PATIENT PRIVACY AND CONSENT MANAGEMENT IN EHEALTH

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF COMPUTER ENGINEERING
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ERDEM ALPAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JUNE 2012

Approval of the thesis:

PATIENT PRIVACY AND CONSENT MANAGEMENT IN EHEALTH

submitted by **ERDEM ALPAY** in partial fulf llment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering** _____

Prof. Dr. İsmail Hakkı Toroslu
Supervisor, **Computer Engineering, METU** _____

Prof. Dr. Asuman Doğaç
Co-supervisor, **Computer Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Özgür Ulusoy
Department of Computer Engineering, Bilkent University _____

Prof. Dr. İsmail Hakkı Toroslu
Department of Computer Engineering, METU _____

Assoc. Prof. Dr. Ahmet Coşar
Department of Computer Engineering, METU _____

Asst. Prof. Dr. Osman Abul
Department of Computer Engineering, TOBB ETU _____

Asst. Prof. Dr. Aybar Acar
Informatics Institute, METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ERDEM ALPAY

Signature :

ABSTRACT

PATIENT PRIVACY AND CONSENT MANAGEMENT IN EHEALTH

Alpay, Erdem

M.S., Department of Computer Engineering

Supervisor : Prof. Dr. İsmail Hakkı Toroslu

Co-Supervisor : Prof. Dr. Asuman Doğaç

June 2012, 79 pages

Health information of patients are preserved either in Electronic Health Records (EHR) repositories which are generally managed in national level or in local hospital systems. However, the real owners of the data are always the patients themselves, without depending where or by whom the data is preserved. Patients should have the rights to permit or deny the access of modif cation of their information to whoever they want. Here comes the concept of Consent. Consent means provision of approval or agreement, after thoughtful consideration. Decisions of patients about sharing their information are collected and preserved in consent documents. These consent documents can be stored in different formats. The eXtensible Access Control Markup Language (XACML) def nes the policy language for this purpose. Also there is another language def ned by XACML called Request/Response Language for creating request to access information and response to reply requests. Even though XACML is the most appropriate standard for conserving consent documents, it has some weak points when used in practical systems. In the f rst part of this study, a new model based on XACML is designed. This model is easily convertable to XACML and vice versa. Then a Consent Management tool is designed using the new model. This tool has two parts, Basic Consent Editor and Consent Manager. Basic Consent Editor is aiming to provide a practical user interface for creating and managing consent documents. Consent Manager on the other hand plays a

decision mechanism role which handle requests and create decision responses according to already created consent documents. In this study, three different tools are implemented based on the Consent Management tool, each for different purposes on different projects. Throughout these implementations, usability and possible extensibility of Consent Management tool is analysed.

Keywords: Consent, Patient Privacy, XACML, SAML, Security

ÖZ

ESAĞLIKTA HASTA VERİSİ GİZLİLİĞİ VE ONAM YÖNETİMİ

Alpay, Erdem
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. İsmail Hakkı Toroslu
Ortak Tez Yöneticisi : Prof. Dr. Asuman Doğaç
Haziran 2012, 79 sayfa

Hastaların sağlık kayıtları ya ulusal çapta yönetilen elektronik sağlık kaydı depolarında, ya da hastanelerin yerel sistemlerinde muhafaza edilmektedir. Ancak bu veriler, verilerin nerede veya kim tarafından muhafaza edildiğinden bağımsız olarak hastaya aittir. Hastaların, bilgilerine erişmek veya güncellemek isteyen kişilere izin verme veya engelleme hakları bulunmalıdır. Bu noktada Onam kavramı devreye girmektedir. Onam istek ve irade ile verilen kabul veya onay anlamına gelmektedir. Hastaların kendi bilgilerini paylaşma hakkındaki kararları onam dokümanlarında saklanmaktadır.Bu onam dokümanları farklı formatlarda saklanabilir. eXtensible Access Control Markup Language (XACML) bu amaç için Poliçe Dili'ni tanımlamıştır. Ayrıca XACML tarafından verilere erişim için talep göndermek ve bu taleplere cevap göndermek için Talep/Cevap Dili adında bir dil daha tanımlanmıştır. Her ne kadar XACML onam dokümanlarını saklamak için en uygun standard olsa da pratik sistemlerde bazı zayıf noktaları ortaya çıkmaktadır. Bu çalışmanın ilk aşamasında XACML tabanlı yeni bir model dizayn edilmiştir. Bu model XACML'e veya XACML bu modele kolaylıkla çevrilebilmektedir. Daha sonra bu modeli kullanan bir Onam Yönetim aracı tasarlanmıştır. Bu araç Temel Onam Editörü ve Onam Yöneticisi olmak üzere iki kısımdan oluşmaktadır. Temel Onam Editörü onam dokümanlarını oluşturmak ve yönetmek için pratik bir ara yüz sağlamayı hedef emektedir. Diğer taraftan Onam Yöneticisi gelen talepleri karşılayıp daha önceden yaratılmış

onam dokümanlarına göre cevap hazırlayan bir karar mekanizması rolü üstlenmektedir. Bu çalışma kapsamında Onam Yönetim aracı temel alınarak, her birinin farklı amacı olan farklı projelerden üç farklı araç geliştirilmiştir. Bu araçların geliştirilmesi sürecinde Onam Yönetim aracını kullanılabilirliği ve geliştirilebilirliği analiz edilmiştir.

Anahtar Kelimeler: Onam, Hasta Gizliliği, XACML, SAML, Güvenlik

To my dearest family...

ACKNOWLEDGMENTS

I would like to express gratitude to my supervisor, Prof. Dr. İsmail Hakkı Toroslu for his guidance and support during my study.

I would like to express my sincere gratitude and appreciation to my co-supervisor Prof. Dr. Asuman Doğaç for her encouragement and support throughout this study.

I am deeply indebted to my dear wife Esra Alpay for her love, continued motivating support and welcomed presence. Without her, this work could not have been completed.

I am deeply grateful to my parents and to my brother for their love and life-long support. Thanks for giving me a shoulder to lean on whenever I need.

I am also grateful to my friends, Fulya Tunçer, Gökçe Banu Laleci Ertürkmen, Mustafa Yüksel, Tuncay Namlı, Yıldıray Kabak and all the other colleagues at the Software Research and Development Center, whose help, stimulating suggestions and encouragement helped me in all the time of research for and writing of this thesis.

Finally I would thank the Scientif c and Technological Research Council of Turkey (TÜBITAK) for providing the f nancial means throughout this study.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ vi

ACKNOWLEDGMENTS ix

TABLE OF CONTENTS x

LIST OF TABLES xiii

LIST OF FIGURES xiv

CHAPTERS

1 INTRODUCTION 1

2 BACKGROUND ON ENABLING TECHNOLOGIES AND STANDARDS . 4

2.1 Security Assertion Markup Language(SAML) 4

2.2 Extensible Access Control Markup Language(XACML) 6

2.2.1 Policy Language Model 8

2.2.1.1 Rule 8

2.2.1.2 Policy 11

2.2.1.3 Policy Set 12

2.2.2 XACML Security Assertion Markup Language Prof le . . 13

3 CONSENT DOCUMENT MODEL 17

3.1 Consent Document 17

3.2 Consent Rule 17

3.3 Consent Rule Target 20

3.3.1 Subjects 20

3.3.2 Resources 22

3.3.3 Actions 23

3.4 Conditions and Obligations 23

4 DESIGN, IMPLEMENTATION AND FEATURES OF CONSENT EDITOR AND MANAGER 24
4.1 Use Cases 26
4.1.1 Use Cases of Consent Editor 26
4.1.1.1 Create/Update Consent Document 26
4.1.1.2 Create/Update Consent Document 29
4.1.1.3 Set Current Consent Document 29
4.1.1.4 Update Subjects 30
4.1.1.5 Update Resources 31
4.1.1.6 Download XACML Document 31
4.1.2 Use Cases of Consent Manager 31
4.1.2.1 Get Decision 31
4.1.2.2 Consent Manager Web Service 35
4.2 Consent Editor Implementations 36
4.2.1 iCardea Consent Editor 36
4.2.1.1 Integration to PHR System - PHR Interface . 38
4.2.1.2 Integration to Care Management Database System 42
4.2.1.3 iCARDEA Consent Editor User Interface . . . 43
4.2.2 eSağlıkKaydım Consent Editor 51
4.2.3 epSOS Consent Editor 55
5 RELATED WORK 58
5.1 Principled Electronic Consent Management 58
5.1.1 First Generation ECM 59
5.1.2 Ex-post ECM 59
5.1.3 Principaled ECM 60
5.2 Privacy and Security Shield for Health Information Systems 61
6 CONCLUSION 63
REFERENCES 65
APPENDICES

A XACML Document 67

LIST OF TABLES

TABLES

Table 3.1 Consent Document Meta Data Fields 18

Table 3.2 Consent Rule Fields 20

Table 3.3 Individual Subject Fields 22

Table 3.4 Group Subject Fields 22

Table 3.5 Resource Fields 23

Table 4.1 IHE Care Management Prof le Care Provision Codes 44

Table 4.2 Consent Manager Resource to IHE CM Prof le Resources Mapping 45

LIST OF FIGURES

FIGURES

Figure 2.1 SAML Transaction Steps 5

Figure 2.2 XACML Context 8

Figure 2.3 XACML Policy Structure 9

Figure 2.4 XACML Policy Language Model 10

Figure 2.5 XACML Request Structure 13

Figure 2.6 XACML Response Structure 13

Figure 2.7 SAML Queries and Statements 15

Figure 3.1 Consent Document 18

Figure 3.2 Consent Rule 19

Figure 3.3 Consent Rule Target 21

Figure 4.1 General View of the System 25

Figure 4.2 Server Side 25

Figure 4.3 Use Case Diagram 27

Figure 4.4 Create/Update Consent Document Sequence Diagram 28

Figure 4.5 Delete Consent Document Sequence Diagram 29

Figure 4.6 Set Current Consent Document Sequence Diagram 30

Figure 4.7 Update Subjects Sequence Diagram 32

Figure 4.8 Update Resources Sequence Diagram 33

Figure 4.9 Download XACML Document Sequence Diagram 34

Figure 4.10 Get Decision Call Types 34

Figure 4.11 Get Decision Sequence Diagram 35

Figure 4.12 'getDecision' Operation of Consent Manager Web Service 36
Figure 4.13 iCARDEA Architecture Overview 37
Figure 4.14 iCARDEA Consent Editor User Interface 45
Figure 4.15 Consent Document Panel View 46
Figure 4.16 Register Consent Document View 47
Figure 4.17 Matrix View of Consent Document 48
Figure 4.18 Consent Rule Panel View 48
Figure 4.19 Conditions in Consent Rule Panel View 49
Figure 4.20 All Consent Documents Panel View 50
Figure 4.21 Admin Panel View 51
Figure 4.22 Admin Panel Edit Individual View 52
Figure 4.23 Menu View 52
Figure 4.24 epSOS Consent Rule 54
Figure 4.25 eSağlıkKaydım Consent Document Matrix View 55
Figure 4.26 epSOS Consent Editor 57

Figure 5.1 The eConsent Process 60
Figure 5.2 Conceptual Architecture 61

CHAPTER 1

INTRODUCTION

Due to the extremely sensitive nature of patient related information, both medical records and patient's context information cannot be disclosed indiscriminately and different healthcare providers must have different access rights. The patient should be at the center of this process controlling his/her consent.

European countries are focused on a common aim which is providing quality healthcare to all its citizens. The benef ts of accessing patient information through this aim are unignorable. This will not only increase the efficiency of the treatment, but also improve the quality of healthcare systems. However, it has to be noted that the security of the patients' health information is at least as important as being accessible, if it is not more important. So enabling access to patients' health information requires proper security mechanism and systems.

To provide these conf dentiality and privacy mechanisms, a patient controlled, conf gurable, patient context dependent privacy mechanism is developed. Furthermore, a framework for establishing trust mechanisms among the actors involved is provided based on advanced identity management mechanisms. Indeed, a very important part of enforcing patient consent is to be able to identify the functional role of the party trying to access the patient information. Currently there are access control mechanisms used for this purpose within a single organization. However, for its dynamic and distributed environment, new tools empowering the patient to decide what specif c types of his/her context and clinical information can be provided to which entities involved in his/her follow-up is provided. OASIS Extensible Access Control Markup Language (XACML)[2] for access control is used to implement the patient privacy and security mechanisms.

The security and privacy of patient's information depends on different concepts to be con-

sidered such as identity management[8], authorization[16], access control, trust and privacy. These concepts are already active research and development areas, especially in the eBusiness domain. The major concern in eBusiness applications is the privacy of the customers. In addition, in the healthcare domain the problem is extended to providing the privacy of the EHRs to be accessed.

Related to the authorization and access control, OASIS XACML standard and IBM Enterprise Authorization Language (EPAL)[6] are the two major industry specif cations. Both EPAL and XACML share an abstract model for policy enforcement def ned by the IETF[12] and ISO[18]. XACML provides more features like combining the result of multiple policies, ability to reference other policies, ability to return separate results for each node when access to a hierarchical resource (f ne-grained access control), and support for attribute values those are instances of XML schema elements, which are needed for constructing complex policies[2].

Today the healthcare sector is still using paper based consents usually within a single organisation with very limited patient control. For EHR sharing, the networked health information systems or individual healthcare enterprises mostly use the opt-in/opt-out model which either deny the sharing of all records with outside or allows all accesses. The IHE initiative published a prof le in 2006, Basic Patient Privacy Consent (BPPC)[17], which provides more choices to patients regarding the sharing of EHR data in IHE[1] document sharing platform.

Healthcare domain presents new research problems to the identity management and privacy issues. Identity Management solutions have not been deployed in healthcare networks yet. Today's healthcare networks generally assume a transitive trust model that is "if I trust the Healthcare Institute A, I also trust the employees of Healthcare Institute A". However, this strong trust relationship may cause privacy leakages.

For the authorization and access control, XACML seems to be the best candidate for def ning policies. Healthcare data may not be under the control of a single party, and thus privacy policies of different issuers; policies of legal entities, policies of healthcare enterprises, and patient consent policies need to be taken into account. There could be conf icts among these privacy policies. Some research by University of Milano, by IBM T.J. Watson Research Center has been performed on this issue; however, there is no practical application so it is still largely an open research problem.

In the BPPC profile published by IHE initiative, the patient can choose the policies defined by the legal entity representing the health information network, not able to define its own consent or further restrictions. In addition, the profile does not define the policy structure how the access control is applied.

Related with security and privacy, this thesis addresses challenges, including defining fine-grained document sensitivity levels and functional roles for healthcare providers as the basis of patient-controlled attributes in privacy consents; enforcing patient requested obligations for privacy consents; handling patient and healthcare provider context; assessing risks and damages through the novel federated audit mechanisms; establishing trust among the actors and all the involved interoperability problems.

This thesis is organized as follows: In Chapter 2 the main technologies that have been used in this thesis are presented. Chapter 3 describes the document model which is developed for this thesis work. Chapter 4 is devoted to the description of the overall privacy system developed in the scope of thesis work. Chapter 5 summarizes the related work by emphasizing the innovative aspects of the proposed solution.Finally Chapter 6 concludes the thesis.

CHAPTER 2

BACKGROUND ON ENABLING TECHNOLOGIES AND STANDARDS

2.1 Security Assertion Markup Language(SAML)

SAML is an XML based language which provides a standard way of expressing security information[10]. SAML standard def nes rules and syntax for data exchange. It is also f exible that enables custom data transmission to external service provider[13]. It can express the following type information:

- authentication information
- authorization information
- attribute information

SAML is used to facilitate Single Sign-on process[13][21]. Single Sign-on is the ability of using a login information with several services with out needing an extra authentication. After your sign-in operation, your authentication information is passed between each application you visit. SAML provides several advantages improved online experience for users. With SAML Single Sign-on users can authenticate only to an identity provider and then access different services with this authentication information. SAML also improves task distribution by giving right task for right person. For example responsibility for the management of identities

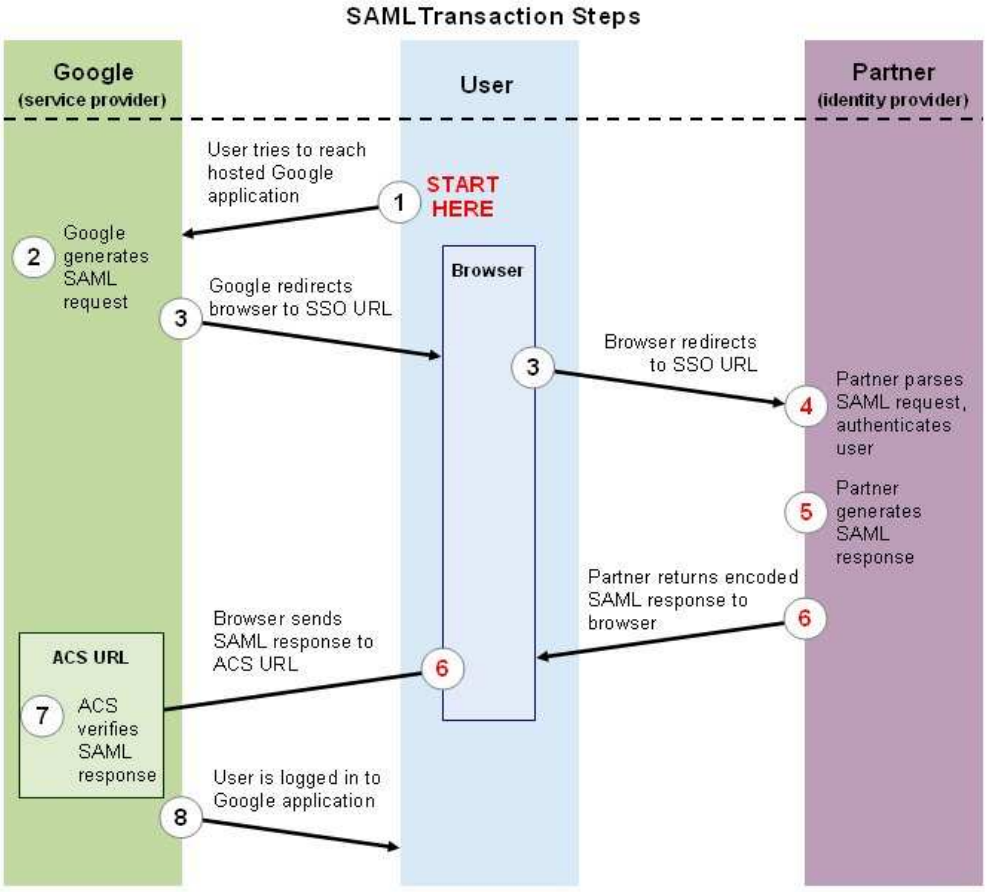


Figure 2.1: SAML Transaction Steps

is covered by the identity provider, which is a more suitable candidate with respect to service provider. Moreover, SAML reduces the unnecessary administrative costs of service providers. Service providers does not have to collect and store identity-related data such as passwords, etc.

SAML Assertion

The authentication information of the patient are provided as an Assertion by the Identity Provider. A SAML Assertion contains common information that applies to assertion as a whole, followed by several statements. There are 3 kinds of SAML Assertions[2]:

- Authentication Assertion (identity information)
  This assertion contains information about the authentication of a subject. It contains authenticatior, authentication date, subject that is authenticated, assertion validity time, method used to authenticate and the time assertion is created.
- Attribute Assertion
  This assertion contains additional information as attributes.
- Authorization Decision Assertion
  This assertion contains decision information about the subjects access rights.

2.2 Extensible Access Control Markup Language(XACML)

XACML (eXtensible Access Control Markup Language) is an XML-based language for access control that has been standardized in OASIS[2]. XACML describes two different languages for both def ning the how consent document needs to be stored (policy language), and how request/response messages should be to apply to grant access to the information. The policy language is used to express access control policies (**Who** can access **what**, under what **conditions**, and for what **purpose**). The request/response language expresses queries about whether a particular access should be allowed (requests) and describes answers to those queries (responses).

