Adding Privacy Protection to Policy Based Authorisation Systems

A thesis submitted to
The University of Kent at Canterbury
for the degree of

Doctor of Philosophy
in
Computer Science

by

KANIZ FATEMA

November, 2013

To my husband
Without his patience and support, this Ph.D. would have remained a dream

Abstract

An authorisation system determines who is authorised to do what i.e. it assigns privileges to users and provides a decision on whether someone is allowed to perform a requested action on a resource. A traditional authorisation decision system, which is simply called authorisation system or system in the rest of the thesis, provides the decision based on a policy which is usually written by the system administrator. Such a traditional authorisation system is not sufficient to protect privacy of personal data, since users (the data subjects) are usually given a take it or leave it choice to accept the controlling organisation's policy. Privacy is the ability of the owners or subjects of personal data to control the flow of data about themselves, according to their own preferences. This thesis describes the design of an authorisation system that will provide privacy for personal data by including sticky authorisation policies from the issuers and data subjects, to supplement the authorisation policy of the controlling organisation. As personal data moves from controlling system to controlling system, the sticky policies travel with the data.

A number of data protection laws and regulations have been formulated to protect the privacy of individuals. The rights and prohibitions provided by the law need to be enforced by the authorisation system. Hence, the designed authorisation system also includes the authorisation rules from the legislation. This thesis describes the conversion of rules from the EU Data Protection Directive into machine executable rules. Due to the nature of the legislative rules, not all of them could be converted into deterministic machine executable rules, as in several cases human intervention or human judgement is required. This is catered for by allowing the machine rules to be configurable.

Since the system includes independent policies from various authorities (law, issuer, data subject and controller) conflicts may arise among the decisions provided by them. Consequently, this thesis describes a dynamic, automated conflict resolution mechanism. Different conflict resolution algorithms are chosen based on the request contexts.

As the EU Data Protection Directive allows processing of personal data based on contracts, we designed and implemented a component, Contract Validation Service (ConVS) that can validate an XML based digital contract to allow processing of personal data based on a contract.

The authorisation system has been implemented as a web service and the performance of the system is measured, by first deploying it in a single computer and then in a cloud server. Finally the validity of the design and implementation are tested against a number of use cases based on scenarios involving accessing medical data in a health service provider's system and accessing personal data such as CVs and degree certificates in an employment service provider's system. The machine computed authorisation decisions are compared to the theoretical decisions to ensure that the system returns the correct decisions.

Acknowledgements

First and foremost, I would like to thank my supervisor Professor David W. Chadwick for his guidance, care, patience and encouragement. He taught me the methods of research and always pushed me forward in every situation. He was always by my side during the hard times. He is a person who is down to earth and full of sparkling enthusiasm. Without his leadership, teaching, inspiration, criticism and support, this work would have never been possible.

I would like to express my gratitude to my examiners, Professor Jason Crampton and Dr. Eerke Boiten, for their critical review of the thesis and insightful comments. Their analysis and direction gave this thesis a richer dimension.

I am so grateful to Brendan Van Alsenoy, who is an expert on the European data protection law, for his invaluable advice and help. His support with his profound knowledge in the area of data protection law was a great help during the process of conversion of the Legal rules into machine executable rules.

I would like to thank Dr. Stijn Lievens who has implemented the complex authorisation system which could have been impossible for me to finish within the time boundary. I would like to express my gratitude to the TAS3 project (an IST FP7 funded Integrated Project) for funding my research. I want to thank Ana Ferreira, George Inman, Kristy Siu and Matteo Casenove for their valuable help and encouragement.

I am very thankful to my mother for being on my side in every situation of my life. Thanks to my father who inspired me "to write an essay which has not been written by any one on earth yet to get a Ph.D." when I was a student at primary school. Thanks to my brother for his endless support and teaching me always to be positive. Special thanks to my husband, who is a source of inspiration and tenderness. He pushed me forward compromising his own career. Without his co-operation in every aspect of our daily life and the responsibility of raising our baby, it would have been impossible for me to finish this work. Finally, thanks to my toddler son for smiling so sweetly and taking away all the stress and exhaustion after working hard.

Table of Contents

Table of Contents

List of Tables

List of Figures

Acronyms

ABAC	Attribute Based Access Control
ACP	Access Control Policy
ACR	Access Control Rule
BTG	Break The Glass
CNL	Controlled Natural Language
CRP	Conflict Resolution Policy
CRR	Conflict Resolution Rule
CVS	Credential Validation Service
ConVS	Contract Validation Service
DAC	Discretionary Access Control
DCA	Decision Combining Algorithm
DN	Distinguishing Name
EU DPD	European Union Data Protection Directive
HIC	Health Insurance Company
IDA	IDentifying Attribute
HIPAA	Health Insurance Portability and Accountability Act
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PID	Policy IDentity
PII	Personal Identity Information
P-PAAS	Privacy-Protecting Advanced Authorisation System
PERMIS	PrivilEge and Role Management Infrastructure Standards
Legal PDP	Legal Policy Decision Point
RBAC	Role Based Access Control
RID	Resource IDentity
TAS3	Trusted Architecture for Securely Shared Services
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language

Chapter 1

1 Introduction

1.1 Introduction

With the advancement of modern technologies, data have become digital, and we have gained the flexibility of accessing data or resources electronically. The Internet is gaining more and more dominance in every aspect of our lives. Communication, learning new things, satisfying daily necessities such as shopping, paying bills or searching for information is just a mouse click away. Presenting oneself in the job market worldwide, promoting a business and even maintaining a social life has become very flexible. However, the blessings of the free flow of information are not free from unwanted consequences such as identity theft and threats to privacy.

Many web sites today collect PII (Personal Identity Information) such as names and addresses from users through online registration, surveys, user profiles, and order fulfillment processes etc. Also, different personal data such as educational records, health data, credit card information and so on are collected by many organisations in order to provide consumers with services. An example of such a service is an online job agency where people post their CVs to hunt for jobs worldwide. Once released, users lose control of their personal data. Personal data like CVs, which contain sensitive information, may invite not only job offers but also identity theft. Theft of personal identity information has serious consequences ranging from significant financial loss to becoming a suspect in a crime which was committed with a stolen identity. Innocent people have been arrested due to a crime committed by an identity thief (Penycate 2001). In the UK, the number of identity theft was an alarming 20% higher in the first quarter of 2010 compared with the same period in 2009. About 27,000 victims were recorded by the Credit Industry Fraud Avoidance System (CIFAS) members during the first 3 months of 2010 (CIFAS report 2010). Loss of personal data from reputable organisations such as HSBC bank (BBC news 2009), Zurich Insurance (BBC news 2010), Sony Play Station (BBC news 2011), the University of Nottingham (Times Higher Education news 2014) or HotelHippo.com (BBC news 2014) has been reported. As a consequence, concerns for the privacy of electronic personal data are increasing day by day (IBTImes report 2011, NBC news 2008, Voice of America news 2009). This necessitates the need for more technical control over the personal data which is collected or stored online to enable users to gain more confidence and trust when submitting their personal data. Technical controls will help to protect personal data from being misused as well as helping to enforce data protection laws.

