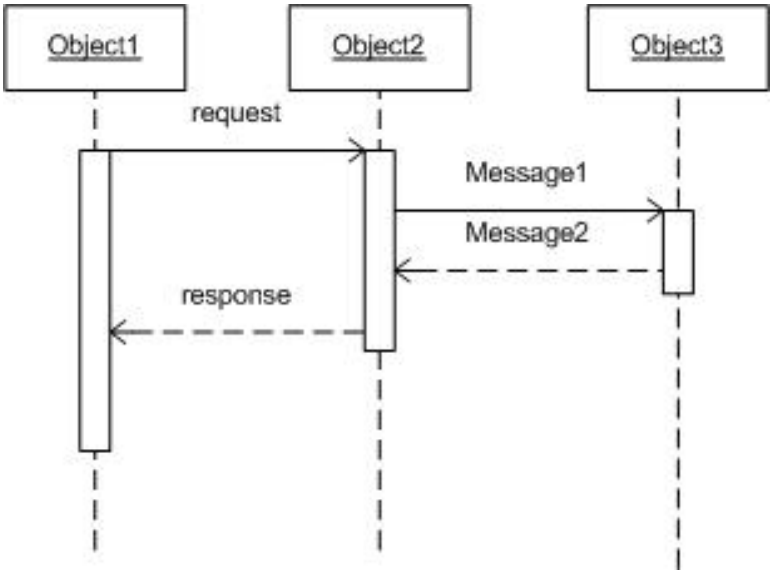


Figure C.4: Example of a sequence diagram following the UML convention [31].

C.5 Activity diagrams

Activity diagrams show the flow of control between several components or actors in the system [31]. Figure C.5 shows an example of a UML activity diagram.

An ***action state*** is a state of doing something, i.e. a software routine. A ***fork*** has one incoming transitions and several outgoing transitions which are all activated at the same time. A ***join*** has several incoming transitions and only one outgoing transition. The actions between a fork and a join are parallel actions.

A ***branch*** has a single incoming transition and several outgoing transition. But, in this case, only one of the outgoing transitions can be chosen, i.e. the outgoing transitions are mutually exclusive. A ***merge*** marks the end of conditional behavior which is started by a branch.

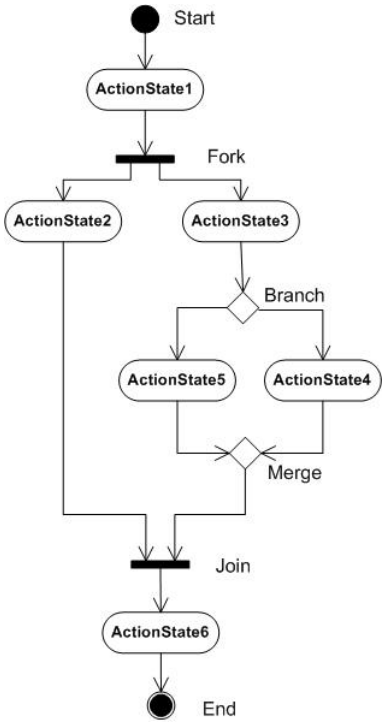


Figure C.5: Example of an activity diagram following the UML convention [31].

C.6 Package diagrams

A package diagram is a UML diagram composed only of packages and the dependencies between them. A package is a UML construct that makes is possible to organize model elements, such as use cases or classes, into groups [31]. The groups within one package often have responsibilities that are strongly related. Packages are depicted as file folders (see Figure C.6) and can be applied on any UML diagram.

A ***dependency*** is in package diagrams visualized as a dashed line between the packages.

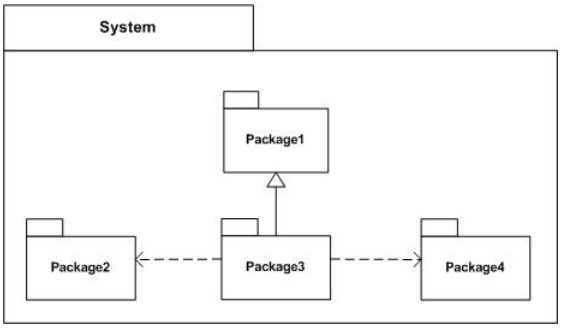


Figure C.6: Example of a package diagram following the UML convention [31].

C.7 Component diagrams

Component diagrams show the software components of a system and how they are related to each other. These relationships are called dependencies. The component diagram contains components and dependencies. Components represent the physical packaging of a module of code. The dependencies between the components show how changes made to one component may affect the other components in the system. Dependencies in a component diagram are represented by a dashed line between two or more components. An example of a dependency between components is shown in Figure C.7. Component diagrams can also show the interfaces used by the components to communicate to each other.