In XACML, there are several actors such as Policy Enforcement Point (PEP), Policy Decision

Point (PDP), etc[22]. PEP is the system that receives a request from a subject to take some action on a resource. This subject may be a group of person, an individual, or even may not be a real person. PEP takes this request and transforms it into an XACML Request to send it to the PDP. PDP is the main decision point of the system. It retrieves an XACML Request, checks it according to consents which are stored in policy language. It decides whether to permit or deny the request, and creates an XACML Response containing the repsonse information to send back to the PEP. Finally PEP receives the XACML Response, processes it and applies or does not applies the action of subject to the resource according to the receives XACML response.

"XACML has many benef ts over other access control policy languages [3]:

- One standard access control policy language can replace dozens of application-specif c languages
- Administrators save time and money because they don't need to rewrite their policies in many different languages
- Developers save time and money because they don't have to invent new policy languages and write code to support them. They can reuse existing code
- Good tools for writing and managing XACML policies are being developed, since they can be used with many applications
- XACML is f exible enough to accommodate most access control policy needs and extensible so that new requirements can be supported.
- One XACML policy can cover many resources. This helps avoid inconsistent policies on different resources.
- XACML allows one policy to refer to another. This is important for large organizations. For instance, a site-specif c policy may refer to a company-wide policy and a country-specif c policy."

XACML is intended to be suitable for a variety of application environments. The core language is insulated from the application environment by the XACML context, as shown in Figure 2.2, in which the scope of the XACML specif cation is indicated by the shaded area.

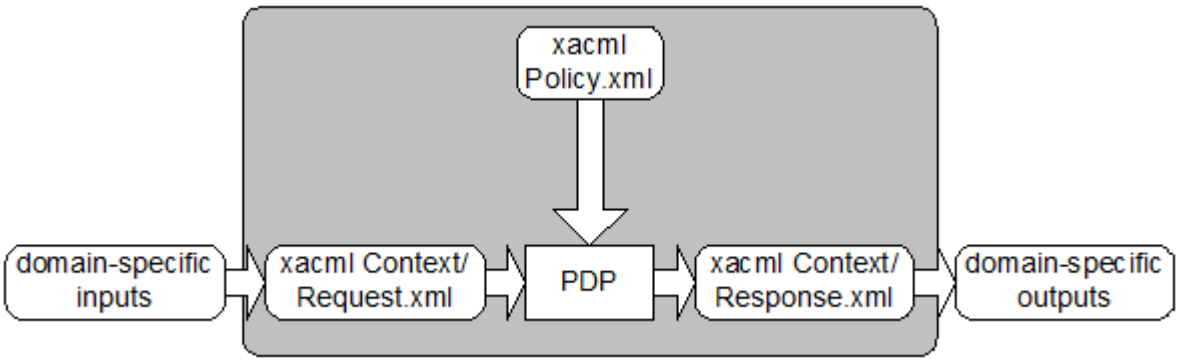


Figure 2.2: XACML Context

The PDP is not required to operate directly on the XACML representation of a policy. It may operate directly on an alternative representation. Decision Policy Service of Basic Consent Editor is working both on XACML representation and on customized version of it, which will be explained in detail in chapter 4.

2.2.1 Policy Language Model

The policy language model is shown in Figure 2.4. It is composed of Rule, Policy and Policy set.

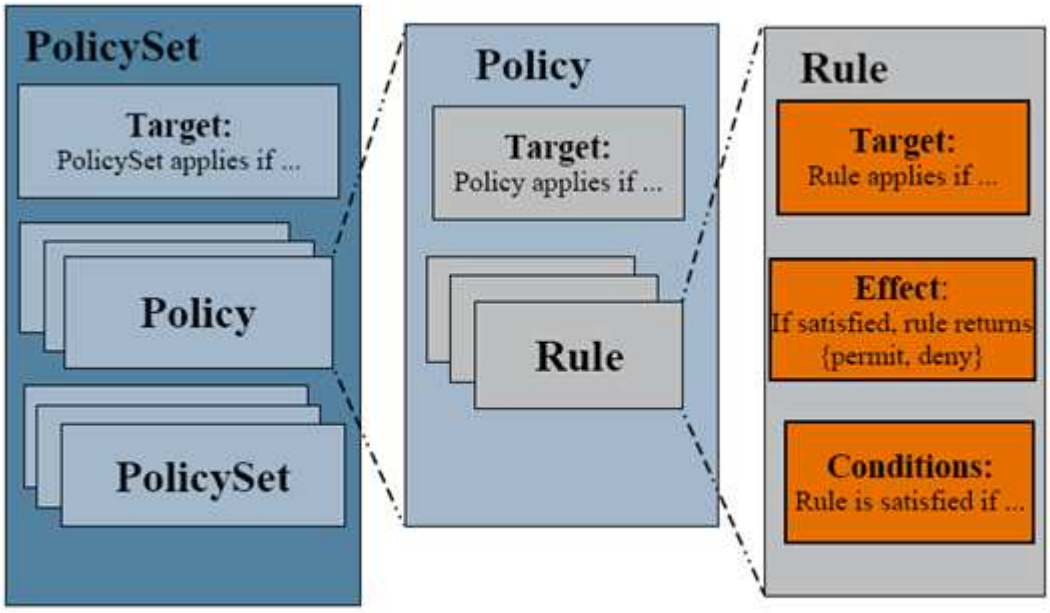
2.2.1.1 Rule

Rule is the basic unit of policy. However it cannot exist alone in an XACML domain. It has to be encapsulated with Policy which is another main component of policy model. The main components of a rule are a target, an effect and a condition.

Rule Target

The target def nes the set of:

- resources
- subjects



Copyright © 2007 Sun Microsystems, Inc. All rights reserved.

Figure 2.3: XACML Policy Structure

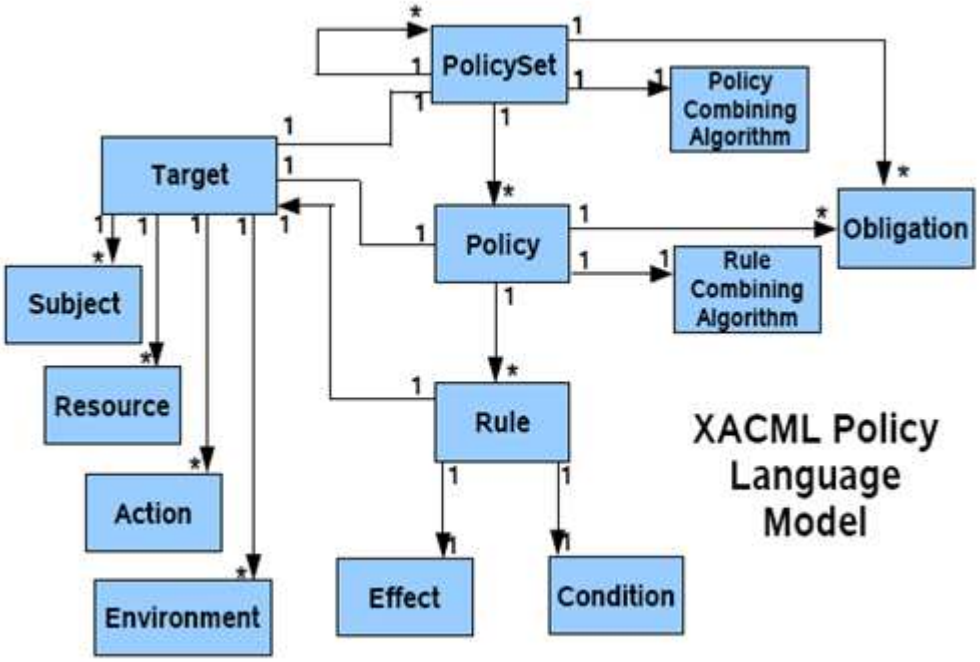


Figure 2.4: XACML Policy Language Model

- actions
- environment

to which the rule is intended to apply. The target of a rule is restricted by the Condition element. However, if the rule is not intended to be restricted, then any Condition entity is ommitted from the rule. Policy Decision Point(PDP) compares the content of a rule with the subjects, resources, actions and environment attributes of the context of a request, to check whether target of rule matches with request. If the match exists, than the effect of the rule applies on the request. If the target elements is omitted from a rule, it inherits the target of the parent Policy element.

Effect

The effect of the rule is the decision that is applied to request when the context of a request matches with the content of the rule. Two values are allowed: "Permit" and "Deny".

Condition

Conditions are the additional restriction criterias of the target which rule applies. Even the subject, resource and action of rule target matches with the request context, if the condition does not validate, than the rule does not apply the effect to the incoming request. It may be omitted.

2.2.1.2 Policy

Even though the rules are the basic unit of policies, they are not designed to be used alone, they only exist in the policies which are exchanged between system entities. Therefore, a Policy Administration Point(PAP)[22][19] combines rules in a policy. A policy comprises four main components:

- obligations
- a rule-combining algorithm
- a set of rules
- target

The components other than rules are described in the following sections:

Policy Target

A Policy element may contain a Target element that specif es the set of subjects, resources, actions and environments to which it applies. If a Rule in a Policy has the same Target of its parent Policy element, it can omit its Target element. Rules which does not have any Target element, inherit their Target from parent Policy elements. Target element of Policy may be def ned by directly within Policy or it may be calculated from the Target elements of the Rules that the Policy contains (union or intersection).

Rule-combining algorithm

The rule-combining algorithm specif es the result of a policy with several rules resulting different decisions. In other words, the Decision value placed in the response context by the PDP is the value of the policy, as def ned by the rule-combining algorithm. For example, "deny-overrides" algorithm return "Deny" as decision if any of the rules return "Deny" as decision. Even though 9 of 10 rules results "Permit", one single "Deny" overrides other rules and the makes the Decision of the Policy as "Deny". A policy may have combining parameters that affect the operation of the rule-combining algorithm.

Obligations

Obligations are the operations needs to be applied whenever the policy target matches with the request context.

2.2.1.3 Policy Set

A policy set comprises four main components:

- obligations;
- policy-combining algorithm
- set of policies
- target.

Policy-combining algorithm

The policy-combining algorithm is the same as the rule-combining algorithm, however this time it applies to Policy elements in a PolicySet respect to Rule elements in a Policy.



Figure 2.5: XACML Request Structure



Figure 2.6: XACML Response Structure

Request and Response

The structures of XACML Request and Response documents are presented in Figure 2.5 and in Figure 2.6.

2.2.2 XACML Security Assertion Markup Language Profile

XACML has several prof les for several different uses. Security Assertion Markup Language (SAML) which is product and standard of OASIS is one of those prof les and is suitable for providing the assertion and protocol mechanisms needed by XACML. The schemas of SAML

are used for both requesting and responding with various types of security assertions. They also include information needed to identify and validate the contents of the assertions, which are the identity of the assertion issuer, the validity period of the assertion, and the digital signature of the assertion. The SAML specif cation describes how these elements are to be used.

> "SAML is used for several purposes such as to protect, transport, and request information needed by an XACML implementation. There are 6 types of queries and statements used in this prof le all of which are stated as follow[7]:
>
> 1. AttributeQuery - A standard SAML Request used for requesting one or more attributes from an Attribute Authority.
> 2. AttributeStatement - A standard SAML Statement that contains one or more attributes. This statement may be used in a SAML Response from an Attribute Authority, or it may be used in a SAML Assertion as a format for storing attributes in an Attribute Repository.
> 3. XACMLPolicyQuery - A SAML Request extension, def ned in this prof le. It is used for requesting one or more policies from a Policy Administration Point.
> 4. XACMLPolicyStatement - A SAML Statement extension, def ned in this prof le. It may be used in a SAML Response from a Policy Administration Point, or it may be used in a SAML Assertion as a format for storing policies in a Policy Repository.
> 5. XACMLAuthzDecisionQuery - A SAML Request extension, def ned in this prof le. It is used by a PEP to request an authorization decision from an XACML PDP.
> 6. XACMLAuthzDecisionStatement - A SAML Statement extension, def ned in this prof le. It may be used in a SAML Response from an XACML PDP. It might also be used in a SAML Assertion that is used as a credential, but this is not part of the currently def ned XACML use model."

All queries and statements between different points are presented in the Figure "SAML Queries and Statements".

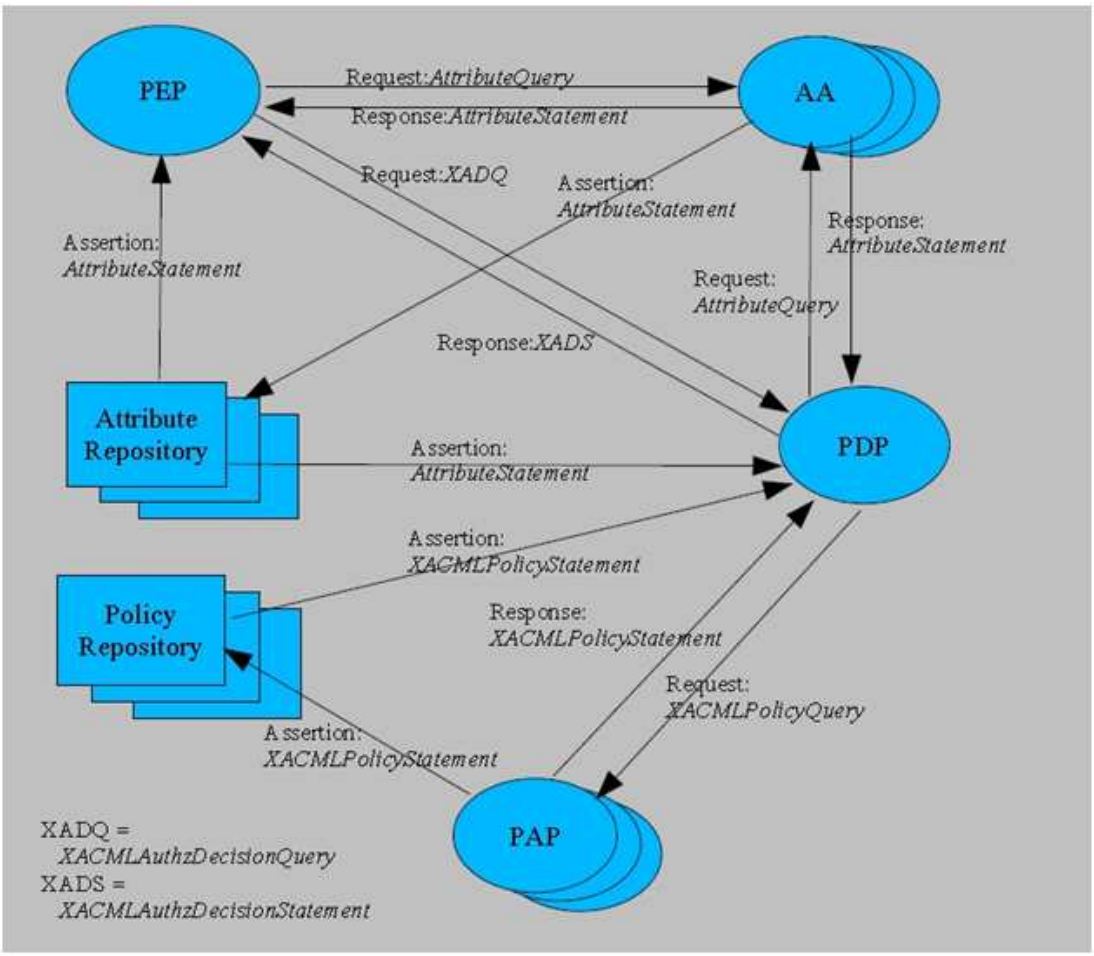


Figure 2.7: SAML Queries and Statements

In the Consent Manager mechanism, not all of these queries and statements are used. For example, PAP is simulated as a registry (an SQL Database), hence XACMLPolicyQuery and XACMLPolicyStatements are not used in Consent Manager. Actually, the only used SAML Prof le messages are XACMLAuthzDecisionQuery and XACMLAuthzDecisionStatements. XACMLAuthzDecisionQuery and XACMLAuthzDecisionStatements are used in the web service part of the Consent Manager mechanism. "getDecision" operation of the Consent Manager web service is waiting for a XACMLAuthzDecisionQuery message. This message contains the XACMLRequest inside. In other words, it can be said that XACMLAuthzDecisionQuery message wraps the XACMLRequest. Furthermore, it may contain any assertions that is used in proving the identity of the requester. "getDecision" operation send an XACMLAuthzDecisionStatement message back which also wraps an XACMLResponse. Details of the webservice and operations will be explained in detail later in this document.

CHAPTER 3

CONSENT DOCUMENT MODEL

Within the development of Basic Consent Editor, another model is needed along the XACML Policy Model. Despite the fact that XACML provides a powerful and extensive model, a new model based on XACML is useful to provide completeness and simplicity. This model can easily be translated to XACML or vice versa. Furthermore, decision engine developed within Basic Consent Editor, can support both types of models while validating requests.

3.1 Consent Document

There exist a Consent Document at top of the Consent Document Model. Consent document is a container which contains several consent rules. Consent Document contains a metadata element which holds the characteristic information of consent document and a list of Consent Rules.

Description about the f elds of Consent Document Meta Data is given in Table 3.1.

3.2 Consent Rule

Consent rule is a control mechanism over the information of patients to specify **who** can access to **what** for what **purpose** under which **condition**.

- Consent rule contains a consent rule target element and lists of conditions and obligations.

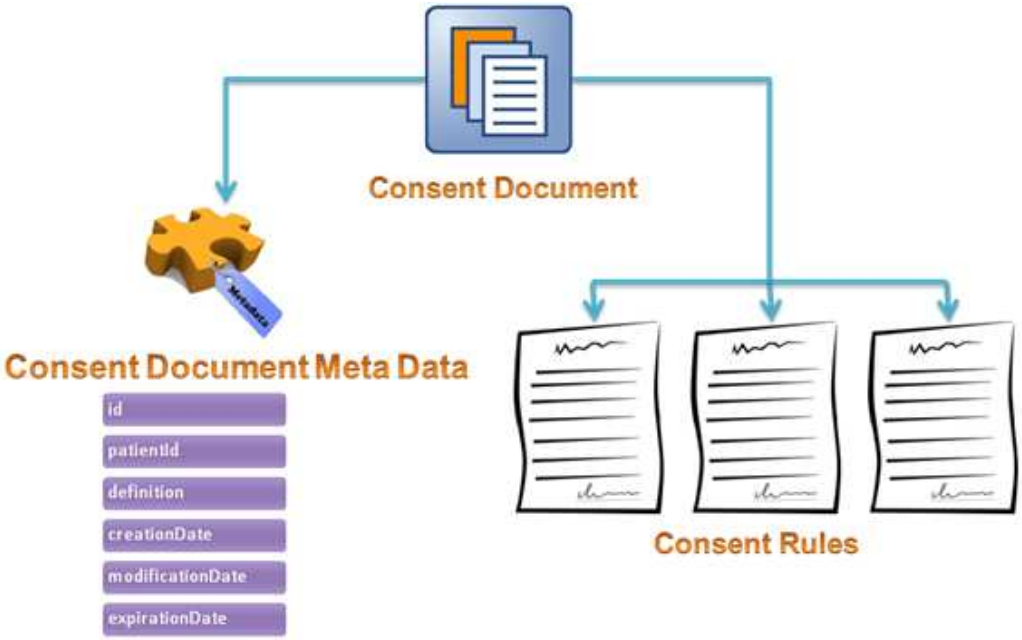


Figure 3.1: Consent Document

Table 3.1: Consent Document Meta Data Fields

Field	**Description**
id	The unique consent id of a consent document. Every consent document has a different id for every user, hence even different users cannot have same consent id.
patientID	The patient id of a consent document. It specif es that whom the consent document is belong to.
def nition	Def nition (description) of the consent document.
creationDate	The date when the consent document is created.
modif cationDate	The date when the consent document has been updated.
expirationDate	The date when the consent document expires(loses its validity).