1.1.1 Personal data

According to the Data Protection Act 1998 (Data protection act 1998) information can be

personal data if any of the following conditions is true

1. If the information (in conjunction with other information) can identify a living individual.
2. The information relates to an identifiable living individual.
3. The information is obviously about a particular individual; e.g. medical record, criminal record.
4. The information is linked to an individual.
5. The information informs or influences actions or decisions affecting an identifiable individual.
6. The information has biographical significance in relation to the individual.
7. The information focuses on the individual as its central theme.
8. The information has impact (or potential to impact) on an individual.

1.1.2 Privacy

Privacy is a fundamental human right which was first defined as "the right to be left alone" by the United States Supreme Court Justice Louis Brandeis and Samuel Warren (Lengwiler 2004). Some other views of privacy are – the protection of an individual's independence, integrity, dignity, secrecy, anonymity, solitude, protection against intrusion into an individual's personal life or affairs (Allmer 2011).

Professor Roger Clarke (Clarke 2006) has defined the different dimensions of privacy as follows –

- **Privacy of person**, which is also referred to as 'bodily privacy', is concerned with the integrity of an individual's body such as blood transfusion without consent, compulsory provision of samples of body fluids and body tissue and so on.
- **Privacy of personal behaviour** relates to all aspects of behaviour such as sexual preferences and habits, political activities and religious practices.
- **Privacy of personal communications** relates to privacy of communications using various media without being monitored. This is sometimes referred to as 'interception privacy'
- **Privacy of personal data** is also referred to as 'data privacy' and 'information privacy', relates to controlling whether or how personal data can be gathered, stored, processed or selectively disclosed. European Union Data Protection Directive (EU DPD) (Directive 95/46/EC 1995) has specified the criteria for privacy protection of personal data which include: notifying the data subject of the processing of data, transparent processing, limiting disclosure of personal data and many more (discussed in details in Chapter 4).

The aspect of privacy that is of particular interest of computer science society is information privacy (Preibusch 2013). Information privacy ensures the control that the data subject has over information about him/herself (Tavani 2007). The online information privacy is defined by Malhotra et al. as the privacy related to the collection of the user's personal information by websites, others' access to the user's personal information, the user's control over the collected information, and the user's awareness of how the collected information is used (Malhotra, Sung and Agarwal 2004).

1.1.3 Privacy breach payoff

"Privacy breach payoff" shows the negative consequences that can be occurred due to the lack of or poor privacy protection (Cavoukian 2009). For example –

- Harm to the person whose data are used or disclosed inappropriately.
- Damage to an organisation's reputation.

- Financial loss.
- Loss of business due to negative publicity.
- Violation of privacy laws.
- Destruction of confidence and trust in the industry.

1.1.4 Privacy and security of data

The primary purpose of information security is to provide the confidentiality, integrity and availability of data (Andress 2011). Confidentiality is the ability to protect access to data from unauthorised persons, integrity refers to the ability to protect data from being changed in an unauthorised manner and availability is the ability to access data whenever needed. The concept of privacy is related to confidentiality but they are not the same (Andress 2011, Borking 2011). Privacy is not simply limited to controlling the disclosure of data, but also requires notifying the data subject about the processing of his/her data, minimising the disclosure, controlling the purpose of use, and limiting the period of storing the data, ensuring that the data subject consents to the processing of the data. The difference between privacy and the different aspects of information security has been specified by Borking (Borking 2011) as shown in Table 1.1. The coloured cells represent the criteria of privacy that are (either strongly or slightly) related to the corresponding security concerns. As can be seen, privacy covers many more concerns than all those of traditional information security.

Table 1.1: Difference between information security and privacy (Borking 2011)

Privacy Criterion / Information Security	Reporting of processing	Transparent processing	"As required" processing	Lawful basis of data processing	Data quality conservation	Rights of the parties involved	Data traffic with countries outside EU	Processing personal data by processor	Protection against loss and unlawful processing
Confidentiality									
Availability									
Integrity									

1.1.5 Access control components and generic access control model

Access control is a process of determining whether a request to access a resource object by a subject should be granted or denied. It should ensure that a user can only access the resources s/he is authorised to (di Vimercati, Samarati and Jajodia 2005). *Access (Action)* defines the operation (e.g. read/ write/ delete/ copy etc.) to be performed on an *object*. A *subject* (e.g. user, system or process) is an active component of an access control model that requests access to an object and an *object* (e.g. file, data base, program, system component) is the resource that is being protected by the access control system. A *Subject* sends *requests* to an access control system specifying an *action* to perform on an *object* and the access control system can grant or deny the request based on whether certain policy constraints have been satisfied.

The access control mechanism constrains the interactions between users and the protected

resources (Crampton and Morisset 2012). ISO/IEC 10181- 3:1996 has defined a generic Access Control Framework (ACF) as shown in Figure 1.1 (ISO 1996). It consists of four components: Initiators (subjects), Targets (resource objects), Access Control Enforcement Functions (AEFs) and Access Control Decision Functions (ADFs). Initiators submit access requests (also known as *user requests*). An access request specifies the operation to be performed on the Target. The AEF transforms the request into one or more decision requests (also known as *authorisation queries*) and sends these to the ADF. The ADF decides whether a decision request should be granted or denied based on some policies and sends the decision back to the AEF.