C.8 Deployment diagrams

Deployment diagrams describe the physical relationships among the software and hardware components in a system. Each node on a deployment diagram represents some kind of computational unit, i.e. a piece of hardware. Figure C.7 shows an example of a UML deployment diagram.

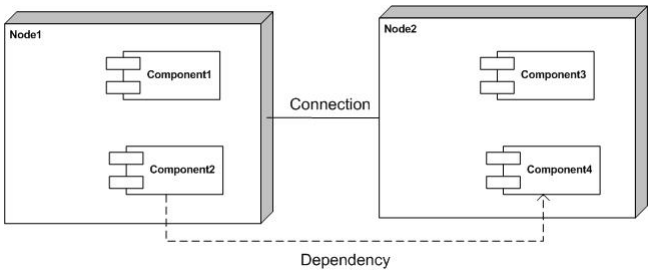


Figure C.7: Example of a deployment diagram following the UML convention [31].

Each ***node*** on a deployment diagram represents a computational unit or a piece of hardware. ***Connections*** among nodes show the communication paths over which the system will interact [8].

C.9 Composite structures

One of the disadvantages with earlier versions of UML has been the gap between the classes created and the system which is being built. It has been possible to graphically define the building blocks, but in order to specify the way these blocks go together, one had to drop down to obscure code [24].

In order to avoid this problem, the Object Management Group (OMG) has delivered UML 2.0, a new version of the standard that addresses the above mentioned gap by introducing composite structures and more specifically "Structured Classes".

C.9.1 Ports and interface

In structured classes, each class has a set of ports, and all interactions (calls or asynchronous messages) go through them [24]. An object still has one-way encapsulation, because it does not know who called it (although it can know which port a message came from). It also has two-way encapsulation, because it calls or sends messages through a port. It knows the port, but not what is behind it. Ports are optional; simple data classes often do not need them, and calls are made straight to the object.

A ***port*** can have two types of ***interfaces*** - provided and required [24]. A ***provided*** interface is a set of operations the port makes available to the outside world. The provided interface is modeled by a lollipop shape as shown on the left side of Figure C.8.

A ***required*** interface is a set of operations the port may use on the outside world. An example of a required interface is shown on the right side of Figure C.8.

Figure C.8: Example of composite structures in UML 2.0.

A port can have zero or more of each type of interface. Most often, a port will have a single provided interface and a single required interface that are the two sides of a protocol. Port compatibility is statically verified: each port at one end of a connector must *provide* every operation that is *required* by a port at the other end of the connector. This allows "plug-and-play" composition of parts [24].

C.10 UML extensions

UML defines three extension mechanisms allowing modelers to tailor UML to specific application domains without having to modify the underlying modeling language. The three mechanisms are stereotypes, tagged values and constraints.

A ***stereotype*** is an extension to the vocabulary of the UML, allowing the modeler to add new building blocks derived from existing ones. Stereotypes are normally shown as text strings surrounded by brackets. Stereotypes can be used to create collections of constructs which are specialized toward given needs. Such collections of constructs are called profiles.

A ***tagged value*** is used to store information about an individual model element. Tags can be defined for existing elements of UML or for individual stereotypes. Its value applies to the element itself and not its instances, and does thus not correspond to a class attribute. A tagged value is represented as a string enclosed by braces and placed below the name of the element it belongs to. The string includes a name of some property the modeler wants to record (the tag), a separator (an equal sign), and the value of that property for the given element.

A ***constraint*** is an extension of the semantics of a UML element, providing the possibility to add new rules or modifying existing ones. Constraints specify conditions that must be true for the model to be well-formed.

A profile puts stereotypes, tagged values and constraints all together.

C.10.1 UMLsec

UMLsec is a profile, allowing security-related information and concepts (e.g. smart cards, encryption, secrecy, access control, integrity, etc.) in UML models. The profile contains stereotypes with tags and constraints to communicate security requirements and security attainment.

UMLsec defines several secure stereotypes - this appendix only presents two of them in Table C.1. Stereotypes are in UML indicated with $\langle\langle stereotype \rangle\rangle$ and tags with $\{tag\}$.

Stereotype	**Tags**	**Constraints**	**Description**
rbac	protected, role, right	Only permitted activities executed	Enforces role-based access control
guarded	guard		Guarded object

Table C.1: UMLsec stereotypes, together with tags and constraints [12].

The $\langle\langle rbac \rangle\rangle$ stereotype contains the following tags: $\{protected\}$ indicates what should be protected, $\{role\}$ contains user/actor memberships in roles and $\{right\}$ indicates access right to protected resource for an indicated role. The constraint for $\langle\langle rbac \rangle\rangle$ is that actors in the diagram only perform activities for which they are authorized.

The $\langle\langle guarded \rangle\rangle$ indicates that objects with this stereotype can only be accessed through components indicated with the stereotype's $\{guard\}$ tag.