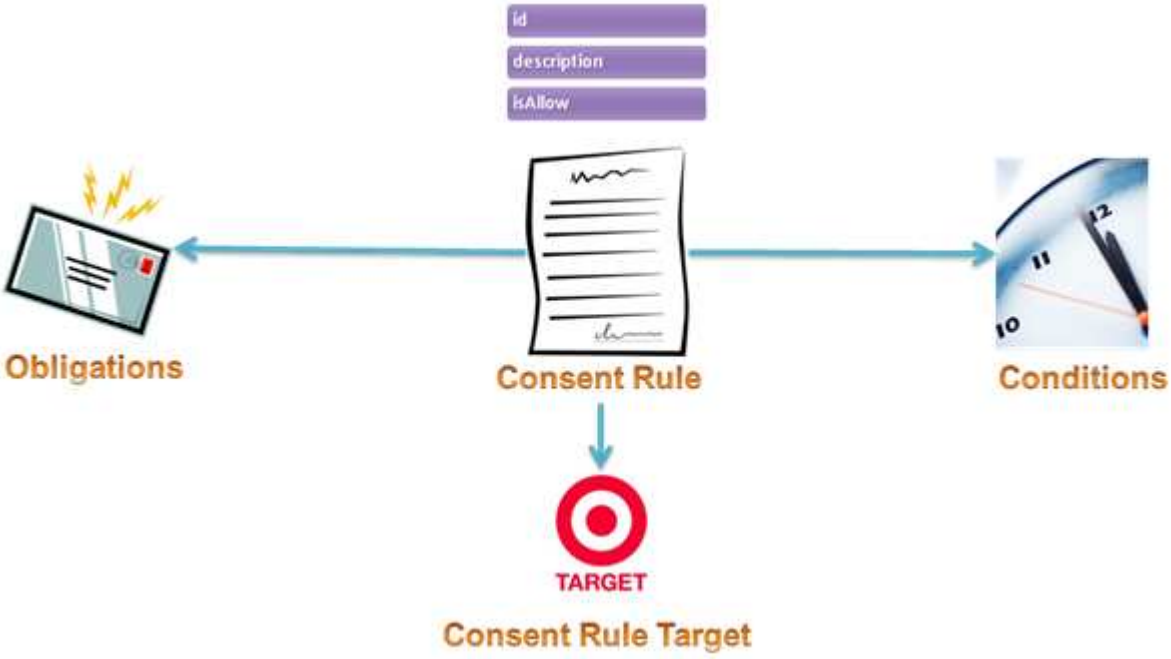


Figure 3.2: Consent Rule

- Consent rule target is a container which contains subjects, resources and actions of consent rule.

- Conditions are the restrictions of the target elements, in other words even though a request matches with consent rule target, if condition does not satisfy, the effect does not applied to the request.

- Obligations specify the implications that needs to be applied if a rule satisf es (e.g. send a mail to patient after her record is accessed).

- Consent rule also has a f eld called isAllow which specif es the effect (permit or deny) when the target is matched with request.

Description about the f elds of Consent Document Rule is given in Table 3.2.

Table 3.2: Consent Rule Fields

Field	Description
id	The unique consent id of a consent rule. Every consent rule has a different id for every document. Even different documents cannot contain consent rules with same id.
description	Description of the consent document.
isAllow	Specif es the effect of the rule provided that the rule is satisf ed.

3.3 Consent Rule Target

Consent rule target is a container which contains target elements of a consent rule. These target elements can be separated under three titles:

- Subjects
- Resources
- Actions

If we want to clarify all target elements brief y;

- Subjects are target person of the rule.They can be individual people or a group of person who are grouped according to their roles.
- Resources are the target object of the rule. It can be anything related with medical information of user such as ’Allergies’ or ’Medications’.
- Actions are the target purpose of the rule. There are limited default actions, however, new actions can be def ned like subjects and resources if needed.

3.3.1 Subjects

There are two types of subjects: Group and Individual. Group refers to some roles such as Doctor, Pharmacist, Family Member etc. ’Group’ may also include specif c members. However, ’Individual’ refers to a specif c person such as "Dr. Gregory House". ’Individual’

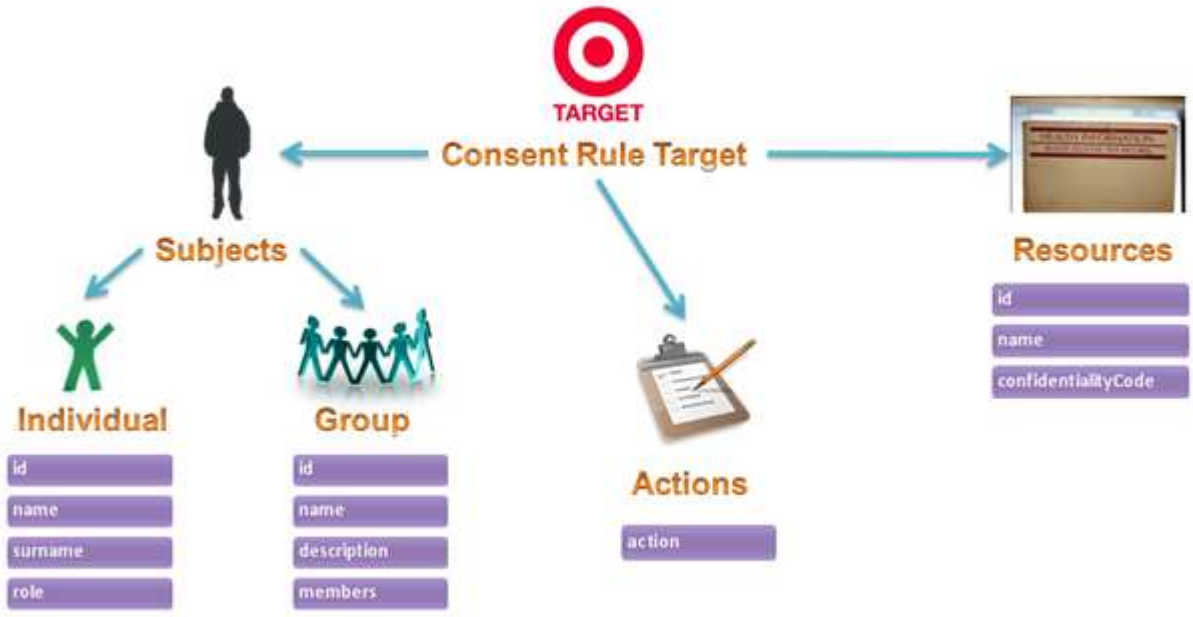


Figure 3.3: Consent Rule Target

may also have a specif ed role. This is facilitating to prepare rules such as "All doctors but Dr. XYZ can read my test results".

There are 6 groups def ned by default. However, number of groups may be extended by system administrators. Default group ids are listed below:

- ROLECODE:DOCTOR
- ROLECODE:NURSE
- ROLECODE:FAMILYMEMBER
- ROLECODE:PSYCHIATRIST
- ROLECODE:PHARMACIST
- ROLECODE:DENTIST

Description of the f elds of the subjects are provided in Table 3.3 and Table 3.4.

Table 3.3: Individual Subject Fields

Field	Description
id	The unique id of the individual.
name	Real name of the individual.
surname	Real surname of the individual.
role	Role of the individual. (Doctor, Nurse, etc...)

Table 3.4: Group Subject Fields

Field	Description
id	The unique id of the group.
name	Name of the group.
description	Description of the group.
members	Specif c members of the group (if there is any).

3.3.2 Resources

Resources are identif ed according to their resource ids. There are 8 different resources as default. However, these may be extended by the system administrators. Below, ids of these resources are provided:

- RESOURCECODE:ALLERGY
- RESOURCECODE:CONDITION
- RESOURCECODE:OPERATION
- RESOURCECODE:MEDICATION
- RESOURCECODE:TESTRESULT
- RESOURCECODE:IMMUNIZATION
- RESOURCECODE:HOSPITALVISIT
- RESOURCECODE:BASICHEALTH

Description of the f elds of the resources are provided in Table 3.5

Table 3.5: Resource Fields

Field	Description
id	The unique id of the resource.
name	Name of the resource.
conf dentialityCode	Conf dentiality Code of the resource. (HIGH, NORMAL, LOW, etc...)

3.3.3 Actions

Actions are only presented with string values which refers the name of the actions themselves. There are three default actions provided, however users may extend the list of actions. Below, names of these actions are provided:

- RESOURCECODE:READ
- RESOURCECODE:CREATE
- RESOURCECODE:UPDATE

3.4 Conditions and Obligations

Specifying who can access to what for what purpose is not always enough. Patients may want to add more specif c rules, such as "My medical information can be reached in only weekdays" or "My medical information can be accessed only from network of Hospital XYZ". For creating such an expanded rules, 'Conditions' can be used. Conditions specify the restrictions that a rule needs to satisfy. In other words, if the target of the request matches with the policy, conditions specif es if the rule is applicable to incoming request. For example, if there is a condition which restricts access of a document only in working hours (9am - 6pm), access is not allowed after 6 pm.

Another issue about the concent management is that patients may want to be informed when somebody read her medical information or somebody made changes. 'Obligations' can be used for this purpose. Obligations specify the obligations that needs to be done if a rule satisf es. For example, patients may add obligations to her policies such as "if anyone access my test results, send me an email".

CHAPTER 4

DESIGN, IMPLEMENTATION AND FEATURES OF CONSENT EDITOR AND MANAGER

The current design of the Basic Consent Editor is made for facilitating integration with any PHR implementation. It has three layers, which separate all works from each other. The presentation layer is just for providing a user friendly graphical user interface for user interaction. All interactions with the users are made through this layer. Service layer provides services to the presentation layer such as storing or getting information of the consent documents. Data access layer on the other hand, is providing the communication between the Basic Consent Editor and the PHR implementation. Data access layer provides an interface called PHR Interface to be implemented by PHR manger. It contains several methods such as "getConsentDocuments" or "setCurrentConsentDocumentOfPatient". These methods need to be implemented by the PHR manager according to type of storing PHR information. In other words, the implementation of PHR Interface depends on the case if the information is kept in the database or in f les that contain plain text (or XML). It also varies between two different implementation of PHR, both of which stores information in databases but have different database models.

In the Figure 4.1, general view of the system is presented. Client Side represents the Presentation Layer, and Server Side represents both Service Layer and Data Access Layer. PHR Database is a repository where the information is stored. Data could be preserved in any other type of repository. To see the details of the Server Side, we need to go deeper in this f gure. Server Side is presented in Figure 4.2 with more detail.

Figure 4.2 presents Service Layer and Data Access Layer separately. Consent Editor Service at left of the Server Side is Service Layer. This layer provides several methods to the Presen-



Figure 4.1: General View of the System

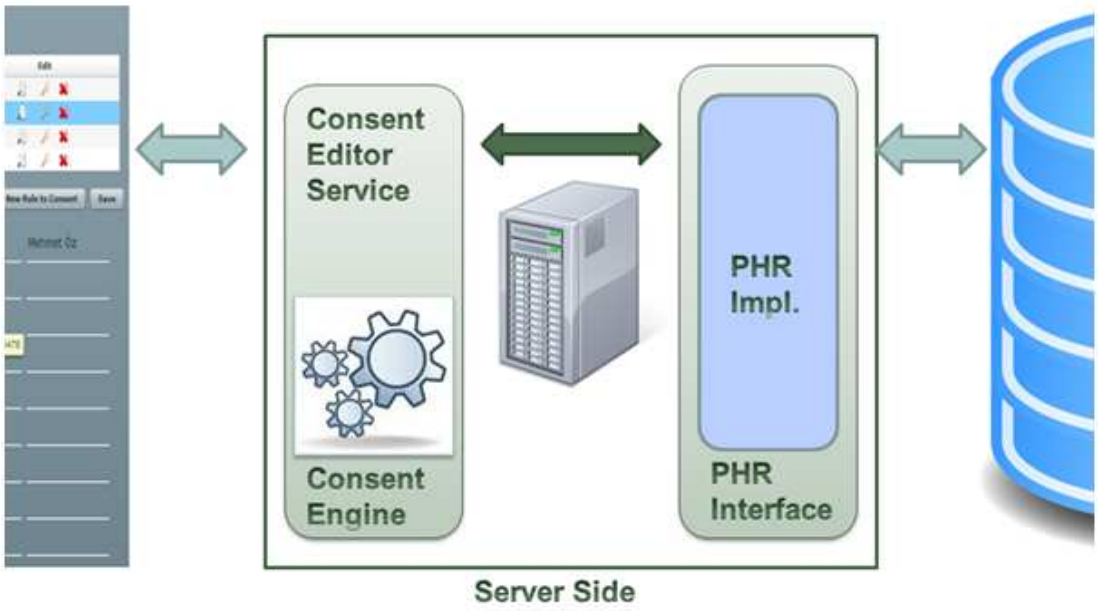


Figure 4.2: Server Side

tation Layer. However, PHR Interface is representing Data Access Layer. This part specifies how to access data according to its implementation.

4.1 Use Cases

4.1.1 Use Cases of Consent Editor

Users of the Basic Consent Editor tool are capable of creating or updating their consent documents, setting their current consent document, specifying the subjects and resources which are referenced in consent documents and downloading the XACML version of their consent documents. All these abilities are presented in six main use cases and four sub use cases:

- Create/Update Consent Document
  - Create/Update Consent Rule
  - Delete Consent Rule
- Delete Consent Document
- Set Current Consent Document
- Update Subject
  - Update Groups
  - Update Individuals
- Update Resources
- Download XACML Document

A use case diagram is presented in Figure 4.3.

4.1.1.1 Create/Update Consent Document

User can create a new consent document with several consent rules or update an existent consent document by adding new consent rules to a consent document, updating one of the consent rules of a consent document, or deleting any of the consent rules.