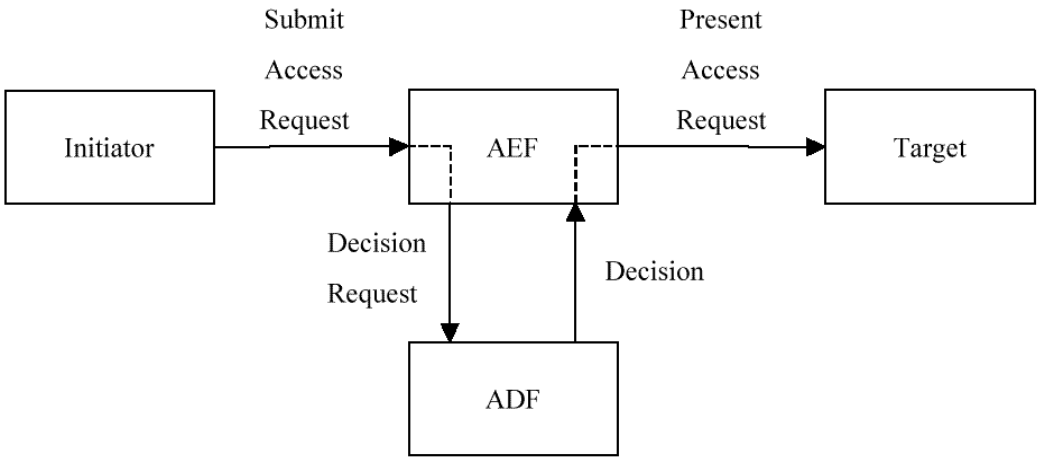

Figure 1-1. Access Control Framework (ACF) from standard ISO/IEC 10181-3

1.1.6 Privacy policy and access control policy

In a policy based authorisation system actions on a resource are controlled by policies; access control policies and privacy policies both protect access to data. An access control policy can relate to any data and any resource in general and typically an organisation's administrator determines the authorisation policy (Crampton and Morisset 2012). However, a privacy policy only relates to personal data and needs to include the preferences of the data subject regarding who can access the personal data for which purposes. A privacy policy is similar to other access control policies but it needs some other features such as purpose of use, retention period and certain types of obligation such as notifying the data subject and keeping an audit trail, and deleting the data after a certain amount of time.

1.2 Research Questions to Answer

The aim of this research is to design and implement a policy-based authorisation infrastructure to meet the requirements for privacy protection of personal data.

The following list provides the questions that this research attempts to answer-

1. What are the requirements for the protection of privacy of personal data?
2. Is a traditional authorisation system sufficient to provide the privacy requirements?
3. Can we provide privacy protection with a policy based authorisation system?

4. Can we design a policy based authorisation system for wide scale distributed use where more than one policy language may be simultaneously involved?
5. If a number of policies from different authorities are involved, how can we handle conflicts among the decisions of the policies?
6. Can we extract access control rules from the data protection legislation so that these legislative rules are automatically enforced by the policy based system?
7. As the EU DPD allows processing of personal data based on contracts, how can we present and validate digital contracts and how can we validate contract based access to personal data?

The following **research method** has been adopted in this thesis-

1. Analysis of prior published work on privacy protection of personal data.
2. Identifying the requirements for a system protecting privacy based on the previously published works and legislative documents.
3. Hypothesize a model for providing privacy in an authorisation system to satisfy the requirements.
4. Analysis of prior research related to the functionality of each component of the model and designing the new required components.
5. Designing a framework for distributed enforcement of the privacy policies, taking into account the requirements of multiple policies and multiple policy languages.
6. Designing a component that can validate a digital contract.
7. Designing a dynamic conflict resolution strategy that can choose a conflict resolution algorithm based on the request context.
8. Designing a methodology to obtain and implement access control rules from the data protection legislation.
9. Designing a model to obtain policies from end users who have no knowledge of writing access control or privacy policies.
10. Validation and evaluation of the implemented model.

1.3 Research Contributions

The main contributions of this research are summarised below-

1. Designed a Privacy Protecting Advanced Authorisation System, P-PAAS, which supports multiple policies from multiple authorities in different policy languages. The conceptual design of the system is provided in Chapter 3, and the implementation is described in Chapter 5.
2. Designed a dynamic conflict resolution strategy that can choose a conflict resolution algorithm based on the request context. The conceptual design is described in Chapter 3 and the implementation in Chapter 5. The design is an enhancement of the method proposed by Mohan et al. (Mohan and Blough 2010) in order to make it suitable for privacy protecting scenarios that incorporate independent policies from multiple authorities.
3. Designed and implemented a new component, Contract Validation Service (ConVS), that can validate a digital contract to allow processing of personal data based on a contract. The conceptual design of the component is discussed in chapter 3 and the implementation of the component is described in chapter 5.

4. Proposed and developed a methodology to obtain automated access control rules from legislation, and map these into two different policy languages. The methodology is provided in detail in Chapter 4.
5. Validated the system based on use case scenarios and conducted performance tests of the system. The validation and performance tests of the system are presented in Chapter 5.

The rest of the thesis is organised as follows-
Chapter 2 presents the previous research related to the authorisation models, privacy protecting authorisation systems, obligation enforcement tactics, conflict resolution strategies, privacy protection for the data in the cloud and the works related to obtaining authorisation rules from legislation.
Chapter 3 provides the conceptual design of the overall system and the various components of the system.
Chapter 4 presents a methodology to obtain the authorisation rules from the European Union Data Protection Directive (EU DPD) (Directive 95/46/EC 1995).
Chapter 5 presents the implementation, validation and testing of the P-PAAS system. The Legal access control and conflict resolution rules are presented, along with validation tests of the Legal rules and of the system when it includes policies from various authorities. Performance tests of the system are reported when it is installed in a single machine and in a cloud server.
Finally, **Chapter 6** concludes the thesis.

The following papers have been published with the research described in this thesis.

- Kaniz Fatema, David Chadwick, "Resolving Policy Conflicts - Integrating Policies from Multiple Authors". In CAiSE 2014 International Workshops, Thessaloniki, Greece, June 16-20, 2014.

This paper shows that the static conflict resolution strategy of XACML is not always sufficient to satisfy the policy needs of an organisation where multiple parties provide their own individual policies. Different conflict resolution strategies are often required for different situations. Thus combining one or more sets of policies into a single XACML 'super policy' that is evaluated by a single policy decision point (PDP), cannot always provide the correct authorisation decision, due to the static conflict resolution algorithms that have to be built in. Therefore a dynamic conflict resolution strategy is proposed that chooses different conflict resolution algorithms based on the authorisation request context. The proposed system receives individual and independent policies, as well as conflict resolution rules, from different policy authors, but instead of combining these into one super policy with static conflict resolution rules, each policy is evaluated separately and the conflicts among their authorisation decisions are dynamically resolved using the conflict resolution algorithm that best matches the authorisation decision request. It further combines the obligations of independent policies returning similar decisions which XACML cannot do while keeping each author's policy intact. The contents of this paper are presented in Chapter 3.

- Kaniz Fatema, David Chadwick, Brendan Van Alsenoy, "Extracting access control and

conflict resolution policies from European data protection law." *IFIP/PrimeLife International Summer School.* Trento, Italy: Springer, September 5-9, 2011. 59-72.

This paper presents the conversion process of the EU DPD into machine executable rules and is presented in Chapter 4. The procedure for obtaining machine executable rules from the EU Data Protection Law presented in this paper is modified in Chapter 4 to provide grammar for presenting the rules and getting executable rules automatically. One of the co-authors, Brendan Van Alsenoy, is a lawyer and an expert on the European data protection law. He contributed to the analysis of the Legal rules which is an essential step in the conversion of them into machine executable rules.