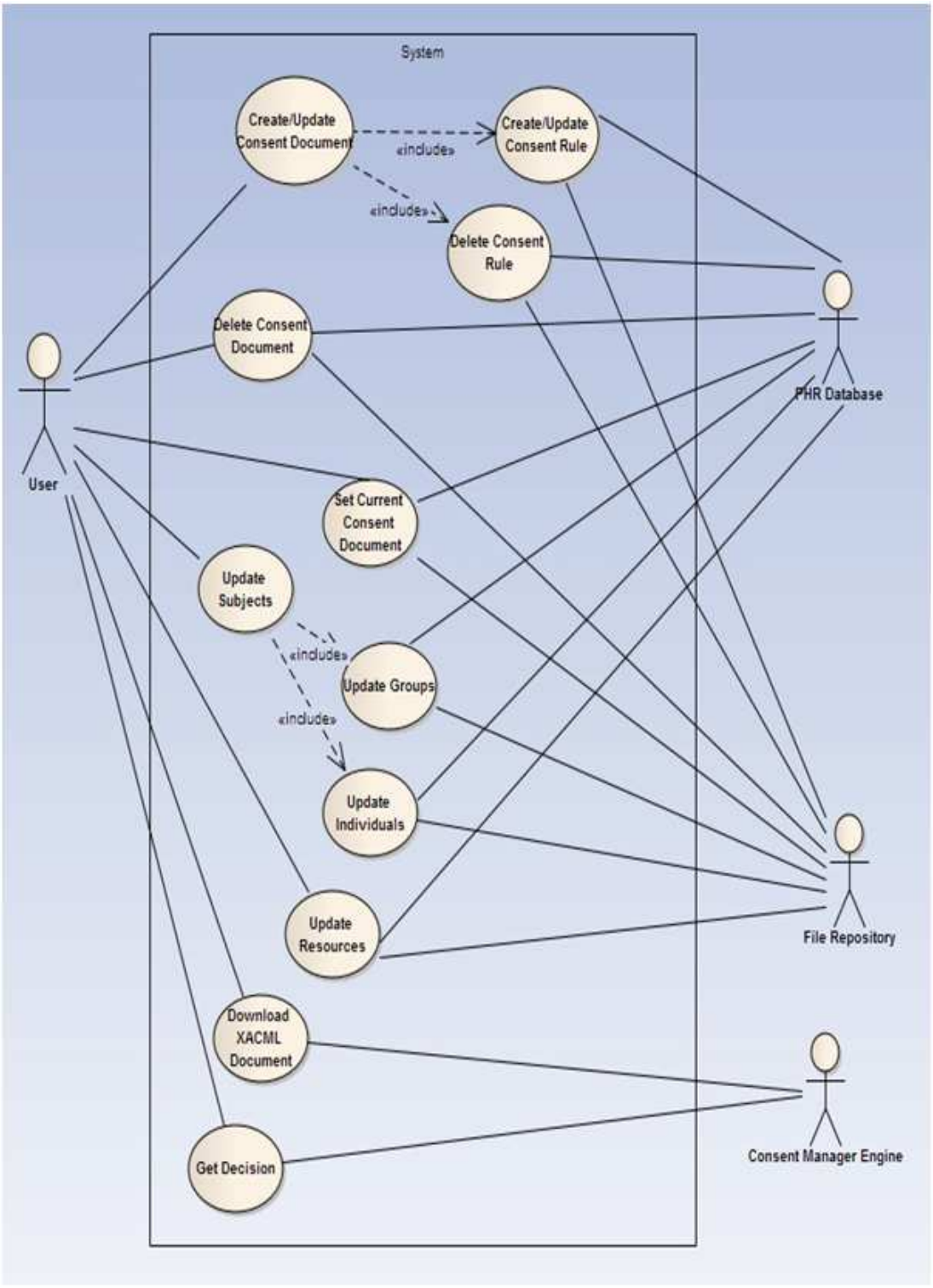


Figure 4.3: Use Case Diagram

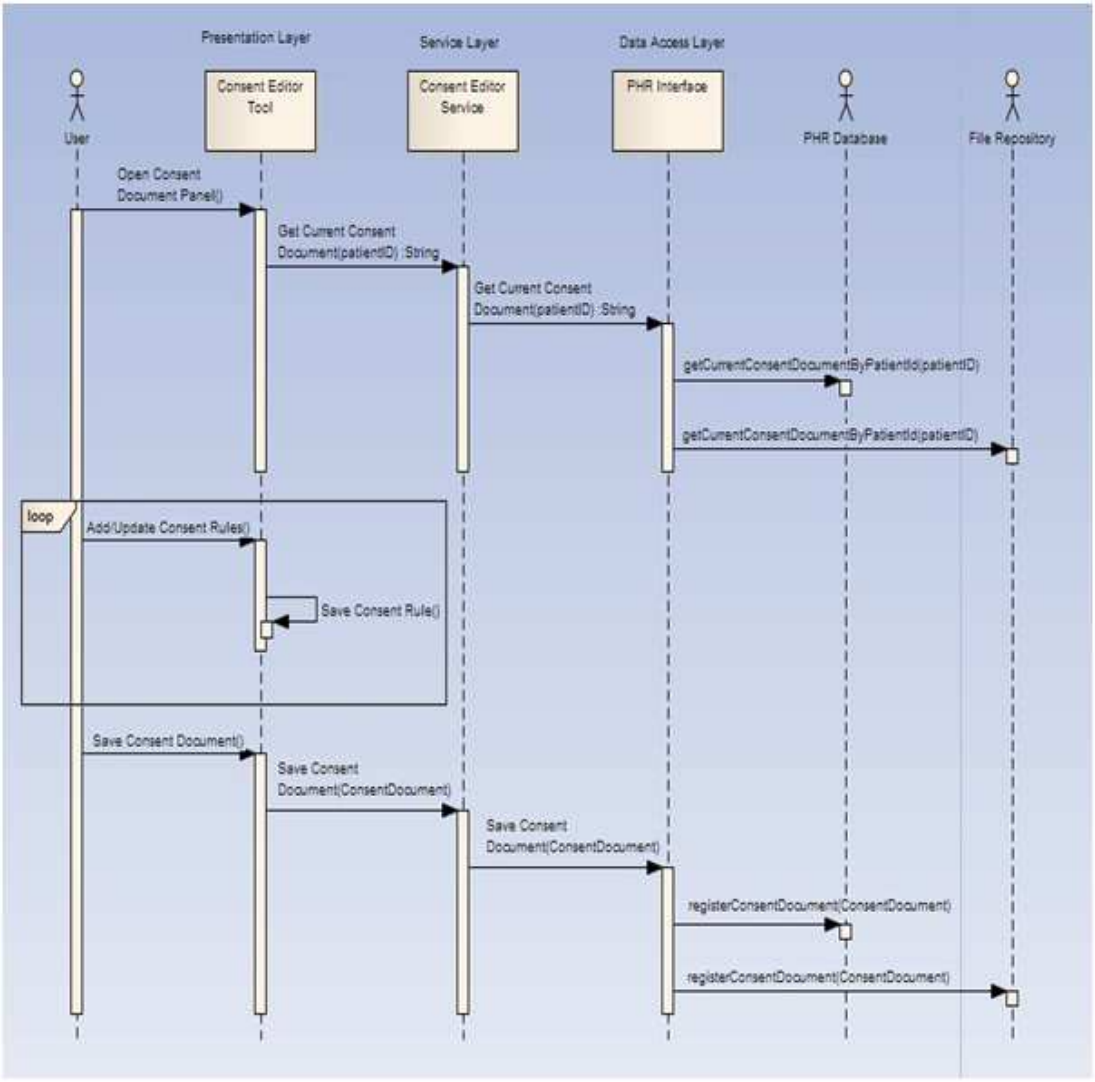


Figure 4.4: Create/Update Consent Document Sequence Diagram

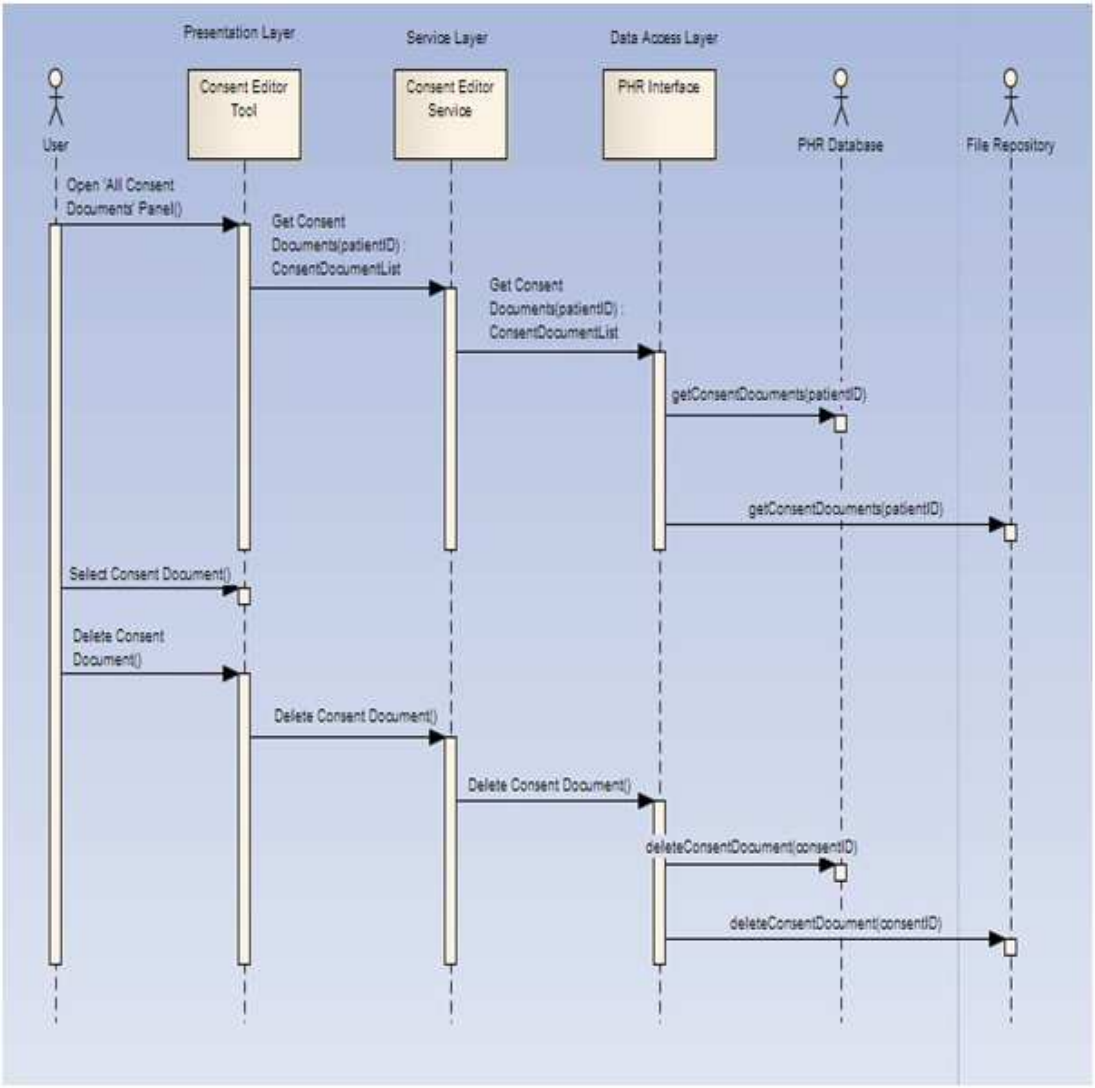


Figure 4.5: Delete Consent Document Sequence Diagram

4.1.1.2 Create/Update Consent Document

User can delete the consent documents that will never be used again.

4.1.1.3 Set Current Consent Document

Users may have several consent documents. However, only one of them is called as "Current Consent Document". When access to a document is requested, its decision determined according to this 'Current Consent Document'. In other words, 'Current Consent Document' is used as consent document by consent manager. Other consent documents are just candidates

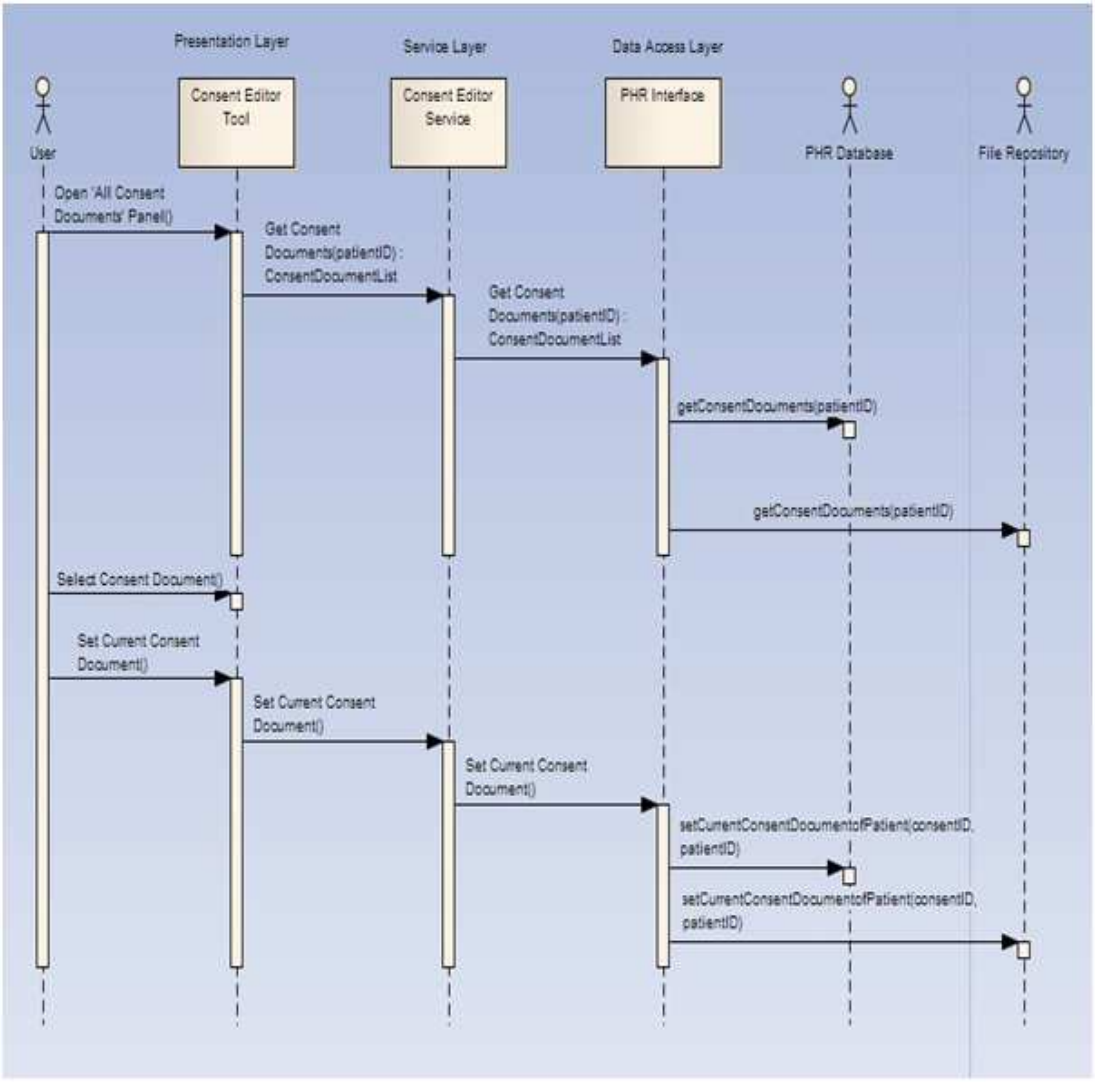


Figure 4.6: Set Current Consent Document Sequence Diagram

to be 'Current Consent Document'. Users can set their 'Current Consent Document' from the Basic Consent Editor tool.

4.1.1.4 Update Subjects

There are two types of subjects: Group and Individual. 'Group' def nes a group of people using a role based naming such as 'Nurse' or 'Family Member'. 'Individual' on the other hand, def nes a unique person by giving name, surname and role of a unique person. Each patient (or user) may want to use different subjects in their consent rules. Hence, they need to be able to def ne new subjects, whenever they want. Using Basic Consent Editor tool, users

are able to def ne new subjects for both types namely Group or Individual.

4.1.1.5 Update Resources

Users may need to def ne new resources to use while creating consent rules. New resources can be def ned from the Basic Consent Editor.

4.1.1.6 Download XACML Document

Consent rules are preserved in Consent Documents which are peculiar to Basic Consent Editor. However, Basic Consent Editor also supports XACML format. Users are able to download XACML Document of their consent documents. These documents are valid against the XACML Schema.

4.1.2 Use Cases of Consent Manager

The main point of the consent manager is evaluating the incoming request and returning an appropriate decision according to stored consent documents. The method that is performing this operation is called "getDecision" method.

4.1.2.1 Get Decision

According to different possible needs of the consent editor, "getDecision" method is implemented in several different ways. Basic Consent Editor does not provide a graphical user interface for sending requests. However, this method is provided through Consent Manager as a service which can be called in several ways.

For example, a request may contain, id of the target patient, id of the subject of access operation, id of the resource that is planned to be accessed and the action which purpose of the access operation. Request may also contain these informations hided in XML Elements. All three types (subjects, resources and actions) are modeled and stored as XML Elements in the database. Request may contain these elements instead of pure ids.

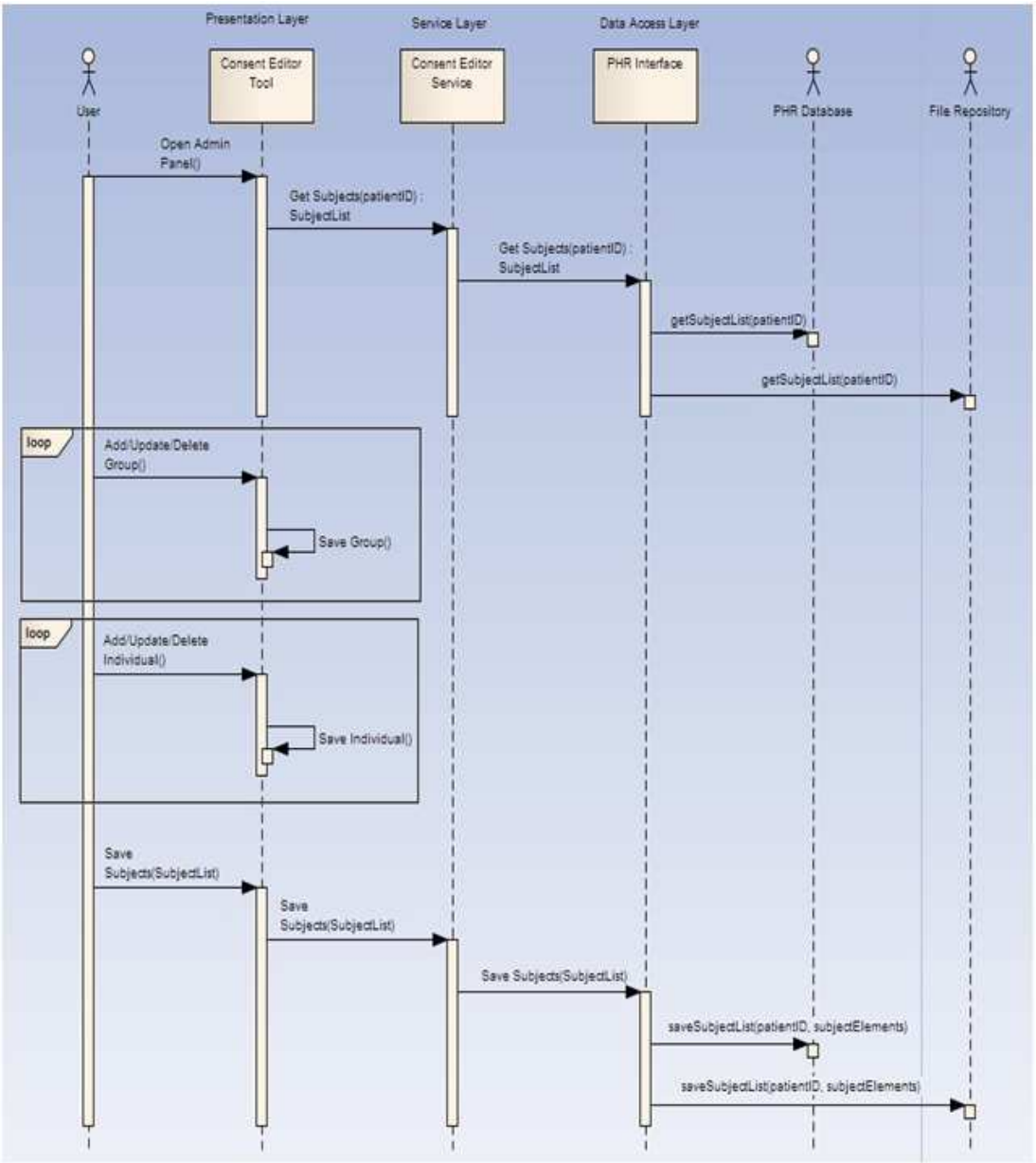


Figure 4.7: Update Subjects Sequence Diagram

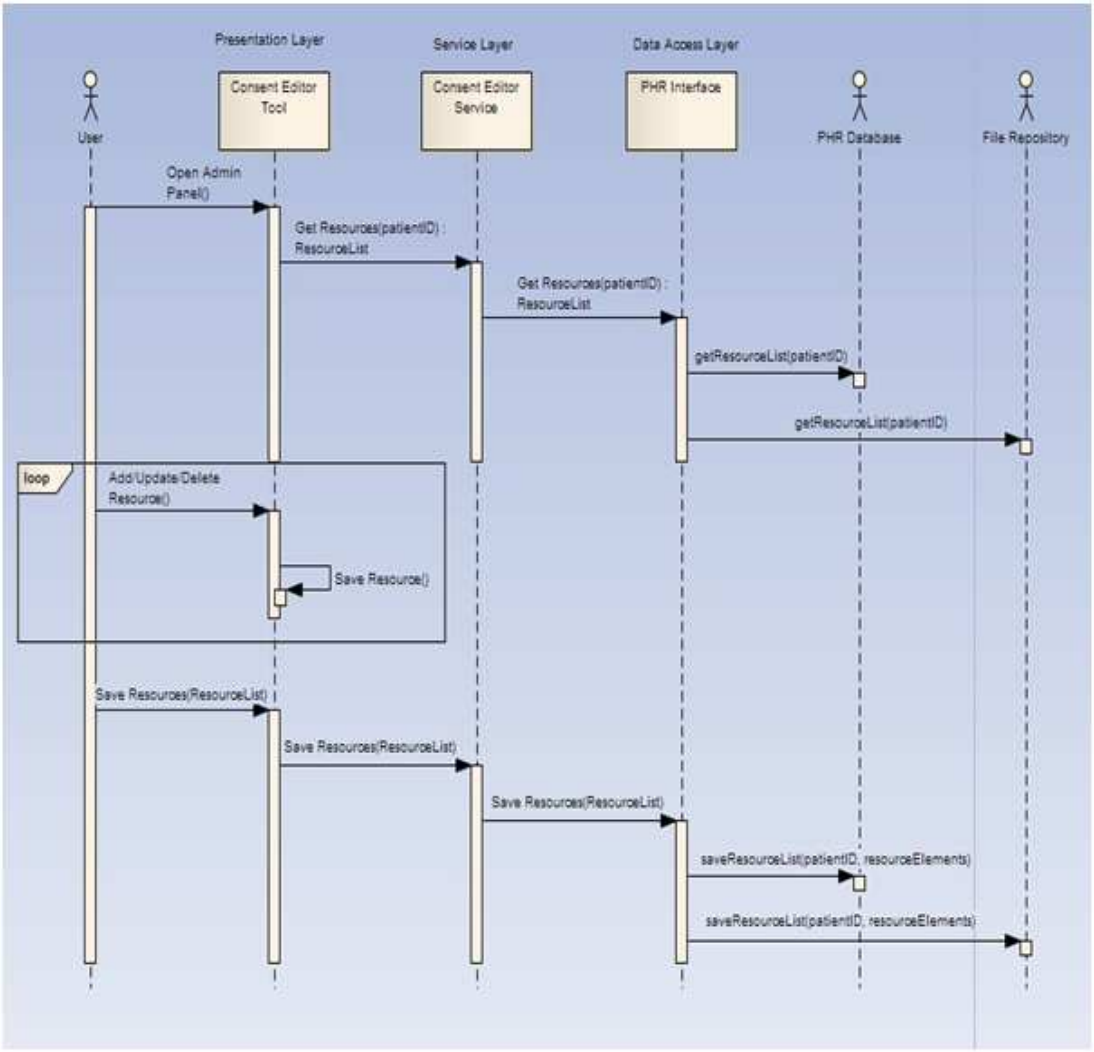


Figure 4.8: Update Resources Sequence Diagram

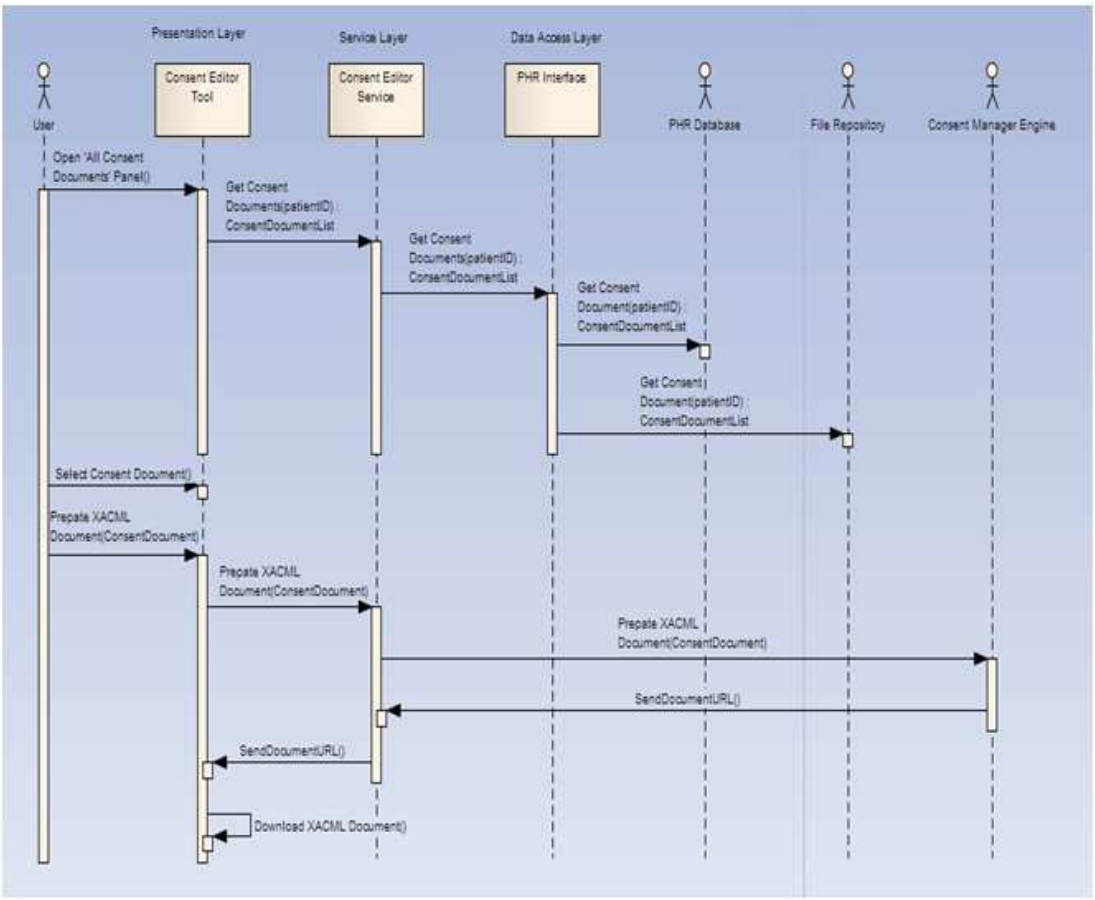


Figure 4.9: Download XACML Document Sequence Diagram

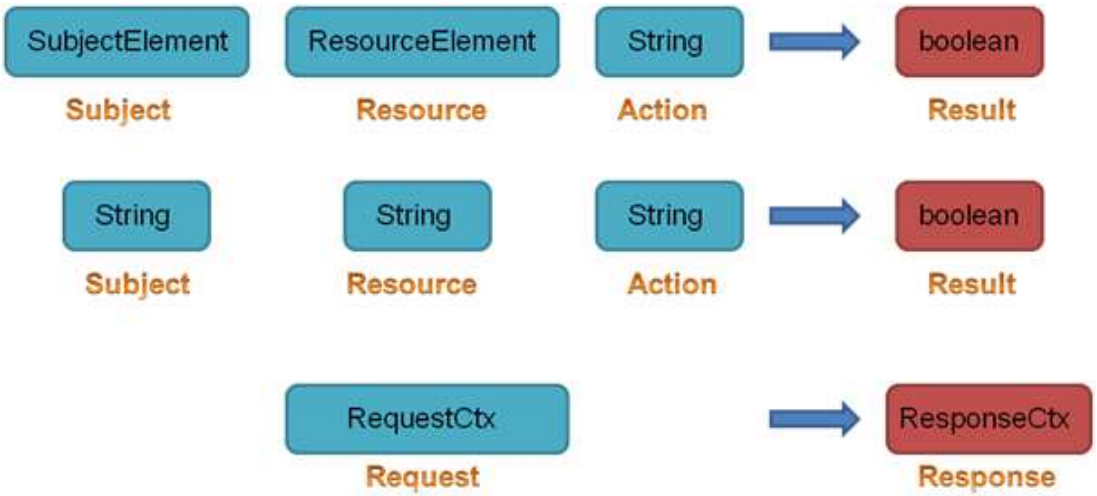


Figure 4.10: Get Decision Call Types

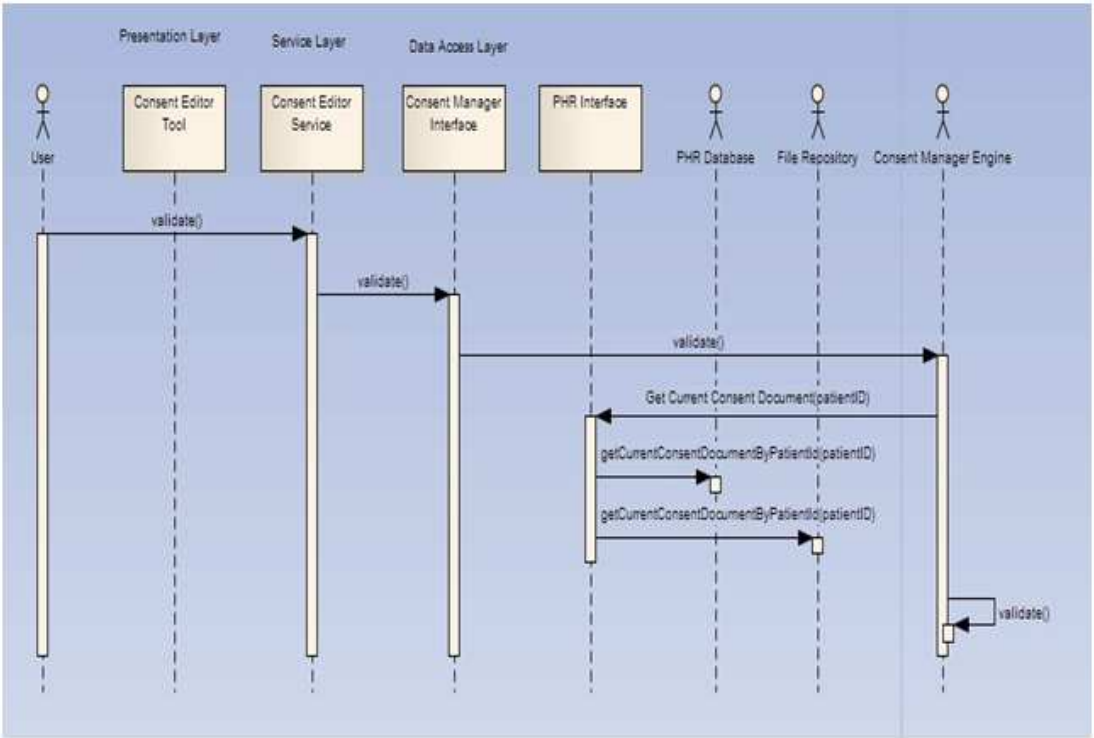


Figure 4.11: Get Decision Sequence Diagram

4.1.2.2 Consent Manager Web Service

"GetDecision" method is served to internal usage of other tools of iCARDEA system like Consent Editor. However, there is a need for serving this and some other operations externally. Hence a webservice is implemented and served with several operations including "getDecision", "getResources" and "getSubjects". This webservice implanted as Axis2 Webservice and can be deployed under any service container like Apahce Tomcat.

The operations "getResources" and "getSubjects" of this service are simple operations that just returns the list of default resources and subjects. "getDecision" operation is resembling the "getDecision" operation Consent Manager Service functionally. However, it is diffent in terms of the request and response types. "getDecision" operation of the Consent Manager Web service is receiving a XACMLAuthzDecisionQuery[15] which is specialized kind of SAML-Request according to XACML SAML Prof le. This XACMLAuthzDecisionQuery contains an XACMLRequest and it may also contain assertions. Consent Manager Web Service, evaluates this XACMLRequest using the getDecision method of Consent Manager Service, and packs the XACMLResponse into a XACMLAuthzDecisionStatement message which it will

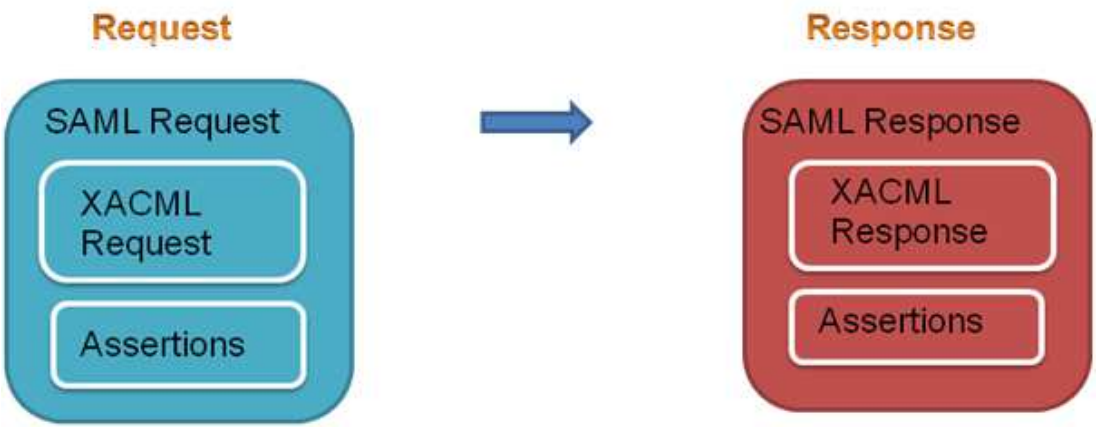


Figure 4.12: 'getDecision' Operation of Consent Manager Web Service

return as a response.

Furthermore, there is another helper operation of the Consent Manager web service called "generateRequest". This operation is generating a XACMLAuthzDecisionQuery Message according to given parameters. The parameters this operation is waiting for is listed below:

- requestId: Sets the id of the XACMLAuthzDecisionQuery message
- issuerName: Sets the name of the issuer of the message
- subjectCode: Sets the id or code of the subject in XACML Request
- resourceCode: Sets the id or code of the resource in XACML Request
- action: Sets the action in XACML Request

This method returns a ready to send XACMLAuthzDecisionQuery message when called with these parameters.

4.2 Consent Editor Implementations

4.2.1 iCardea Consent Editor

The iCARDEA system aims to automate and personalize the follow-up of cardiac arrhythmia patients with implanted Cardiovascular Implanted Electronic Devices (CIED) [14] with com-

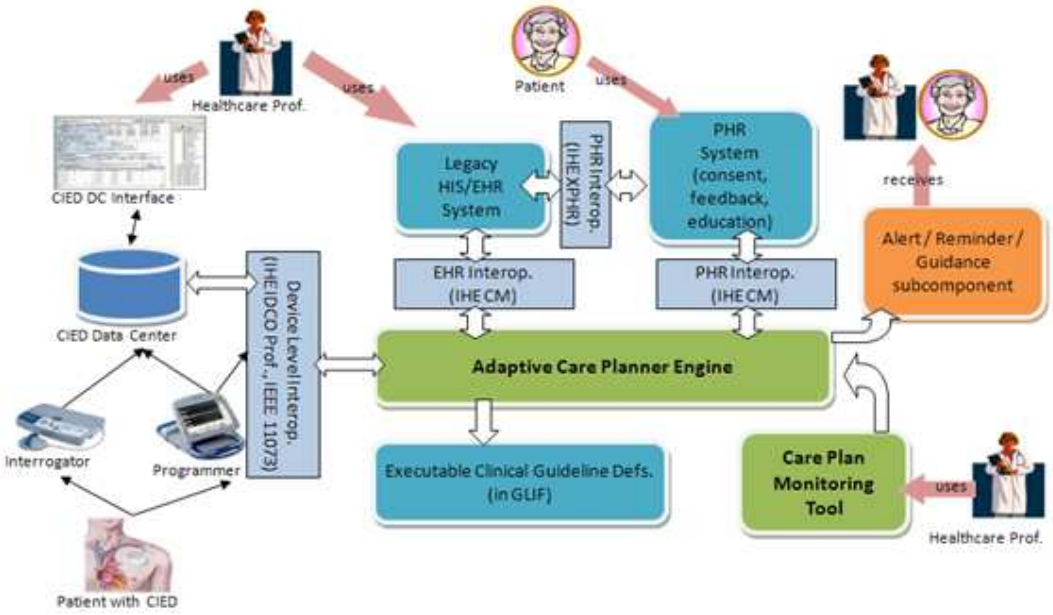


Figure 4.13: iCARDEA Architecture Overview

puter interpretable clinical guideline models using standard device interfaces and integrating patient EHRs.

iCARDEA Consent Editor is used to provide a patient centric access control mechanism for PHR users. iCARDEA Consent Editor is a compatible tool and can be easily integrated to different PHR systems through its PHR interface. Consent Editor produces XACML documents to accommodate the standards and uses those documents while generating access decision, hence PHR's are also able to use Consent Editor's decision making service just by sharing the restriction policies that created in XACML standard, without being completely integrated with Consent Editor.

iCARDEA Consent Manager is also integrated to the Care Management Database System. The Care Management Database is the central repository of the Care Plan Engine that holds the unif ed information from different data sources of iCARDEA system such as PHR, EHR and CIED Data Exposure System. The Care Plan Engine and the Patient Parameter Monitoring Tool accesses patient related data from this repository. Consent Manager is integrated to the Care Management Database system in order to ensure that the request of accessing patient related data stored in this repository should be authorized according to the consent of the patient in question.

4.2.1.1 Integration to PHR System - PHR Interface

iCARDEA Consent Editor contains a data access layer which facilitates the integration with the different PHR systems. Consent Editor does not keep any information itself. It always sends the data which needs to be stored, to PHR systems, and it receives the data again from the PHR systems when it is needed. To provide this functionality, it provides an interface which has to be implemented according to model of the PHR system. For example, 'getConsentDocuments' should take the patient id as an input parameter and should return all the consent documents that have the given patient id. If PHR system is keeping the consent documents in a MySQL database, it should run a query, find out which consent documents have given patient id, then return those consent documents. However, if data is preserved in PostgreSQL database, the queries would be different. Moreover, assume a PHR system that does not use a database, and preserves data in XML files. The implementation of the 'getConsentDocuments' method would be completely different. Below, all the methods of the PHR interface are listed:

- getPatientID
- getConsentDocuments
- getConsentDocumentByID
- registerConsentDocument
- deleteConsentDocument
- getCurrentConsentDocumentByPatientID
- setCurrentConsentDocumentByPatientID
- getSubjectList
- getResourceList
- getActionList
- saveSubjectList
- saveResourceList

Details of each method of PHR Interface and how they should be implemented are explained below.

getPatientID

This is a very simple method. It does not take any input and returns the patientID of the user that logged into PHR system. Each PHR system keeps a unique id for their patients. This method just requests this unique id to make consent editor consistent with the PHR system.

getConsentDocuments

Consent Editor provides all consent documents to its user for updating or setting as current consent document. Hence, all consent documents are needed to be listed for the user.

This method sends patient id and requests for all of the consent documents that belong to the user with patient id given as parameter. As explained above, when this method is called, PHR system will check the consent documents, make a list of the consent documents that belong to the user, and send them back to the Consent Editor.

getConsentDocumentByID

This method is a helper method to other methods. Other methods use this method to get consent document itself by its consent id.

This method takes one parameter, consent id, and returns the releated consent document.

registerConsentDocument

After creating or modifying a consent document, users need to save it for future use.Since iCARDEA Consent Editor does not keep any message itself, it sends this message to PHR system.

This method takes one parameter, consent document to be registered and registers that document into registry. While saving the consent document, it checks whether there is another

consent document with same consent id in the registry. If there is, then this method updates the consent document with new one. Otherwise, it creates a brand new consent document.

deleteConsentDocument

Users may create several consent documents in time. However one day, they may decide that any of their consent documents will not be used any more. They may want to delete these consent documents, hence these consent documents will no longer be kept at PHR system.

This method takes consent id as parameter and deletes the consent document with that consent id.

getCurrentConsentDocumentByPatientID

Users may have several consent documents, but only of them is the 'Current Consent Document'. This consent document is the one which will be used in 'getDecision' service (request validation).

This method takes one parameter, and as it is clearly indicated from the name of the method, this parameter is the patient id. This method first looks into the 'CurrentConsentDocument' table in database, and finds out the consent id of the 'CurrentConsentDocument' of given patient. Then, it returns the consent document from 'ConsentDocuments' table according to found consent id.

setCurrentConsentDocumentOfPatient

When users change their 'CurrentConsentDocument' from the user interface, it should be also noted in the database.

This method takes two parameters which are consent id and patient id. It updates consent id value in the 'CurrentConsentDocument' table according to given patient id.

getSubjectList - getResourceList - getActionList

Users don't always have the same subject list. For example a user may add her dentist as an individual to her subject list. However another user may not even know that dentist, and does not want to include that dentist in the list. Hence, different users may have different subject list. This situation is valid also for resources.

These methods get patient id as a parameter and return the related list according to this given patient id.

Although PHR providers can implement PHR Interface according to their systems, an implementation of the PHR Interface has been served for direct usage without need of any further efforts. In this ready-to-use implementation, Hibernate implementation of JPA is used for registry access operations. Using JPA for registry access operations provides ability to change the database management system at the end with minimal conf guration.

saveSubjectList - saveResourceList

Subjects and resources may be updated as explained above. Actions are not designed to be updated. They are set once and all users use same actions. Hence there is no method such as setActionList.

These methods take two parameters, f rst parameter is the patient id and the second parameter is the related list.

Implementation of PHR Interface

Although PHR providers can implement PHR Interface according to their systems, an implementation of the PHR Interface has been served for direct usage without need of any further efforts. In this ready-to-use implementation, Hibernate implementation of JPA is used for registry access operations. Using JPA for registry access operations provides ability to change the database management system at the end with minimal conf guration.

4.2.1.2 Integration to Care Management Database System

The Care Management Database is the central repository that holds the unif ed information from different data sources of iCARDEA system such as PHR, EHR and CIED Data Exposure System. In other words, this system contains all the medical information and CIED information about the patients. The Care Management Database is mainly composed of two sub modules: The relational database that holds the actual data and the API submodule that allows to reach the relational database through Java classes. The relational database of the Care Management Database is designed according to IHE IDCO Prof le data model and IHE Care Management Prof le Data Model.

As the Care Management Database System holds all the medical information about the patients, it can be regarded as the most critical part of the iCARDEA system to be secured. The consent management facility therefore enables the authorization of the requesting parties, such as the healthcare actors using the Careplan Engine or Patient Parameter Monitoring module. The main idea to integrate the consent management mechanism to Care Management Database system is that before delivering the requested content to the requestor party, the request should be authorized according to the consent of the patient in question.

To be more specif c, when the Care Management Database system gets a request:

- First, the role of the requestor is identif ed
- Then, the "getDecision" operation of Consent Manager Web Service, described in Section 5.2.1, is called.
- If the request is granted the information is returned to the requestor. Otherwise, necessary error report is delivered.

The "getDecision" operation is called with three parameters:

- The patientID of the patient in question
- The role of the requestor

- The requested resource type

In case the requested resource type is CIED information, the value to be passed for resource parameter is "CIEDREPORT". Considering the other medical information, as mentioned previously, the relational database of the Care Management Database is designed according to IHE Care Management Prof le and the following codes are used to represent the clinical information of the patients. In other words, the following codes (Table 4.1) are used to reach the clinical data in the Care Management Database:

However, when the users design their consent rules, they use more simple codes to represent the resources as presented in Resources Section of chapter 3. In other words, the resource types in the Consent Manager Web Service is different than the resources types in the Care Management Database. The following mapping in Table 4.2 is used, when accessing the Consent Manager Web service:

4.2.1.3 iCARDEA Consent Editor User Interface

Users of the iCARDEA Consent Editor need a user friendly, graphical user interface. They should be able to control (update, delete or conf gure) their consent documents from this interface. For this purpose, a graphical user interface has been developed with Adobe Flex. In the following sections, details of every interface of this GUI are explained. Below a general view of the consent document is displayed in Figure 4.14.

Consent Document Panel

When the user logins to iCARDEA Consent Editor, a panel showing the user's current consent document comes across. This panel is called Consent Document Panel, since it shows details of a consent document. A screenshot of this panel is provided in Figure 4.15.