- Kaniz Fatema, David Chadwick, Stijn Lievens, "A Multi Privacy Policy Enforcement System". In *Privacy and Identity 2010, IFIP AICT 352*, 2011, pp. 297–310.

This paper presents the requirements that a privacy protecting authorisation system should meet, and how our proposed system could be used for protecting the privacy of personal data. It also provides a use case scenario of protecting access to personal data with our system and the contents are presented in Chapter 3. One of the co-authors, Stijn Lievens, implemented the idea presented and contributed to writing the implementation section.

- David Chadwick, Kaniz Fatema, "A Privacy Preserving Authorisation System for the Cloud", J.C.S.S Cloud Computing 2011 Special Issue, Received 15 December 2010, Revised 20 May 2011, Accepted 8 December 2011. Available online 29 December 2011.

This paper presents the applicability of our authorisation system in cloud computing. The policy based authorisation infrastructure can be run by a cloud provider as a service for its users. It will protect the privacy of users' data by allowing the users to set their own privacy policies, and then enforcing them so that no unauthorised access is allowed to their data. The performance figures are presented which show that the system performs well and that each additional PDP only imposes a small overhead. Chapter 5 contains the results of the performance tests that were run using our system in the cloud.

- David Chadwick, Kaniz Fatema, "Distributed Privacy Policy Enforcement by using Sticky Policies", W3C workshop on Privacy and data usage control, 4-5 October, 2010, Cambridge, MA.

This paper presents the details of our sticky policy enforcement mechanism which contains information on how policies can be stored and attached with the data and how the sticky policy can be passed to another system for ensuring its distributed enforcement. The contents of the paper are available in Chapter 3.

- David Chadwick, Kaniz Fatema. "An Advanced Policy Based Authorisation Infrastructure". Proc DIM'09, November 13, 2009, Chicago, Illinois, USA. ACM.

This paper presents the idea of an advanced authorisation system that can use multiple PDPs capable of evaluating multiple policy languages. A component, master PDP, is designed to combine the decisions of multiple PDPs and resolve their conflicts. The basic idea presented in this paper has later been modified to ensure its applicability for the purpose of privacy protection. The design presented in Chapter 3 is based on the initial idea described in this paper.

Chapter 2

2 Review of Related Work

2.1 Introduction

This chapter provides an extensive review of the relevant literature. While designing a privacy protecting authorisation system the existing models and standards of authorisation systems are reviewed. Furthermore, the requirements for privacy protection are identified and whether traditional authorisation models are sufficient to satisfy them are justified. The previous works related to the design of privacy protecting authorisation systems are studied to ascertain how others have added privacy protection and also to identify what is missing and how to provide solutions to that. Since enforcing obligations is one of the important requirements for privacy protecting systems a review of the prior research relevant to that is performed. Our designed authorisation system integrates independent policies from a number of authorities; hence a mechanism is needed for resolving conflicts among the decisions returned by the various policies of different authorities. Therefore, the previous research related to conflict resolution strategies is studied as well as the research on privacy protection in the cloud. Our designed system includes authorisation rules from the EU DPD (Directive 95/46/EC 1995), so consequently, works related to obtaining authorisation rules from legislation have been reviewed.

2.2 Authorisation Models

Authentication and authorisation are two terms (with different meanings) very frequently used in the field of information security. Authentication is a way of identifying an entity and is a process by which it is possible to determine whether someone/something is genuine. In contrast, authorisation is a process of determining whether the person/entity has the right to do something.

An authorisation system[1] determines whether a subject (a person or process) has the right to perform an action on an object (resource). Access control systems are categorised into mandatory access control (MAC) and discretionary access control (DAC) based on the entity that enforces the access control. Based on the methods used for implementing access control the authorisation systems can be categorised as identity, role, attribute, or token based and so on. Systems that provide protection to resources based on policies are much more flexible than those offering protection based on lists of authorised users and various policy languages are also introduced regarding this. In a traditional policy based access control system requests

[1] The terms access control system and authorisation system have been used as alternative terms in this thesis.

to access the secured resources are granted based on policies mainly set by the administrator based on the organisation's rules. It is up to the administrator to determine, who is authorised to access the secured data, and up to what level, by using an authorisation policy. In this section, the authorisation models from the literature are presented.

2.2.1 Mandatory Access Control (MAC)

Mandatory Access Control (MAC) is implemented as a multi-level access control system based on a hierarchical classification of levels and its policy is defined only by the system administrator. Each resource and subject in the system is classified as a member of one of those levels. One example of the MAC model is the Bell-LaPadula (Bell and La Padula 1976) confidentiality model, which concentrates on the confidentiality of data. In this multilevel-security model a subject can read an object only if his classification level is greater than or equal to that of the object. The Biba integrity model (Biba 1977) is another example of the MAC model that concentrates on the integrity of data. It allows the subject to read data of higher classification levels and write data to lower classification levels which hinders the confidentiality as it allows the information to flow at different classification levels.

This model does not provide fine grained access control to allow subjects to access resources based on complex conditions. For example, it is not possible to allow a person to access a resource based on the consent of the data subject or only for a legitimate purpose.

2.2.2 Discretionary Access Control (DAC)

Contrasted to MAC, in DAC the owner of an object defines who is allowed to access his/her resources and the creator of an object is the owner by default. This model can be implemented by access control lists, access control matrices or capability certificates (see the next section), and allows the flexibility of defining access control policies by the owner. Nevertheless, as the policies are coming from and being maintained by the owners of various objects, not by a centralised administrator like MAC, the overall security policy of the system is hard to verify (e.g., what objects a specific subject is allowed to access) (Mohan 2011). Furthermore, it is more prone to misconfiguration of policies as they are specified by the resource owners.

2.2.3 Identity based access control

In an identity based system the access rights are based on the identity of the subject and can be implemented using an access control matrix, access control lists or capabilities (Sandhu and Samarati 1994).

In an access control matrix the access rights are presented in a matrix where the rows represent the subjects and the columns represent the objects. Each cell of the matrix presents the operations a subject can perform on an object.

Access control lists (ACLs) are typically defined for each object and specify the list of subjects who can access it. The problem of using ACLs is that it is hard to determine for a specific subject what resources s/he is allowed to access. If a user leaves the system all the ACLs allowing his/her access need to be identified and modified which makes the system less flexible to use.

Capabilities are the subject centric view of the access control matrix. Each subject is given lists of access rights for a set of objects in the form of a capability certificates. In order to access a resource the subject presents his/her capability certificates to prove him/herself entitled. The problem of using capabilities is that it is hard to verify who are allowed to access a resource as it requires checking the capability certificates of all subjects.