At the top of the page, name of the tool and two logos are displayed in the header. This header is displayed in all panels. There is a 'MENU' button at the top right of page, which facilitates to go between different panels. More explanations about this button are provided in its own section. Next to the 'MENU' button there is a drop down for selecting the language. Now

Table 4.1: IHE Care Management Prof le Care Provision Codes

Information Category	**Code**	**Returns**	**Template Id**
Vital Signs	COBSCAT	All Vital Signs	Vital Sign Observation
	Any Code from the Vital Signs Table in the Vital Signs Observation	The vital sign identif ed by the code	Vital Sign Observation
Problems and Allergies	MEDCCAT	All Problem Entries	Problem Entry
	CONDLIST	All Concern Entries	Concern Entry
	PROBLIST	All Problem Concerns	Problem Concern
	INTOLIST	All Allergy Concerns	Allergy and Intolerance Concern
	RISKLIST	All Risks	Concern Entry
Diagnostic Results	LABCAT	All Lab Results	Simple Observations
	DICAT	All Imaging Results	Simple Observations
Medications	RXCAT	All Medications	Medications
	MEDLIST	All Medications	Medications
	CURMEDLIST	All Active Medications	Medications
	DISCHMEDLIST	Discharge Medications	Medications
	HISTMEDLIST	All Historical Medications	Medications
Immunizations	IMMUCAT	All Immunizations	Immunizations
Professional Services	PSVCCAT	All Professional Service Entries	Encounters

Table 4.2: Consent Manager Resource to IHE CM Prof le Resources Mapping

Code	**Resource type in Consent Manager WS**
COBSCAT	Basic Health
Any Code from the Vital Signs Table in the Vital Signs Observation	
MEDCCAT	Condition
CONDLIST	
PROBLIST	
RISKLIST	
INTOLIST	Allergy
LABCAT	Test Result
DICAT	
RXCAT	Medication
MEDLIST	
CURMEDLIST	
DISCHMEDLIST	
HISTMEDLIST	
IMMUCAT	Immunization
PSVCCAT	Encounter

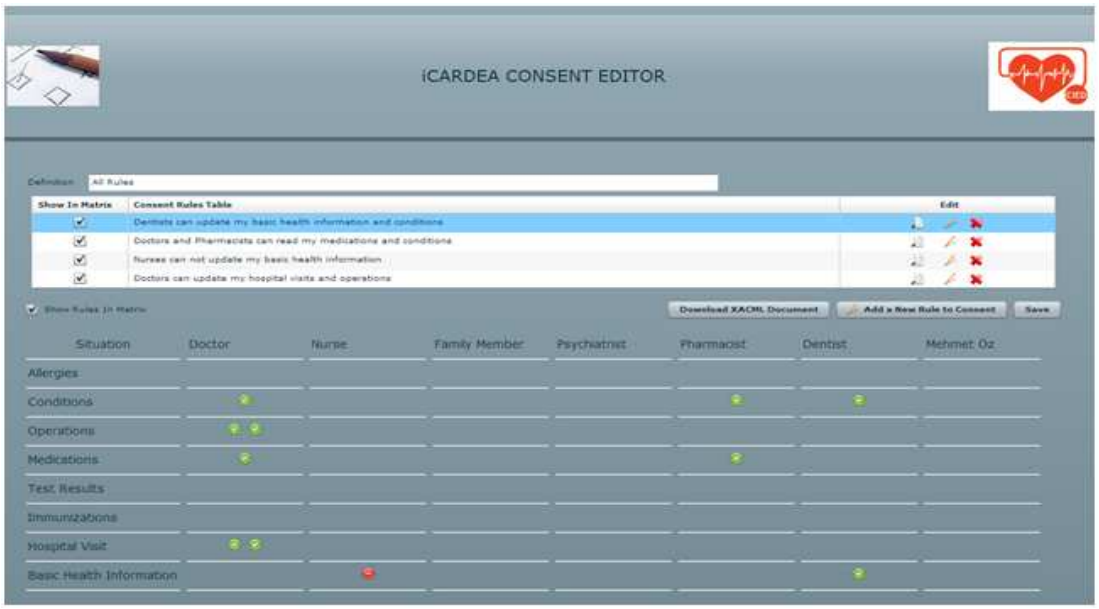


Figure 4.14: iCARDEA Consent Editor User Interface



Figure 4.15: Consent Document Panel View

only English and Turkish are available. However, this list can be extended.

In the main part of the panel, def nition and a list of consent rules are provided. Def nition is given in a text box at the top, and the consent rules are displayed as a list. User can view, edit or delete consent rules from the 'Edit' f eld of each row.

- **View:** It displays the details of the consent rule in Consent Rule Panel. However, updates cannot be saved in view mode.
- **Edit:** It displays the details of the consent rule in Consent Rule Panel, but this time, updates can be saved.
- **Delete:** It deletes the consent rule from the consent document.

Users may also add new consent rules to the consent document. To do that, user can press the "Add a New Rule to Consent" button at lower right of the panel. This will open Consent Rule Panel with empty values.

After making changes to consent documents, user needs to save the consent document using the 'Save' button at lower right of the panel. Consent documents can be updated or save as a new consent document.

Users are also able to download XACML document of their consent documents. This can be

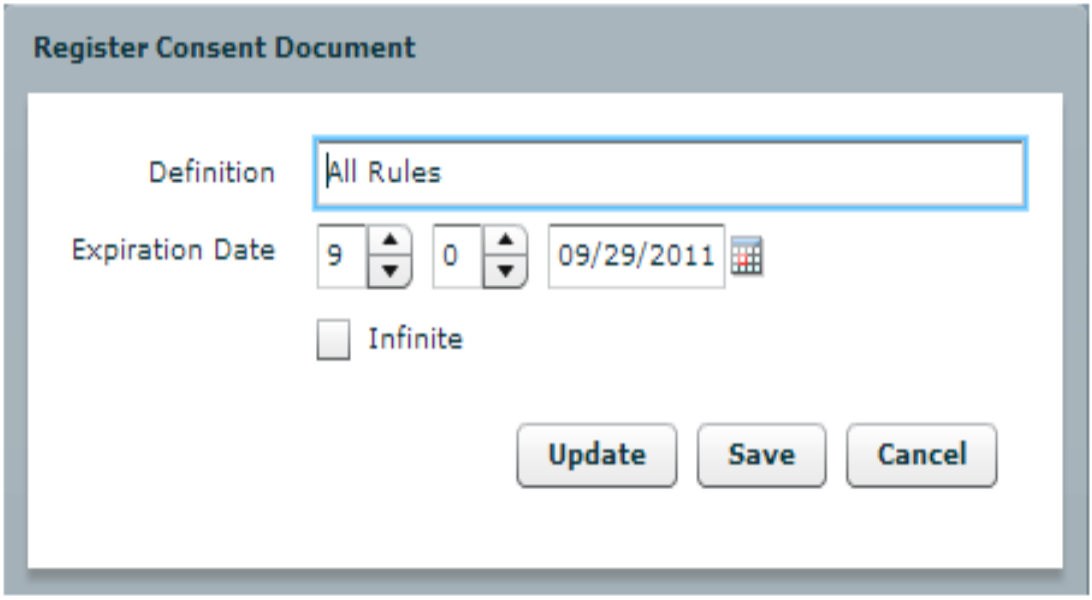


Figure 4.16: Register Consent Document View

done from Consent Document Panel by pressing the "Download XACML Document" button or from "All Consent Documents" Panel as explained later in this deliverable.

If the descriptions of the consent rules are not clear enough, user may not understand this consent document permits or denies which accesses. To see the general view of consent document in terms of permissions, user can check the "Show Rules In Matrix" checkbox. This will display the details of the consent document in a matrix like view.

Consent Rule Panel

Consent Rule Panel displays the details of a consent rule, enables modifying and saving it. A screenshot of the Consent Rule Panel is displayed in Figure 4.18.

Consent Rule Panel displays the description, effect and the target of the consent rule. Description is provided in a text box at the top of the panel. Effect which may take values "Permit" or "Deny" can be specif ed by the radio buttons at right of the description.

Target of the consent rule contains three dimensions, which are Subjects, Resources and Actions. User can select one or more value from each dimension of the target. For example, consent rule in Figure 4.15 means that "**Doctors** and **Pharmacists** (which are selected sub-

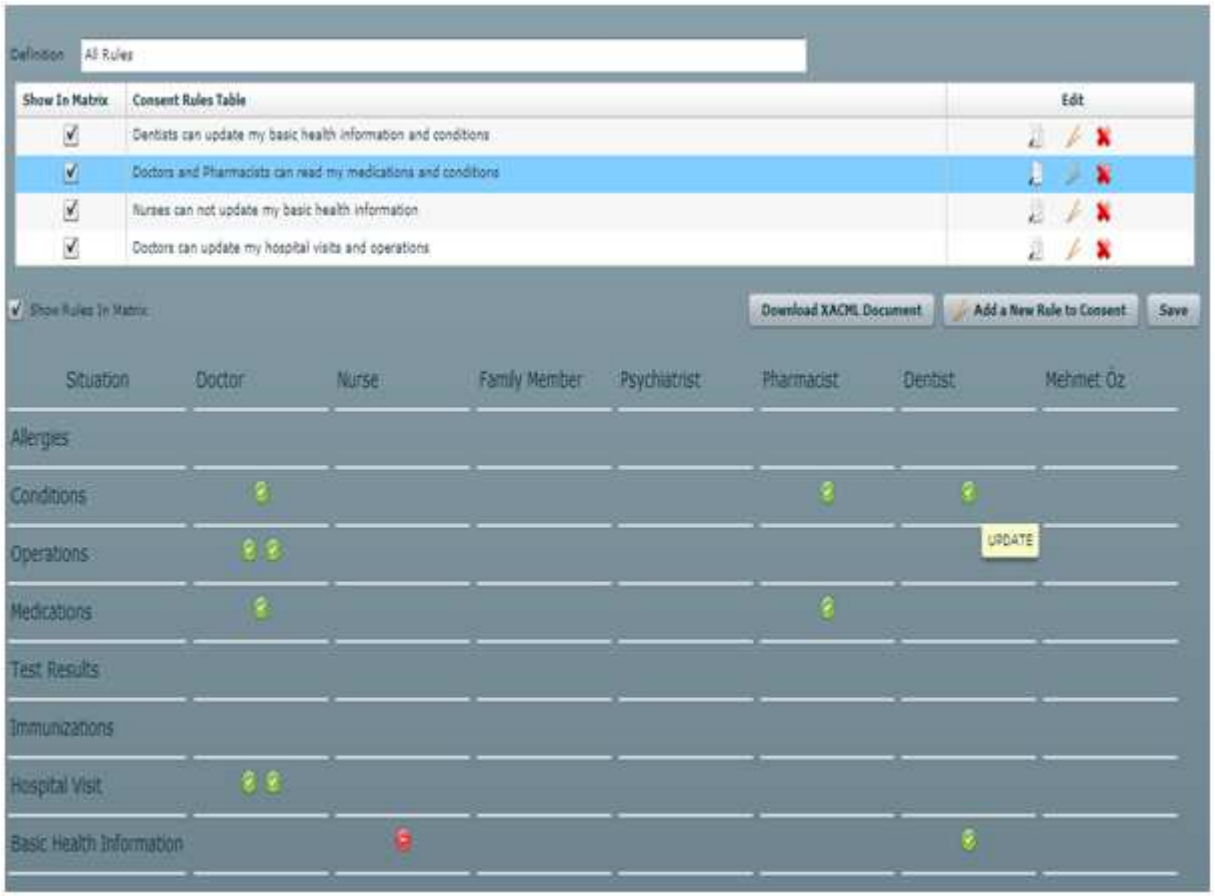


Figure 4.17: Matrix View of Consent Document

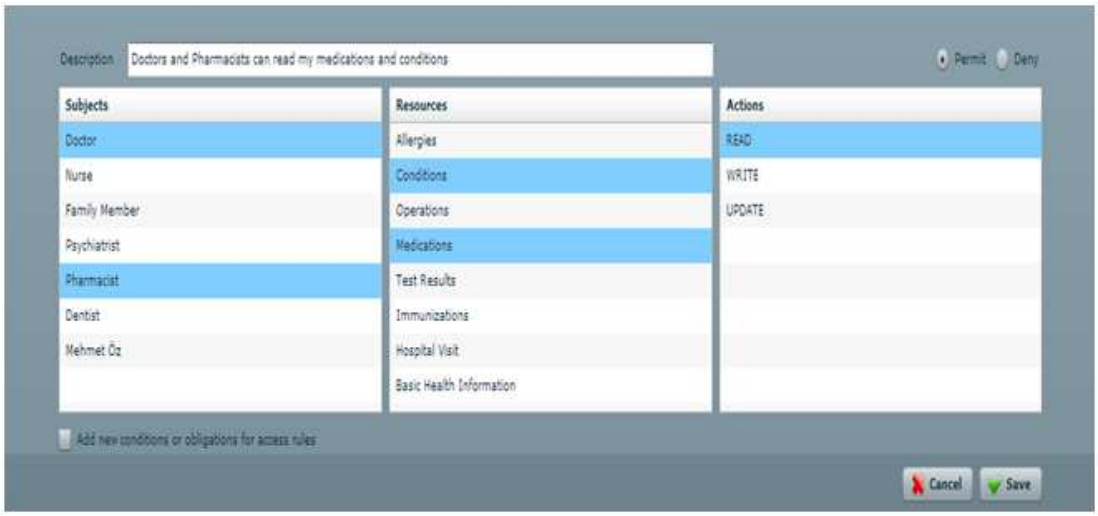


Figure 4.18: Consent Rule Panel View

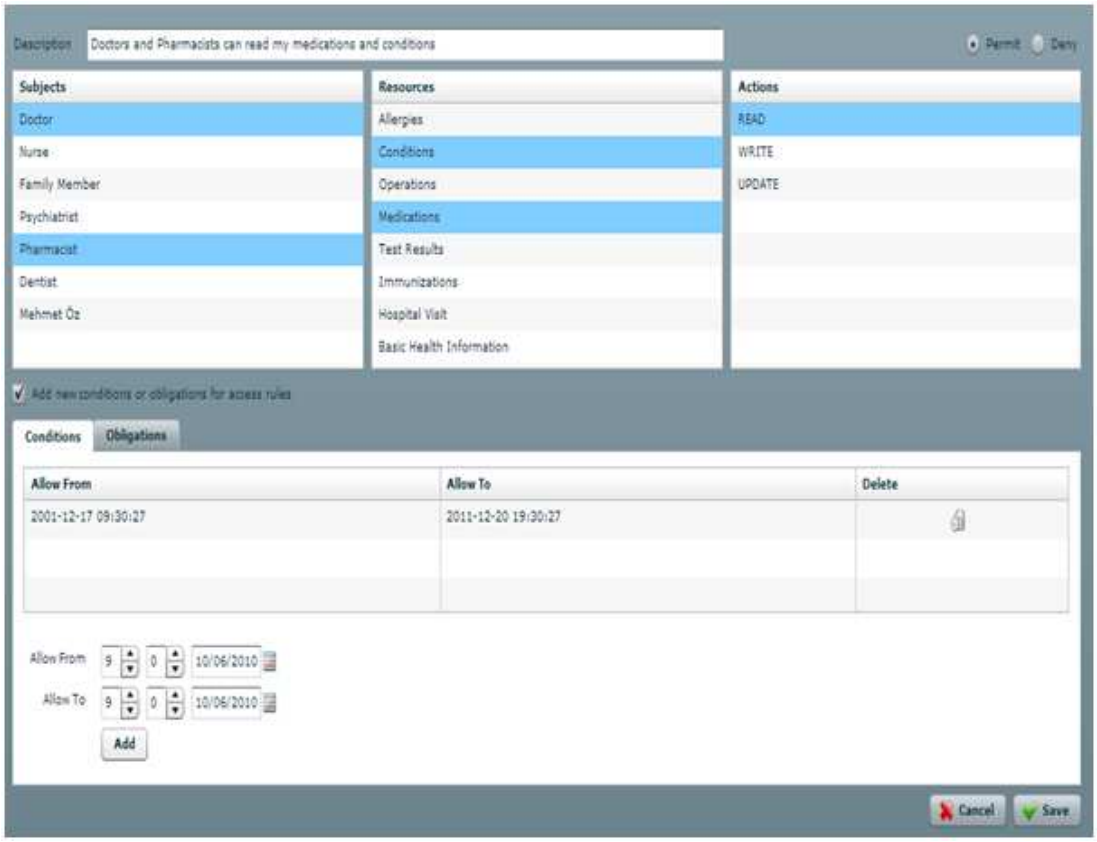


Figure 4.19: Conditions in Consent Rule Panel View

jects) **can** (selected effect) **read** (selected action) my **Conditions** and **Operations** (selected resources)". Description is clearly stating this situation.

Sometimes, target is not enough to def ne a new rule. Users may want to add some conditions such as "Doctors can read my conditions **only in weekdays**". To add such a condition, iCARDEA Consent Editor provides adding a condition into a consent rule. To do this, user has to check the check box "Add new conditions or obligations for access rules" at lower left of the panel. A screenshot of the Condition part of the Consent Rule Panel is displayed in Figure 4.19.

User can add conditions and obligations from that part of the panel. Adding conditions spec-if es a time limit for the access permission, and adding obligation enforces system to send a mail whenever a record of the user has been accessed.

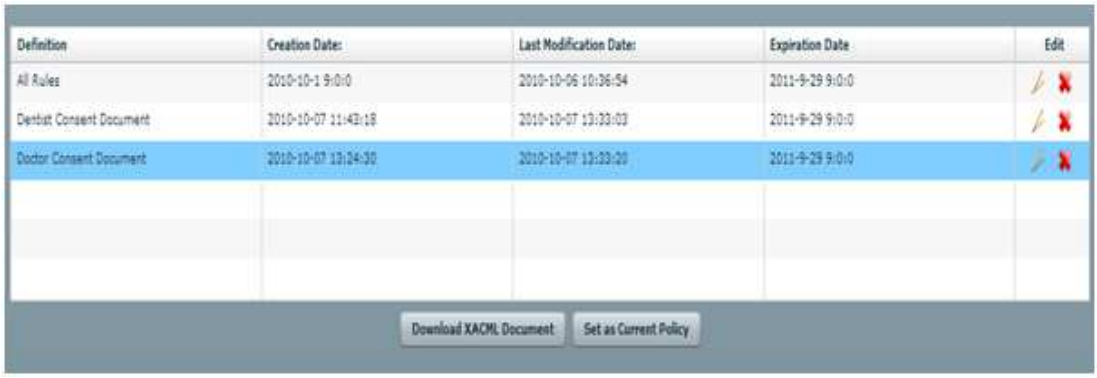


Figure 4.20: All Consent Documents Panel View

All Consent Documents Panel

This panel lists all the consent documents that created by the patient.

User can see the definition and creation/modification/expiration dates of the consent documents. From the Edit field in each row, users are able to edit or delete consent documents. When a user clicked the edit button, Consent Document Panel that showing the selected consent document is opened. User can set any of these consent documents as his Current Policy (Current Consent Document). When a consent document is set as Current Consent Document, validation is done according to that document anymore. As mentioned at the Consent Document Panel section, users are also able to download XACML versions of their consent documents from this panel.

Admin Panel

Different users may have different subjects or resources list. For example, users may want to specify a rule for a new group or an individual. In this situation, user has to extend her subject list. However, new subjects added to user's subject list may not be used by other patients. Hence users may different subject lists. To prevent enforcing users to create all subjects from scratch, a default subject list is provided. Users can use default subject list if it satisfies their needs. Resources are on the same conditions with subjects. Users can update resources whenever they need.

Admin panel is providing a user interface for modifying subjects and resources lists. Users

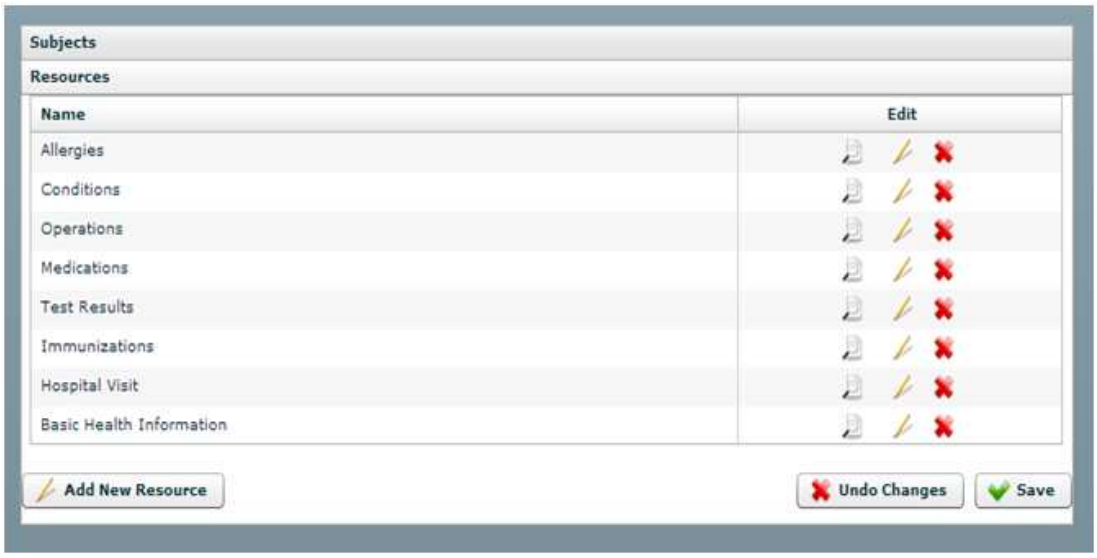


Figure 4.21: Admin Panel View

may add, delete or update the properties of every subject and resources from this panel. A screenshot of the Admin panel is provided in Figure 4.21.