2.2.4 Role Based Access Control (RBAC) model

The RBAC has become the prominent model for authorisation due to the advantages it offers reducing the complexity of security administration. Unlike identity based systems adding or removing users is much easier in this model. The formal model of RBAC (also known as Role Based Security) was introduced by Ferraiolo and Kuhn (Ferraiolo and Kuhn 1992) and a framework was proposed by Sandhu et al. (Sandhu, et al. 1996). Later they were both integrated to form a unified standard (Sandhu, Ferraiolo and Kuhn 2000).

The main concept of RBAC is that permissions are associated with roles and users get permissions based on the roles assigned to them. Different roles are defined according to the activities and structure of an organisation and the administrator of the organisation assigns appropriate permissions according to the role. For example, a student may not have permission to see a confidential database, a lecturer may have permission to read the confidential database but not to change it, whereas an administrator may have permission to both read and change it. When the security requirement changes, only the permissions assigned to roles are changed and it does not require changing of permissions for every user separately. When a user is assigned a role they get all the permissions assigned to that role. According to the ANSI standard (ANSI 2004) the Core-RBAC model is described next.

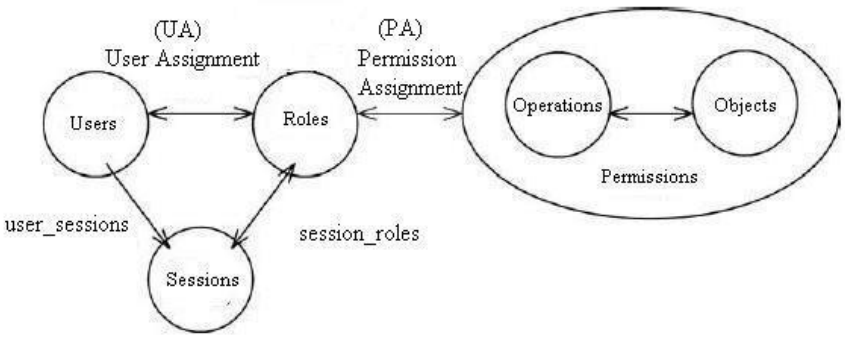

Figure 2-1. Core-RBAC

The Core-RBAC model (Figure 2-1) is described by the elements – users, roles, operations, objects and sessions. Users are assigned roles in a many-to-many relationship; roles are assigned permissions in a many-to-many relationship where permission is an approval to perform an operation on one or more objects. A user may have many roles assigned to him/her and can activate a subset of the roles for a session. Each session is assigned to only one user

while one user can have many sessions. Before exercising the permissions assigned to a role held by a user the user must activate the role. In this model users are not only human beings but can also be other entities like machines, networks and intelligent autonomous agents.

The hierarchical RBAC model introduces a new component Role Hierarchy (RH) which is a natural means of structuring roles to reflect an organisation's lines of authority and responsibility. A role r1 inherits the role r2 if all the permissions of r2 are also permissions of r1. In other words the roles in an upper level of a role hierarchy inherit the permissions assigned to the roles in a lower level. For example, the lecturer role inherits the permissions assigned to a student role but not vice-versa.

2.2.5 Attribute Based Access Control (ABAC) model

The Attribute Based Access Control Model (ISO 1996, Yuan and Tong 2005, Wang, Wijesekera and Jajodia 2004) is an extension of the RBAC Model where permissions are given based on the attributes possessed by the user and attributes are not limited to organisational roles, they can be anything such as degree, qualification, name, age and of course roles. Attributes (usually assigned by Attribute Authorities (AAs)), are assigned to users and permissions are assigned to attributes and thus users get permissions based on the attributes they possess. Attributes can be assigned to the subject, object, action and environment. The ABAC model is more flexible and manageable than traditional identity based access control models where access to objects is restricted by the identities of subjects; and consequently is being widely used by many real life access control systems such as on-line shopping with credit cards. This model offers the flexibility of using various attributes and is especially useful for presenting the EU DPD rules in the form of machine executable rules by the use of number of attributes. Therefore, we have used this access control model for our work presented in this thesis.

2.2.6 Role based Trust management (RT) framework

The RT (Li, Mitchell and Winsborough 2002, Li and Mitchell 2003) framework offers to combine the strength of RBAC and TM (trust management). From RBAC it takes the concept of role, session and selective role activation and from TM the idea of a credential for managing distributed authority. RT defines a language for representing policies and credentials for controlling access in decentralised collaborative systems where protected resources and the requesting subjects belong to different security domains controlled by different authorities. The most basic language in the RT family of TM languages is RT_0 which consists of two basic constructs: entities and role names. The principals/ entities are uniquely identified individuals, processes, public keys etc. and role names are string identifiers. A role in RT is presented by an entity followed by a role name, separated by a dot, e.g., A.r which indicates that the role A.r is defined by entity A and only A can issue policy defining the role A.r and its members.
There are various kinds of credentials in RT_0, each corresponding to a different way of defining role membership (Li, Winsborough and Mitchell 2001).
RT_0, only allows a simple form of role name and does not take any parameters. RT has introduced some policy concepts such as parameterised roles, intersections of roles, role-product operators, manifold roles, and delegation of role activations. RT_1 extends RT_0 to allow parameterised roles and RT_2 extends RT_1 to allow grouping of logically related objects. RT^T provides manifold roles and role-product operators to express thresholds, i.e. the agreement

of a certain number of entities out of a set of entities and separation of duty policies. RT^D provides delegation of role activations. The RT model provides advantages of

- a declarative, logic based semantic foundation,
- support for distributed chain discovery and vocabulary agreement,
- strongly-typed credentials and policies,
- flexible delegation structures,
- expressive support for Separation-of-Duty (SoD) policies.

Whilst this model considers the combination of RBAC and Trust management, it is not suitable for our purpose since the Legal rules are not based only on roles, therefore they cannot be converted into enforceable RT rules.

Many of the concepts of RT are incorporated into our model and into the PERMIS implementation. Similar to RT, we have the concept of credentials and a Credential Validation Service with rules to say who is trusted to issue which credentials. We also support parameterised roles, since roles comprise a type and value (where value is the parameter in RT notation). Whilst PERMIS also supports Separation of Duties (Chadwick, et al. 2007) and delegation of authority (Chadwick, et al. 2009) these are not the features that we make use of in this thesis.

2.2.7 OAuth authorisation model

OAuth (OAuth 2012) provides a way to publish and access protected data without requiring sharing credentials. Suppose that a user has some protected data in a service provider; with the OAuth protocol the consumer can gain access to the resources from the service provider, without requiring the user to disclose his/her service provider credentials to the consumer.

For example, if a user has some photos in a website (photo.net) OAuth allows them to permit the other site (print.com) to access the private data (photos) on behalf of the user without entering his/her username or password.

OAuth defines four roles: *Resource owner* is the entity that grants access to a protected resource, *resource server* accepts and responds to the requests to access protected resources using *access tokens*, *client* is an application that requests to access protected resources on behalf of the resource owner and *authorisation server* issues access tokens to the client after successfully authenticating the resource owner and obtaining authorisation.

OAuth uses Tokens (a string denoting a specific scope, lifetime and other access attributes) instead of users' credentials for requesting access to the protected resources. The following two types of token are used -

Refresh Token: The Refresh Token is a string representing the authorisation granted to the client by the resource owner and is used only to get a new Access Token when the current access token expires or becomes invalid.

Access Token: Only the Access Token can be used by the consumer to access the protected resources on behalf of the user. Access Tokens may have a limited lifetime. With this token the consumers can be authorised to have limited access to the protected resources. The user has the right to revoke the Access Token.

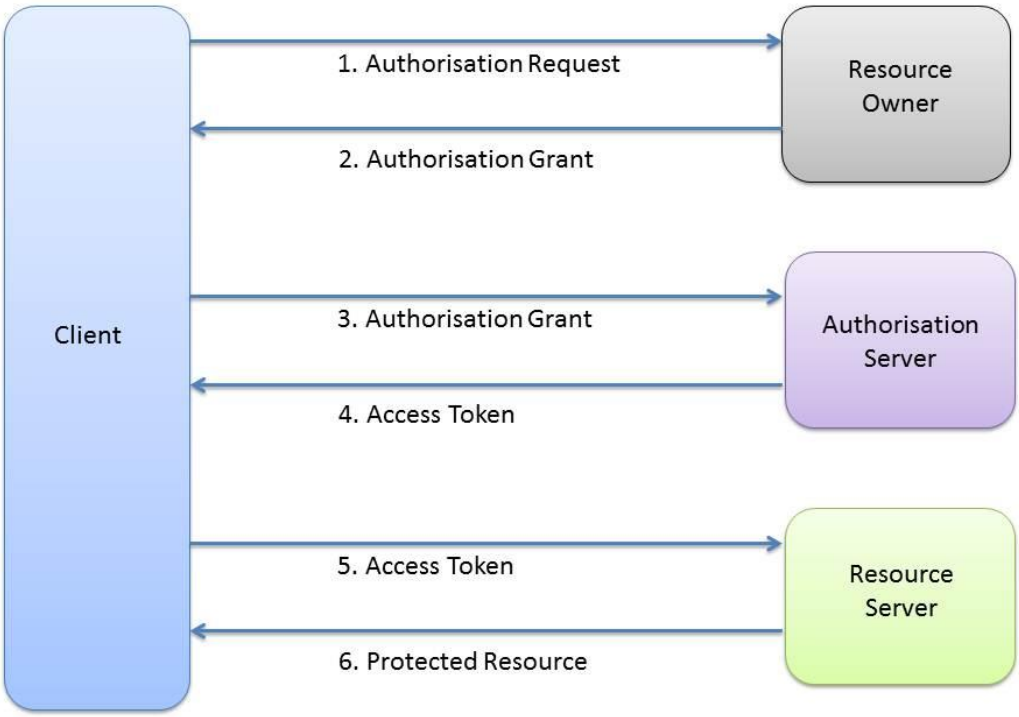

Figure 2-2. Protocol flow of OAuth

OAuth allows accessing protected resources on behalf of the resource owner without sharing the credentials. The protocol flow of OAuth involves the following steps:

1. The client requests for authorisation from the resource owner directly or via the authorisation server.
2. The client receives an authorisation grant credential.
3. The client requests an access token from the authorisation server and presents the authorisation grant credential obtained in step 2.
4. The authorisation server authenticates the client and validates the presented authorisation grant credential and if valid issues an access token.
5. The client presents the access token to the resource server and requests to access the protected resource.
6. The resource server serves the requests of accessing protected resource if the access token is valid.

This is one of the latest authorisation models which provides a way of authorising or delegating a third party to access a protected resource on behalf of the user without revealing the username and password. As noted previously delegation of authority is already implemented in PERMIS (Chadwick, et al. 2008) but it is not something we need for enforcing access control rules from the EU DPD.

2.2.8 Usage Control (UCON) model

The core of Usage Control (UCON) (Park and Sandhu 2004) family model is the $UCON_{ABC}$ model which integrates Authentication (A), oBligation (B) and Conditions (C). The UCON models consist of eight core components: subjects, subject attributes, objects, object attributes, rights, authorisations, obligations, and conditions, as shown in Figure 2-3. A brief description of the components is provided below.

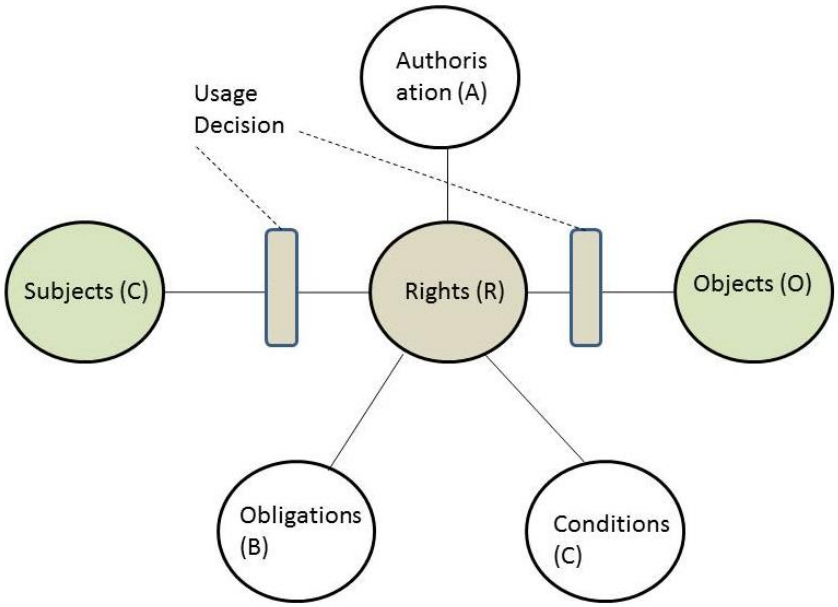

Figure 2-3. $UCON_{ABC}$ model components

In the UCON model a subject entity possesses certain rights on objects and is defined by subject attributes which represent the properties of that. Examples of subject attributes include identities, roles, memberships, security clearances and many more. Attributes can be of two kinds: an *immutable* attribute can only be changed by the administrative actions whereas a *mutable* attribute can be modified as a side effect of subjects' access to objects. In this model subjects can be of three types: consumer subjects (CS) who exercise rights to access objects, e.g. e-book reader; provider subjects (PS) who provide objects and holds certain rights on it, e.g. e-book author; identifee subjects (IS) who are identified in digital objects that hold their privacy sensitive data, e.g., a patient in a health care system.