Another screenshot is displayed in Figure 4.22, which is showing the "Edit Individual" Operation from Admin Panel.

Menu

Menu is a shortcut for moving in different interfaces(Figure 4.23).

There are 4 options under the menu button. **Create a New Consent Document:** Opens 'Consent Document Panel' with an empty template. **View Current Consent Document:** Opens 'Consent Document Panel' with Current Consent Document in it. **My Consent Documents:** Opens 'All Consent Documents Panel'. **Admin Panel:** Opens 'Admin Panel'.

4.2.2 eSağlıkKaydım Consent Editor

eSağlıkKaydım is a personal health record system developed by SRDC. It contains several dynamic modules such as disease management appications, decision support systems and also other applications to allow patients to access their health information. These information are

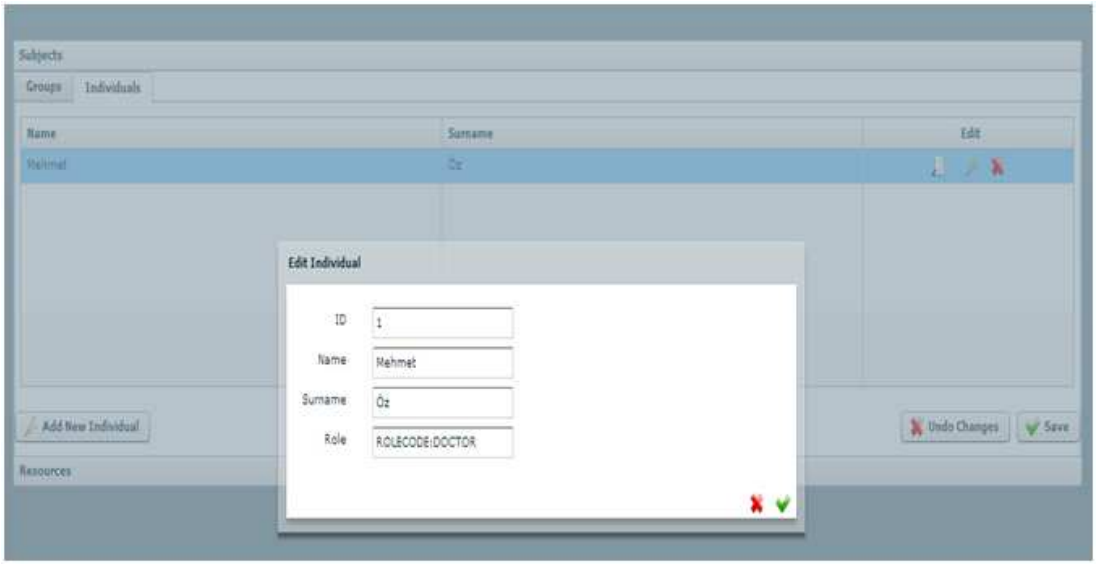


Figure 4.22: Admin Panel Edit Individual View



Figure 4.23: Menu View

collected from several data sources such as National Health Information System and Family Medicine Information System. These are national EHR systems of Turkey which are maintained by Ministry of Health. The main aim of the eSağlıkKaydım portal is enabling patients to access their records. Moreover, it also supporting with several decision support systems to help patients while managing their chronic diseases.

Patients are not the only users of the eSağlıkKaydım system. Healthcare professionals (HPs) are also using the system to monitor their patients. HPs can also interact with the patient through eSağlıkKaydım system. They can add records to the patients, create care plans or adjust the medications. Enabling access and update of patients information by their HP upon approval of the patient is favourable, however unauthorized accesses may create great problems in respect to privacy of patients information[4]. Hence, developing a powerful and modular consent mechanism was unavoidable for managing information access within eSağlıkKaydım system.

Basic Consent Editor is designed modular and easily adaptable to other PHR systems. It can be ready to use after implementing the provided interface. However, the interface developed for interoperability with other PHR systems only provides basic functionality. eSağlıkKaydım is a comprehensive system which has additional requirement for a consent management system. To cover these additional requirements, the provided interface is extended with new methods:

- **getDisallowedResource:**
- **updateDisallowedResource**

eSağlıkKaydım system has an extensive security level with respect to Basic Consent Editor. It allows setting consent documents on specif c resourse instances. For example, a patient who has caught AIDS disease may want to share all his previous diseases with his general practitioner (GP) but this. In Basic Consent Editor patients are only able to give access permission to only a type of resource. Rule can say "GP can/cannot read my diseases." without specifying any disease. To cover such a functionality a list of records that has not been seen anyone are stored in a seperate table. These records are called "disallowed resources". The additional methods are for getting list of disallowed resources and updating the list of disallowed resources.

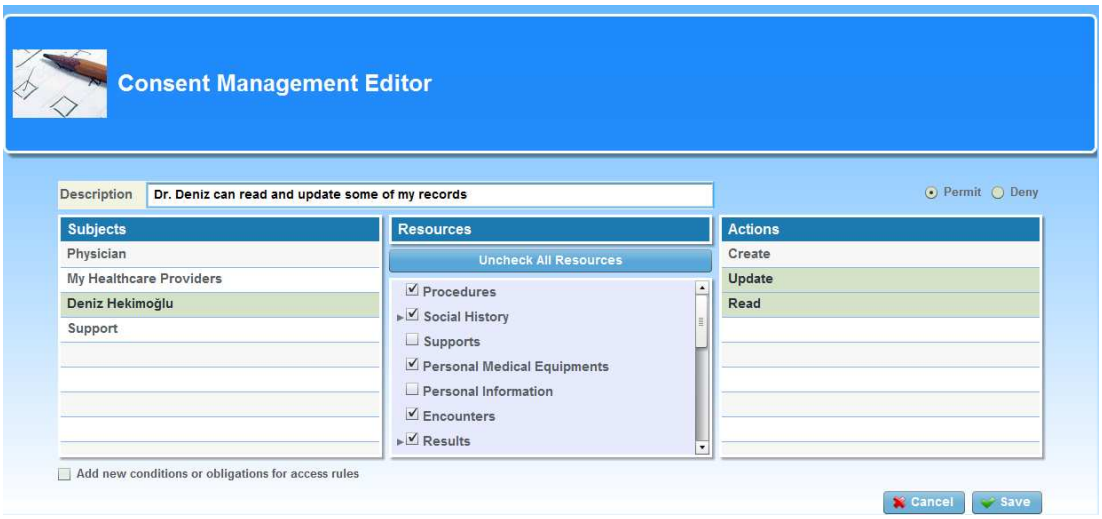


Figure 4.24: epSOS Consent Rule

Another feature of eSağlıkKaydım Consent Editor is creating consent documents for hierarchical resources. Resources may have related with other resources, such as problems and episodes. A patient may have several problems and these problems may have several episodes. In most of the diseases, episode information is as important as the problem itself. Hence, in EHR systems episode information are also stored as seperate records. Basic Consent Editor is already providing support for consent documents with episodes as resources. In eSağlıkKaydım Consent Editor, the hierarchical relation between problems and episodes are concidered and whenever, a rule is created for a parent resource type such as problem, the rule applies to all the subresource types of that resource. The hierarchical rules are displayed in 4.26.

eSağlıkKaydım Consent Document View has also a matrix view, however it is slightly different. It is displaying the action without hovering on them, and it also shows hierarchical resources since they are included in eSağlıkKaydım Consent Editor. This view is displayed in Figure 4.25.

Consent Manager of the eSağlıkKaydım is integrated into PHR system as a bundle service. Since eSağlıkKaydım is developed with OSGI[5] technology, integrating a new bundle into the system is easily applicable. Consent Manager provided it's decision point as a service, and Authentication Manager of the eSağlıkKaydım has used it while managing access operations.

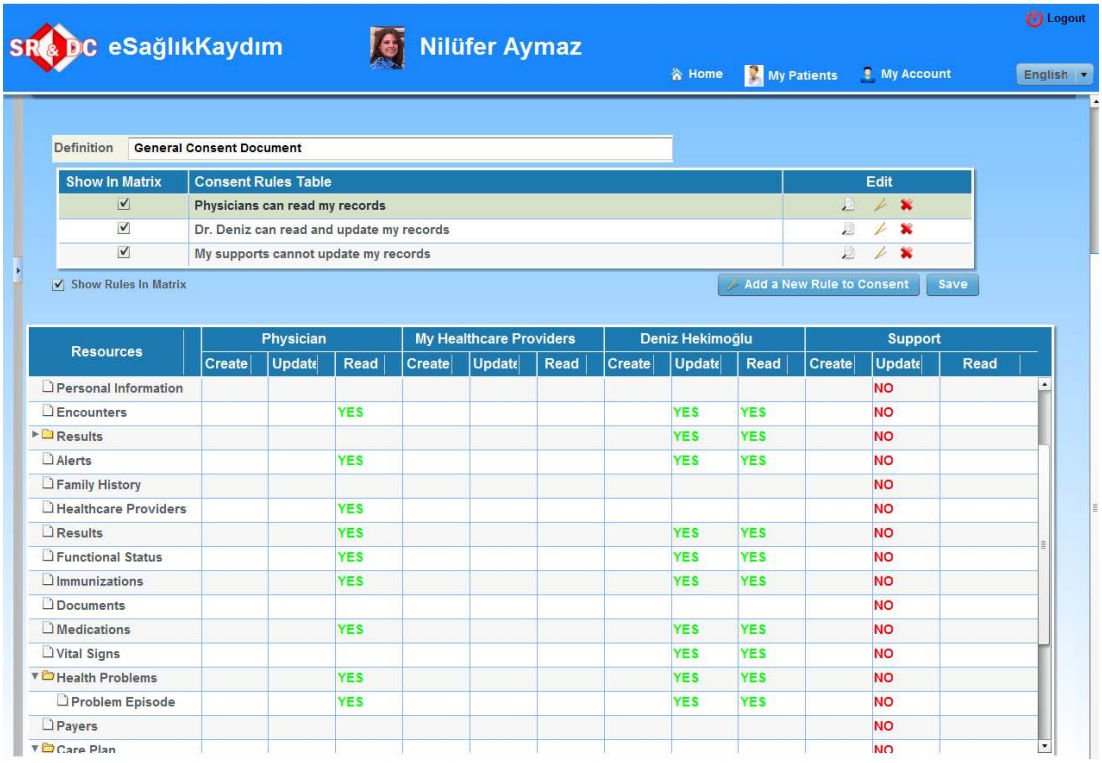


Figure 4.25: eSağlıkKaydım Consent Document Matrix View

4.2.3 epSOS Consent Editor

epSOS is an ICT-PSP Project supported by European Commission under 7th Framework Programme. It aims enabling access to patient summary documents by the healthcare professionals when a patient visited a healthcare facility in a foreign country within Europe. Project has more than 40 partners from different 23 countries. SRDC and Ministry of Health Turkey is the partners from Turkey. Project is also covering the sharing of ePrescription and eDispensation documents, however some of the countries such as Turkey are not joining this part since using common ePrescription is only available for European Union members.

A classic scenario for epSOS usage is an Portuguese patient visiting Turkey for touristic reasons. During his vacation, he has some problems with his stomach and visits a healthcare provider which is epSOS supported. Both patient and healthcare professional (HP) knows basic level English as common language. Before HP starts treatment he need to know some basic information about the patient. Here comes epSOS to solve this problem. The patients gives his identity card to the HP. HP opens epSOS Portal, selects the patient's country, enters

patient's identif cation number and starts search. A result arrives within seconds which shows name, surname and birthdate of the patient. HP checks this information with the identif cation card provided, and if the information is true, he conf rms the "Treatment Relationship Conf r-mation". This means, "this patient is right here with me, and I am accessing his records upon his request." After conf rming the Treatment Relationship, HP queries patient summary documents of the patient. There exist two versions of the patient summary. Former is the original patient summary document, generally scanned copy of the patient summary if the country is providing patient summaries. This version is translated as a PDF document. Some of the countries including Turkey does not preserve any kind of information such as Patient Summary. So the information of the patient has been collected from national Electronic Health Record Systems, and a Patient Summary is generated for epSOS project. Latter version of the patient summary has same content with the former, however it is transformed into a machnine compatible version of the scanned copy. It is preserved as an XML and during its transfer from the patient's home country to the point of treatment, its content is transformed into common code systems specif ed by epSOS and the language of the document is translated into HP's language. This transformation and translation occurs in this way: Every participating nation(PN) of the epSOS has provided mappings of their national code systems with the common epSOS code systems. Moreover, all PNs are also have provided translations of the codes in epSOS Code System from their language to English and vice versa. When a patient summary is requested, the country providing the patient summary transforms the document so all the codes in the document is replaced with their mappings in epSOS code system. In this step codes are also be translated to English since all the codes in epSOS code system are preserved as English. When the requestor country takes the document, it translates it from English to their language and HP can access and easily understand the document.

While providing security of patients' information in a country is a serious problem, sharing documents through international level needs special attention. epSOS Consent Editor enables patients to share their patient summaries within a time and location limit. After conf rming to use the epSOS system, patients selects countries to which their information is shared. epSOS Project is useful for only patients that are going out their country. Hence they do not need to share their information as long as they do not go abroad. When a patient planned a trip to a foreign country, patient enable access of her information within the time limit of the travel, and only from the country that the trip is planned to be.

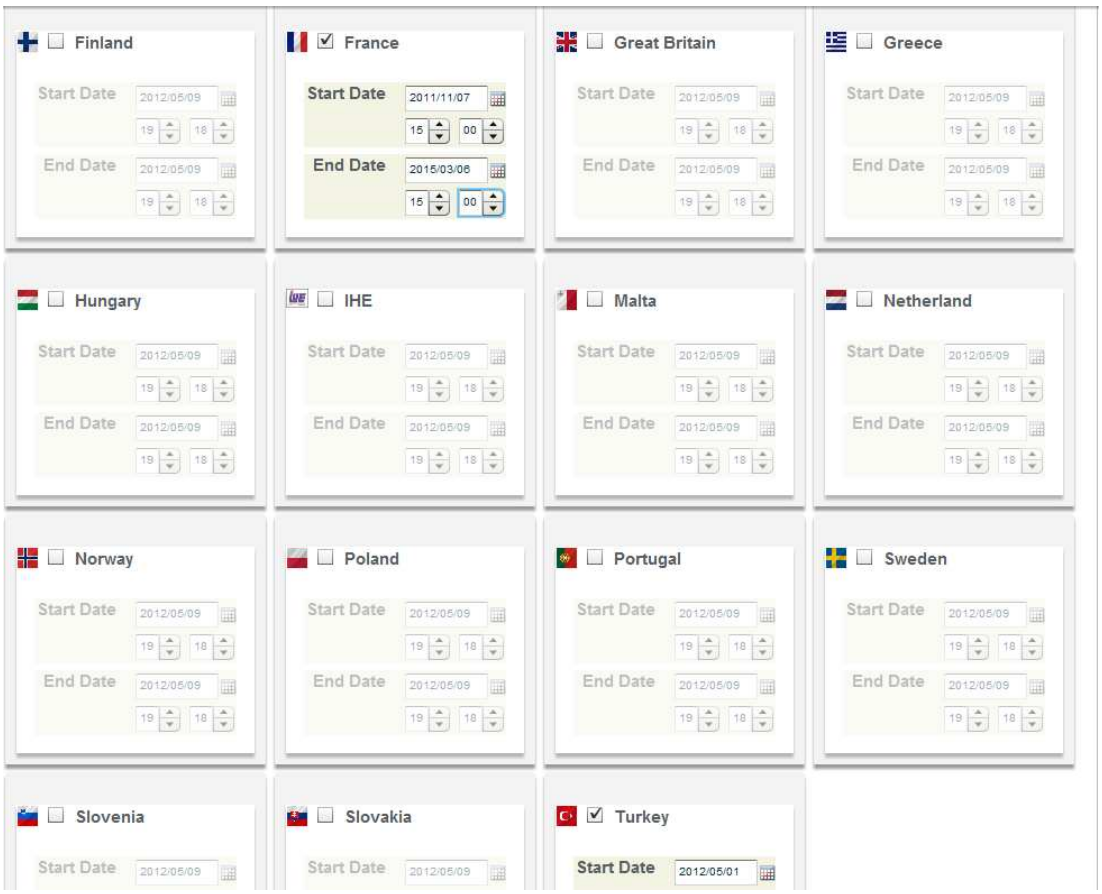


Figure 4.26: epSOS Consent Editor

In f gure 4.26 interface of the epSOS Consent Editor is displayed.

CHAPTER 5

RELATED WORK

There is not much consent management systems in commercial sector, however there are some ongoing eHealth initatives from several countries which has long term strategies for developing country wide electronic consent mechanism on top of still country wide electronic health record systems. Leadership in this f eld is being holded from the countries such as USA, United Kingdom, Australia and New Zealand.[11]

Beside the countries' national iniatives, there is not any widely accepted implementation of a consent management system, even though the patient based consent management systems are trending for several years. Still there are a number of research, development and standardization initiatives in this area.

5.1 Principled Electronic Consent Management

One example is the paper describing a preliminary research framework called Principled Electronic Consent Management[9]. In this research, electronic consent decision making systems are categorized into three levels which are First Generation Electronic Consent Management(ECM), Ex-post ECM and Principal ECM. Authors support the principled ECM and outline a research framework proposing three key components: Consent Theory, ECM norms, and ECM norms's manifestation.

5.1.1 First Generation ECM

This f rst generation ECM pushed to the service providers by laws and regulations. It involves three activities which are the disclosure of information, signaling of the consent decision and enforcement of consent decisions.

First generation information disclosure are consist of privacy policies, End User License Agreements and Platform for Privacy Preferences. They are categorised in terms of their role such as Format-biased and Process-biased.

First generation signaling includes several mechanisms such as:

- Opt-in and opt-out tick box mechanisms.
- Shrink-wrap, browse-wrap and click-wrap agreement techniques.
- P3P technologies that enable automatic consent signaling based on comparisons between privacy policies and end user preferences.
- Less tangible activities, such as browsing through a website.

First generation enforcement mehcanisms includes data protection legislation, information security standards and privacy enhancing technologies.

Shortcomings of First Generation ECM

Even though First Generation ECM has a constructive starting point, it is not able take care of some important problems. First of all it is cannot earn the end users' trust in online privacy and online security practices. Moreover service providers' are not able to successfully authenticate the consenting parties all the time.

5.1.2 Ex-post ECM

Ex-post ECM is aiming to address the shortcomings of First Generation ECM. It also uses signaling, enforcement, authentication and storage mechanisms just as First Generation ECM,

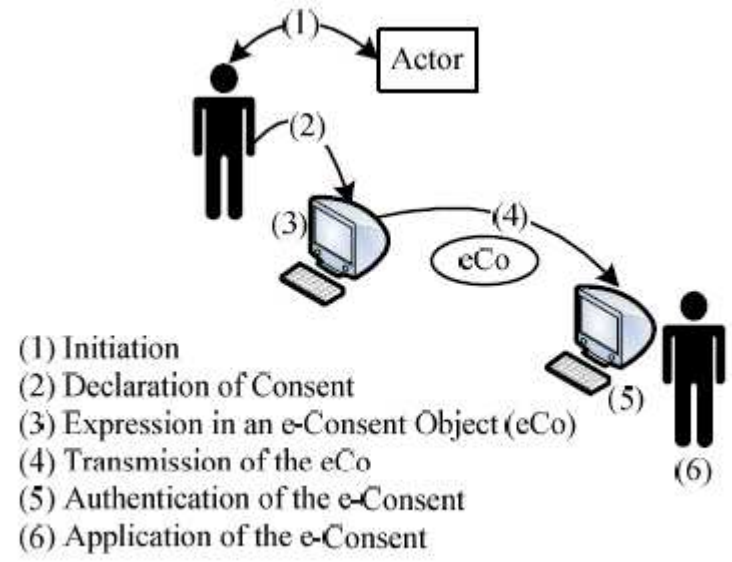


Figure 5.1: The eConsent Process

not only for communication exchanges but also for declaration, expression, transmission, authentication and application of consent decisions.