An object is similarly defined by object attributes that can be used for access decisions. Examples of object attributes can be security labels, ownerships and so on. An object in this model is derivative if it is created in consequence of obtaining or exercising rights on an original object. For example, playing MP3 music files can create usage log information.

The privilege a subject can have on an object is defined by right. It consists of a set of usage functions defining subjects' access to objects. Rights can be of three types: consumer rights (CR), provider rights (PR) and identifee rights (IR). Usage decision function (see Figure 2-3) determines the rights based on subject and object attributes, authorisations, obligations and conditions.

Authorisations provide usage decisions evaluating subject and object attributes, requested rights and authorisation rules. Authorisation can be of two kinds: a pre-authorisation (preA) is performed before a requested right is exercised, and an on-going-authorisation (onA) is performed while the right is exercised which can be either continuously or periodically.

Obligations are functional predicates that verify the mandatory requirements a subject has to perform before or during a usage exercise. Obligations can be either pre-obligations (preB) or on-going-obligations (onB). A preB checks whether certain activities have been fulfilled or not; for example, checking a user has agreed to provide usage log information before listening to a music file. An onB has to be satisfied continuously or periodically while the allowed rights are exercised. An example of onB can be a user may have to keep watching certain advertisements while he is logged in.
Conditions are environmental or system-oriented decision factors which return Boolean values to indicate whether the relevant requirements are satisfied or not. It does not include subject or object attributes and its variables are not mutable.
The core $UCON_{ABC}$ model is classified based on the three criteria: i) decision factors that consist of authorisations, obligations, and conditions, ii) continuity of decision being either pre or on-going with respect to the access in question, and ii) mutability that can allow updates on subject or object attributes at different times.
Whilst this is a nice conceptual model it is not free from limitations. It neither supports multiple policy languages, nor does it cater for passing resources and policies among multiple parties which our system does with sticky policies. Furthermore, our system has a rich dynamic conflict resolution strategy allowing multiple parties to provide their own independent conflict resolution policies which make the system suitable for operating in a dynamically multi-authority environment and this feature is lacking in the UCON model. Furthermore, proper formulation of this model does not exist in the literature in any policy specification standard (Um-e-Ghazia, et al. 2012) and there is no public implementation of the model available to use. Long after our work was published, Ghazia et al. (Um-e-Ghazia, et al. 2012) suggested interpretation of the UCON model in XACML. However, the full realisation of the UCON model in XACML or in any policy language is still lacking.

2.2.9 XACMLv2 reference model

XACMLv2, the eXtensible Access Control Markup Language, is an XML based OASIS (Organisation for the Advancement of Structured Information Standards) standard language for expressing authorisation policies and a standard format for expressing queries over these policies. The XACML specifications were developed through a collaborative effort of OASIS members including IBM, Sun Micro systems and Entrust. XACML is the model that we use in our research, and PERMIS was enhanced to conform to the XACML model by the research in this thesis.
According to the OASIS XACML TC (Godik, et al. 2002) the data flow of the XACML reference model is described here.

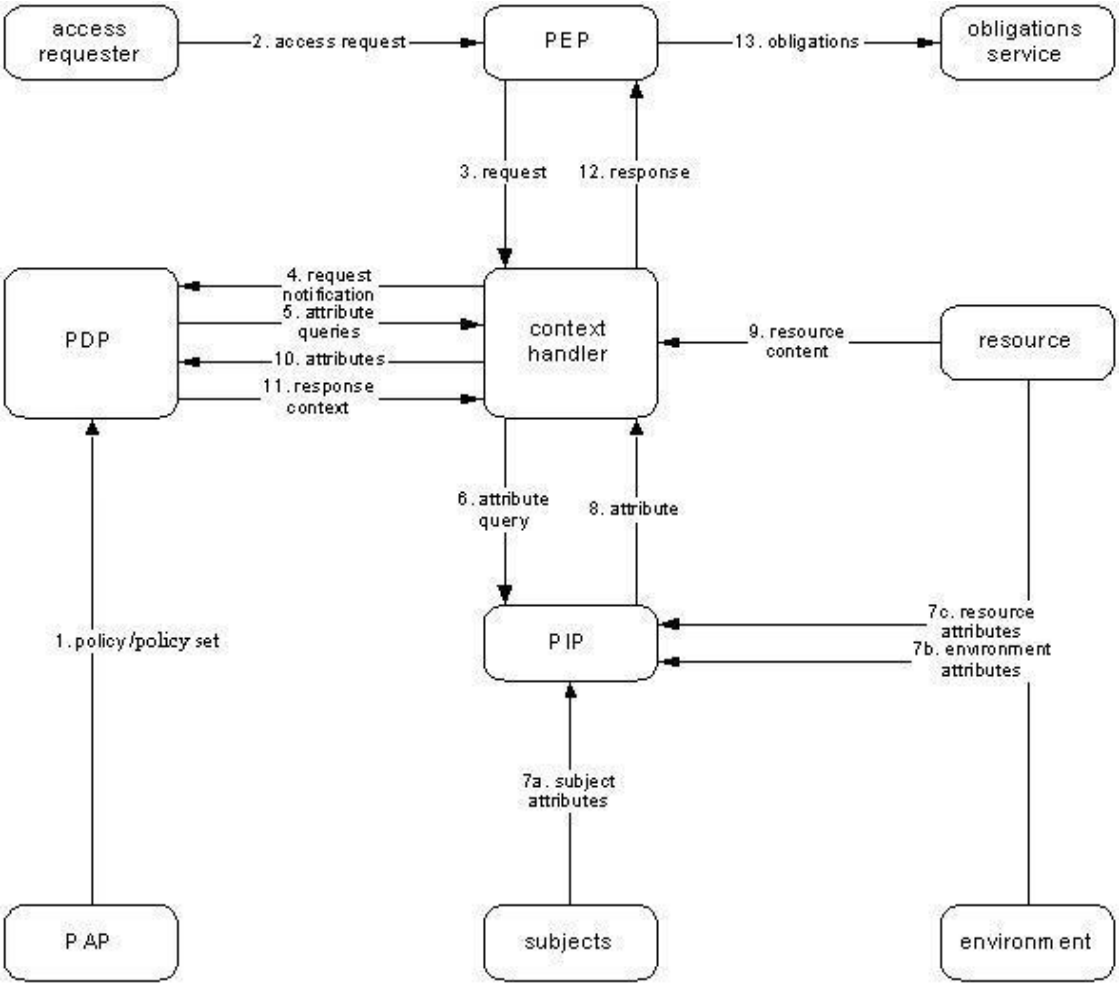

Figure 2-4. Data flow diagram of XACML

The data flow of the XACML model follows the steps described below

1. *PAPs (Policy administration points)* write policies and policy sets defining access rules for a specific target and make them available to the *PDP*.
2. The access requester (person/web service/ anything that needs access to the secured resource) sends a request for accessing the secured resource to the *PEP (Policy Enforcement Point)*.
3. The *PEP* sends the access request and optionally the *attributes* of the *subjects*, *resource, action* and *environment* to the *context handler* in its native request format.
4. The *context handler* constructs an XACML request *context* and sends that to the *PDP (Policy Decision Point)*.
5. The *PDP* requests any additional *attributes* of *subject*, *resource, action* and *environment* from the *context handler*.
6. On receiving the request from the *PDP*, the *context handler* requests the *attributes* from a *PIP (Policy information point)*.
7. The *PIP* obtains the requested *attributes*.
8. The *PIP* returns the requested *attributes* to the *context handler*.