Shortcomings of First Ex-post ECM

Ex-post ECM is providing a better security, and safer consent decision signaling. However, it only focuses on the steps after signaling the consent documents. Sadly, the steps before signaling consent document are more important when it comes to reliableness of consent decisions.

5.1.3 Principaled ECM

There is need for both theorical analysis of consent and it's practical implementation. Principled ECM clarif es this difference and it's necessity. It focuses on both before and after the consent decision taken process. Hence it is a step forward compared to First Generation and Ex-post ECM.

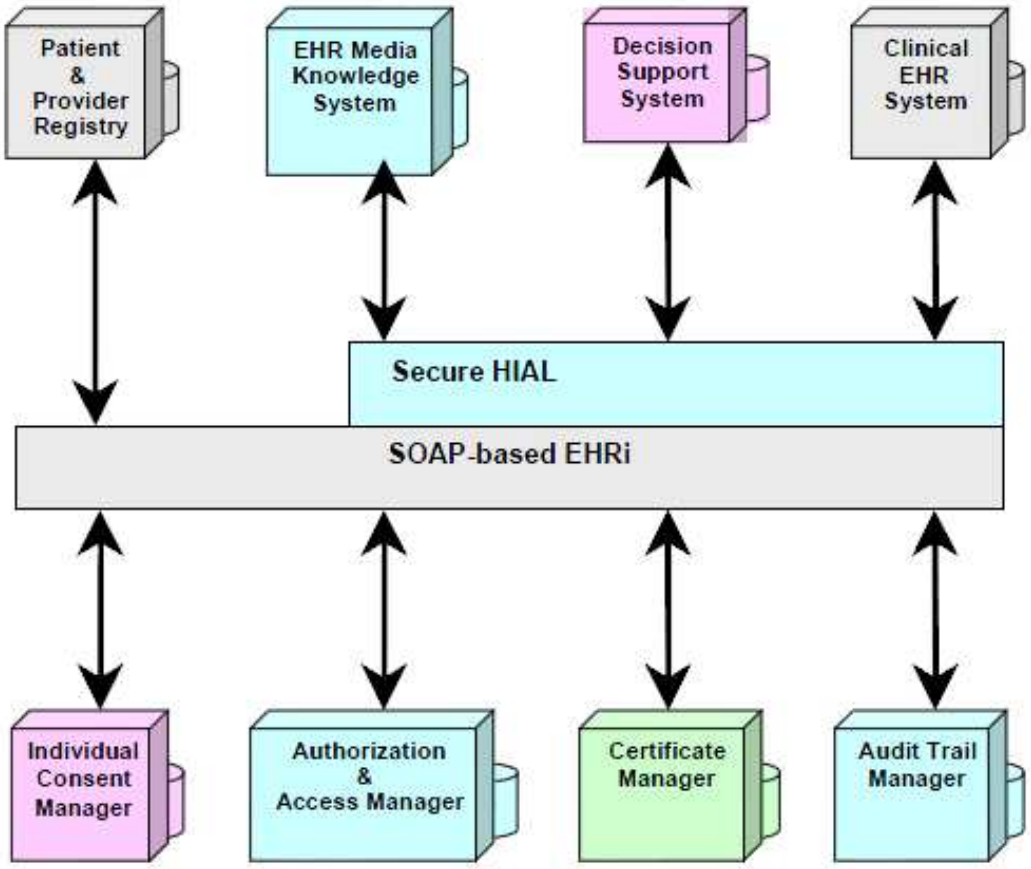


Figure 5.2: Conceptual Architecture

5.2 Privacy and Security Shield for Health Information Systems

Second example of consent management research is another paper with title Privacy and Security Shield for Health Information Systems[20]. It is stating that integration of enterprise level eHealth applications under a secure and safe context in eHealth domain definitely improves quality of the care and reduces the cost. However, there are important challenges which needs to be taken care off such as providing security of the sensitive patient information across different organizations.

One of the major parts of the system is Individual Consent Manager(ICM) which maintains information about patients' consent which specifies the use of their personal health information. It has both Web Service Interface to be used by other systems and has an user interface for patients to set their consent documents just like Basic Consent Editor.

ICM mainly provides desision to requests such as "user X applies to do function Y on data Z" which correlates with subject, resource, action terminology in Basic Consent Editor. X in this context might be a real person or a system. Y might be the an action such as display or copy. and Z is a resource such as patient’s medical records.

CHAPTER 6

CONCLUSION

Today the importance of patient empowerment is more important than any time before. The old style that patient are not included in his healthcare and most of the time has no access to his/her information is not acceptable anymore. European Comission is one of the leading supporter of the patient empowerment concept and there has been wide studies in this respect. However, there are some consequences of the empowering patient and making patients the owner of their health information. Now patients has rights to access and share their information whenever and whenever they want. This is a great challenge when the safety and security issues of accessing patient information is considered.

Basic Consent Editor is providing a general baseline for developing improved and comprehensed consent manager system. It covers most of the basic requirements of consent management system, and provides easily integratable interfaces to be used direclty within any Personal Health Record System.

Basic Consent Editor seperates the consent policy creation and access control into two different tools. Consent Editor Interface provides a user friendly graphical user interface. It enables creating consent policies which speicif es the access rules to be used by Consent Manager tool. Consent Manager is a seperate tool which is used to create decisions for applying requests according to created consent documents.

The contributions of the thesis is as follows:

- A new model based on OASIS XACML Policy Model is designed for providing more comprehensive and managable rules. This rule is basically similar to XACML Policy Model, however despite its simpler structure, it can cover more situation compared to

XACML Policy Model. This model is easily convertable to the XACML Policy Model and vice versa.

- A common interface is developed with considering needs of different PHR systems. This interface is making the Basic Consent Editor pluggable to any PHR system. Even though the interface covers most of the basic requirement, it can be extended when it does not f ts with the needs of PHR systems.

- Three different Consent Editor/Manager are implemented within the scope of thesis, two of which used the provided common interface. Beside providing proof of concept to designed consent editor, they are also giving perfect examples of how the Basic Consent Editor can be extended for different purposes. These implementations have each their own Consent Editor with user friendly GUIs.

- Global standards such as XACML and SAML are analysed and possible extensions are indicated.

As a future work the new consent model will be improved with requirements of new PHR system which are themselves improving their systems and expect new functionalities from consent management systems. Moreover, with the publication of new version of the XACML which 3.0, both consent model and Basic Consent Editor tool would be updated with new capabilities of XACML 3.0.

REFERENCES

[1] Integrating the healthcare enterprise. http://www.ihe.net/. [Online]. [Accessed: May. 12, 2012].

[2] A brief introduction to xacml. http://www.oasis-open.org/committees/download.php/2713, 2003. [Online]. [Accessed: May. 12, 2012].

[3] Sun's xacml implementation. http://sunxacml.sourceforge.net/, June 2006. [Online]. [Accessed: May. 12, 2012].

[4] Health care's most wired online. *H&HN: Hospitals & Health Networks*, 81(8):68, 2007.

[5] O. Alliance. *Osgi Service Platform, Release 3*. IOS Press, Inc., 2003.

[6] A. H. Anderson. A comparison of two privacy policy languages: Epal and xacml. In *Proceedings of the 3rd ACM workshop on Secure web services*, SWS '06, pages 53–60, New York, NY, USA, 2006. ACM.

[7] H. L. Anne Anderson. Saml 2.0 prof le of xacml v2.0. http://docs.oasis-open.org/xacml/2.0/SAML-PROFILE/access_control-xacml-2.0-saml-prof le-spec-os.html. [Online]. [Accessed: May. 12, 2012].

[8] E. Bertino and K. Takahashi. *Identity Management : Concepts, Technologies, and Systems.* Information security and privacy series. Artech House, Inc, 2010.

[9] C. J. Bonnici and L. Coles-Kemp. Principled electronic consent management: A preliminary research framework. In *Proceedings of the 2010 International Conference on Emerging Security Technologies*, EST '10, pages 119–123, Washington, DC, USA, 2010. IEEE Computer Society.

[10] R. L. Costello. An introduction to saml. http://www.xfront.com/saml/saml_f les/frame.htm. [Online]. [Accessed: May. 12, 2012].

[11] P. A. B. Galpottage and A. C. Norris. Patient consent principles and guidelines for e-consent: a new zealand perspective. *Health Informatics Journal*, 11(1):5–18, 2005.

[12] M. Gencer. The evolution of ietf standards and their production. *International Journal of IT Standards & Standardization Research*, 10(1):17 – 33, 2012.

[13] K. D. Lewis and J. E. Lewis. Web single sign-on authentication using saml. http://cogprints.org/6695/, August 2009. [Online]. [Accessed: May. 12, 2012].

[14] G. Marinskis, L. van Erven, M. G. Bongiorni, G. Y. H. Lip, L. Pison, C. Blomstroumlm-Lundqvist, and .

[15] M. A. Mizani. An xacml based framework for structured patient privacy policy. Master's thesis, Middle East Technical University, 2006.

[16] N. U. Moe and V. Oleshchuk. Decision-cache based xacml authorisation and anonymisation for xml documents. *Computer Standards & Interfaces*, n.d.

[17] T. NAMLI. Security, privacy, identity and patient consent management across healthcare enterprises in integrated healthcare enterprises (ihe) cross enterprise document sharing (xds) affinity domain. Master's thesis, Middle East Technical University, 2007.

[18] M. G. Prajapati, M. B. P. Marolia, and M. N. K. Patel. International organization for standardization. 2007.

[19] L. Seitz, E. Rissanen, T. Sandholm, B. S. Firozabadi, and O. Mulmo. Policy administration control and delegation using xacml and delegent. In *Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*, GRID '05, pages 49–54, Washington, DC, USA, 2005. IEEE Computer Society.

[20] M. Ulieru and D. Ionescu. Privacy and security shield for health information systems (e-health). *International Journal of Computer Systems Science Engineering*, 21(3):215–221, 2006.

[21] W. Xu, Q. Huang, and X. Liang. Design and implementation of a novel embedded real-time kernel based on hybrid architecture. In *Proceedings of the 2009 IITA International Conference on Control, Automation and Systems Engineering (case 2009)*, CASE '09, pages 132–135, Washington, DC, USA, 2009. IEEE Computer Society.

[22] R. Yavatkar, D. Pendarakis, and R. Guerin. A framework for policy-based admission control, 2000.

APPENDIX A

XACML Document

SAMPLE XACML POLICY

```
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
http://docs.oasis-open.org/xacml/accesscontrol-xacml-2.0-policy-schema-os.xsd"
PolicySetId="571FBA96-CD29-72C7-2967-72D071473363"
PolicyCombiningAlgId=
"urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
  <Description>All Rules</Description>
  <Target/>
  <Policy xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" PolicyId="12132923489"
    RuleCombiningAlgId=
  "urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
    <Description>
        Dentists can update my basic health information and conditions
    </Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema\#string">
                ROLECODE:DENTIST
            </AttributeValue>
```

```
      <SubjectAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
          DataType="http://www.w3.org/2001/XMLSchema\#string"/>
    </SubjectMatch>
  </Subject>
</Subjects>
<Resources>
  <Resource>
    <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema\#string">
          RESOURCECODE:CONDITION
      </AttributeValue>
      <ResourceAttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
          DataType="http://www.w3.org/2001/XMLSchema\#string"/>
    </ResourceMatch>
  </Resource>
  <Resource>
    <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema\#string">
          RESOURCECODE:BASICHEALTH
      </AttributeValue>
      <ResourceAttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
      DataType="http://www.w3.org/2001/XMLSchema\#string"/>
    </ResourceMatch>
  </Resource>
</Resources>
<Actions>
  <Action>
    <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema\#string">
          UPDATE
```

```
        </AttributeValue>
        <ActionAttributeDesignator
        AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
        DataType="http://www.w3.org/2001/XMLSchema\#string"/>
      </ActionMatch>
    </Action>
  </Actions>
</Target>
<Rule RuleId="12132923489:rule1" Effect="Permit">
  <Condition>
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
      <Apply
  FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-greater-than">
        <EnvironmentAttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-date"
            DataType="http://www.w3.org/2001/XMLSchema\#date"/>
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema\#dateTime">
            2010-12-17 09:30:27
        </AttributeValue>
      </Apply>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-less-than">
        <EnvironmentAttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-date"
            DataType="http://www.w3.org/2001/XMLSchema\#date"/>
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema\#dateTime">
            2011-12-20 19:30:27
        </AttributeValue>
      </Apply>
    </Apply>
  </Condition>
</Rule>
<Obligations>
  <Obligation
```

```
  ObligationId="urn:oasis:names:tc:xacml:example:obligation:email"
  FulfillOn="Permit">
  <AttributeAssignment
      AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:mailto"
      DataType="http://www.w3.org/2001/XMLSchema\#string">
          hastamail@deneme.com.tr
      </AttributeAssignment>
  <AttributeAssignment
      AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:text"
      DataType="http://www.w3.org/2001/XMLSchema\#string">
          Your medical record has been accessed by:
      </AttributeAssignment>
  <AttributeAssignment
      AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:text"
      DataType="http://www.w3.org/2001/XMLSchema\#string">
    <SubjectAttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
      DataType="http://www.w3.org/2001/XMLSchema\#string"/>
  </AttributeAssignment>
</Obligation>
<Obligation
  ObligationId="urn:oasis:names:tc:xacml:example:obligation:email"
  FulfillOn="Permit">
  <AttributeAssignment
      AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:mailto"
      DataType="http://www.w3.org/2001/XMLSchema\#string">
          hastayakin@srdc.com.tr
      </AttributeAssignment>
  <AttributeAssignment
      AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:text"
      DataType="http://www.w3.org/2001/XMLSchema\#string">
          Your medical record has been accessed by:
      </AttributeAssignment>
```

```
      <AttributeAssignment
          AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:text"
          DataType="http://www.w3.org/2001/XMLSchema\#string">
        <SubjectAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
          DataType="http://www.w3.org/2001/XMLSchema\#string"/>
      </AttributeAssignment>
    </Obligation>
  </Obligations>
</Policy>
<Policy xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" PolicyId="dsfdsffdsfsfds"
RuleCombiningAlgId=
"urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
  <Description>
  Doctors and Pharmacists can read my medications and conditions
</Description>
  <Target>
    <Subjects>
      <Subject>
        <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema\#string">
              ROLECODE:DOCTOR
          </AttributeValue>
          <SubjectAttributeDesignator
              AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
              DataType="http://www.w3.org/2001/XMLSchema\#string"/>
        </SubjectMatch>
      </Subject>
      <Subject>
        <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema\#string">
              ROLECODE:PHARMACIST
```

```
      </AttributeValue>
      <SubjectAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
          DataType="http://www.w3.org/2001/XMLSchema\#string"/>
    </SubjectMatch>
  </Subject>
</Subjects>
<Resources>
  <Resource>
    <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema\#string">
          RESOURCECODE:CONDITION
      </AttributeValue>
      <ResourceAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
          DataType="http://www.w3.org/2001/XMLSchema\#string"/>
    </ResourceMatch>
  </Resource>
  <Resource>
    <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue
          DataType="http://www.w3.org/2001/XMLSchema\#string">
          RESOURCECODE:MEDICATION
      </AttributeValue>
      <ResourceAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
          DataType="http://www.w3.org/2001/XMLSchema\#string"/>
    </ResourceMatch>
  </Resource>
</Resources>
<Actions>
  <Action>
    <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
```

```
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema\#string">
            READ
        </AttributeValue>
        <ActionAttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
            DataType="http://www.w3.org/2001/XMLSchema\#string"/>
      </ActionMatch>
    </Action>
  </Actions>
</Target>
<Rule RuleId="dsfdsffdsfsfds:rule1" Effect="Permit">
  <Condition>
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
      <Apply
  FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-greater-than">
        <EnvironmentAttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-date"
            DataType="http://www.w3.org/2001/XMLSchema\#date"/>
        <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema\#dateTime">
                2001-12-17 09:30:27
            </AttributeValue>
      </Apply>
      <Apply
  FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-less-than">
        <EnvironmentAttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-date"
            DataType="http://www.w3.org/2001/XMLSchema\#date"/>
        <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema\#dateTime">
                2011-12-20 19:30:27
            </AttributeValue>
      </Apply>
```

```
      </Apply>
    </Condition>
  </Rule>
  <Obligations>
    <Obligation ObligationId="urn:oasis:names:tc:xacml:example:obligation:email"
      FulfillOn="Permit">
      <AttributeAssignment
          AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:mailto"
          DataType="http://www.w3.org/2001/XMLSchema\#string">
              desee@srdc.com.tr
          </AttributeAssignment>
      <AttributeAssignment
          AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:text"
          DataType="http://www.w3.org/2001/XMLSchema\#string">
              Your medical record has been accessed by:
          </AttributeAssignment>
      <AttributeAssignment
          AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:text"
          DataType="http://www.w3.org/2001/XMLSchema\#string">
        <SubjectAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
          DataType="http://www.w3.org/2001/XMLSchema\#string"/>
      </AttributeAssignment>
    </Obligation>
  </Obligations>
</Policy>
<Policy xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" PolicyId="abnmkllhgllhgf"
  RuleCombiningAlgId=
"urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
  <Description>Nurses can not update my basic health information</Description>
  <Target>
    <Subjects>
```

```
  <Subject>
    <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema\#string">
          ROLECODE:NURSE
      </AttributeValue>
      <SubjectAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
          DataType="http://www.w3.org/2001/XMLSchema\#string"/>
    </SubjectMatch>
  </Subject>
</Subjects>
<Resources>
  <Resource>
    <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema\#string">
          RESOURCECODE:BASICHEALTH
      </AttributeValue>
      <ResourceAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
          DataType="http://www.w3.org/2001/XMLSchema\#string"/>
    </ResourceMatch>
  </Resource>
</Resources>
<Actions>
  <Action>
    <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema\#string">
          UPDATE
      </AttributeValue>
      <ActionAttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
      DataType="http://www.w3.org/2001/XMLSchema\#string"/>
    </ActionMatch>
```

```
      </Action>
    </Actions>
  </Target>
  <Rule RuleId="abnmkllhgllhgf:rule1" Effect="Deny">
  </Rule>
</Policy>
<Policy xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" PolicyId="12345678hnvdsad"
  RuleCombiningAlgId=
  "urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
  <Description>Doctors can update my hospital visits and operations</Description>
  <Target>
    <Subjects>
      <Subject>
        <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema\#string">
              ROLECODE:DOCTOR
          </AttributeValue>
          <SubjectAttributeDesignator
              AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
              DataType="http://www.w3.org/2001/XMLSchema\#string"/>
        </SubjectMatch>
      </Subject>
    </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema\#string">
              RESOURCECODE:HOSPITALVISIT
          </AttributeValue>
          <ResourceAttributeDesignator
              AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
              DataType="http://www.w3.org/2001/XMLSchema\#string"/>
```

```
    </ResourceMatch>
  </Resource>
  <Resource>
    <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema\#string">
          RESOURCECODE:OPERATION
      </AttributeValue>
      <ResourceAttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
      DataType="http://www.w3.org/2001/XMLSchema\#string"/>
    </ResourceMatch>
  </Resource>
</Resources>
<Actions>
  <Action>
    <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema\#string">
          READ
      </AttributeValue>
      <ActionAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
          DataType="http://www.w3.org/2001/XMLSchema\#string"/>
    </ActionMatch>
  </Action>
  <Action>
    <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema\#string">
          UPDATE
      </AttributeValue>
      <ActionAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
          DataType="http://www.w3.org/2001/XMLSchema\#string"/>
    </ActionMatch>
```

```
        </Action>
      </Actions>
    </Target>
    <Rule RuleId="12345678hnvdsad:rule1" Effect="Permit">
    </Rule>
  </Policy>
</PolicySet>

SAMPLE XACML REQUEST
<Request
  xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Subject>
    <Attribute
      AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>admin</AttributeValue>
    </Attribute>
    <Attribute AttributeId="group"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>admin</AttributeValue>
    </Attribute>
  </Subject>
  <Resource>
    <Attribute
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>
        http://localhost:8280/services/echo/echoString
      </AttributeValue>
    </Attribute>
  </Resource>
  <Action>
```

```
    <Attribute
      AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>read</AttributeValue>
    </Attribute>
  </Action>
  <Environment/>
</Request>
```