9. Optionally, the *context handler* includes the *resource* in the *context*.
10. The *context handler* sends the requested *attributes* and (optionally) the *resource* to the *PDP*. The *PDP* evaluates the *policies* and makes an authorisation decision based on the policies and attributes.
11. The *PDP* returns the response *context* including the *authorisation decision* to the *context handler*.
12. The *context handler* translates the response *context* to the native response format of the *PEP* and returns it to the *PEP*.
13. The *PEP* fulfills the *obligations* which are the operations specified in a *policy* or *policy set* that should be performed by the *PEP* in conjunction with the enforcement of an *authorisation decision*.
14. (Not shown) The *PEP* either permits or denies access to the *resource* depending on the authorisation decision it obtains.

2.2.9.1 XACML policies and access request

XACML policies are constructed with the components: *PolicySet* (PS), *Policy* (P), *Rule* (R), *Target* (t), *Policy Combining Algorithm* (PCA) and *Rule Combining Algorithm* (RCA).

Target (t) is composed of four types of attributes: *Subjects*(S), *Resources* (Rs), *Actions* (A) and *Environments* (En).

Rule is defined as a tuple, r= (id, t, e, c), where id is a rule-id, t is a rule target, e ∈ {Permit, Deny} is an *Effect* and c is a *Condition* which is an optional element and is evaluated to a Boolean value.

Policy is defined as a tuple, p = (id, t, R, RCA, O), where id is a policy id, t is a policy target, R = $\{r_1, r_2, \ldots\ldots, r_n\}$ is the set of rules and RCA is the Rule combining Algorithm and O is an optional set of obligations.

PolicySet is defined as tuple PS = (id, t, P, PAC, O), where id is the PolicySet id, t is the PolicySet target, P= $\{p_1, p_2, \ldots\ldots, p_n\}$ is the set of policies, PCA is the Policy combining Algorithm and O is an optional set of obligations.

An access request in XACMLv2 is defined as a set of attributes, Rq = (s, rs, a, en), where s is a set of subject attributes, rs is a set of resource attributes, a is a set of action attributes and en is a set of environment or context attributes. Each attribute set describes the corresponding object. The request means that the subject described by 's' is requesting to do action described by 'a' on resource described by 'rs' in the presence of the context described by 'en'.

2.2.9.2 Evaluation of a request

The set of attribute names and values that *Subject*, *Resource*, *Action* and *Environment* elements of an XACML request contains are used in taking an authorisation decision. While evaluating a *Policy*, these attribute names and values of each category (i.e. *Subject*, *Resource*, *Action* and *Environment)* presented in the request context are compared with those presented in the *Policy* according to some criteria defined in the *Policy* (which are briefly presented next).

A *Target* (in a *PolicySet, Policy* or *Rule*) identifies whether the parent element is applicable to a given request. Boolean functions compare the attribute names and values found in a request with those included in the *Target*. If all the evaluation criteria of the Boolean functions of a *Target* are met, the *Target* element is evaluated as *Match* and its associated *PolicySet*, *Policy*, or *Rule* applies to the request. If any one of the elements specified in the *Target* is *Indeterminate*, the *Target* is evaluated as *Indeterminate*. Otherwise, it is evaluated as *NoMatch*.

If the *Target* of a *Rule* is evaluated as *Match* then the *Conditions* (if any) are checked. The strategy of *Rule value* evaluation is presented in Table 2.1. The *Policy* evaluation strategy presented in Table 2.2 and the *PolicySet* evaluation strategy presented in Table 2.3 are used to evaluate a *Policy* and a *PolicySet* respectively. The *Policy* and *PolicySet* evaluation also depends on the *Rule* and *Policy Combining Algorithm* which are explained next.

Table 2.1: XACML Rule evaluation

Target	Condition	Rule value
Match	True	Effect (i.e. Permit / Deny)
Match	False	NotApplicable
Match	Indeterminate	Indeterminate
NoMatch	----------	NotApplicable
Indeterminate	----------	Indeterminate

Table 2.2: XACML Policy evaluation

Target	Rule value	Policy value
Match	At least one rule value is Permit/Deny/Indeterminate	Effect (i.e. Permit / Deny/ Indeterminate) depending on Rule Combining Algorithm
Match	All rule values are NotApplicable	NotApplicable
NoMatch	----------	NotApplicable
Indeterminate	----------	Indeterminate

Table 2.3: XACML PolicySet evaluation

Target	Policy value	PolicySet value
Match	At least one Policy value is Permit/Deny/Indeterminate	Effect (i.e. Permit / Deny/ Indeterminate) depending on Policy Combining Algorithm
Match	All policy values are NotApplicable	NotApplicable
NoMatch	----------	NotApplicable
Indeterminate	----------	Indeterminate

2.2.9.3 Rule/policy combining algorithm of XACML

The standard *Policy Combining Algorithms* of XACML v2 (XACMLv2 2005) and v3 (XACMLv3 2013) are defined as:

- Deny-overrides (Ordered and Unordered in v3),
- Permit-overrides (Ordered and Unordered in v3),
- First-applicable and
- Only-one-applicable.

In the case of the Deny-overrides algorithm, if a single *Policy* element evaluates to Deny, then, regardless of the evaluation result of the other *Policy* elements, the combined result is Deny. For the Ordered Deny-Overrides the behaviour of the algorithm is the same except that the order in which the collection of *Policies* is evaluated will match the order as listed in the *PolicySet.*

Similarly, in the case of the Permit-overrides algorithm, if a single Permit result is encountered, regardless of the evaluation result of the other *Policy* elements, the combined result is Permit and for the Ordered Permit-overrides the behaviour of the algorithm is the same except that the order in which the collection of *Policies* is evaluated will match the order as listed in the *PolicySet*.

In the case of the "First-applicable" combining algorithm, the first decision encountered by the *Policy* element in the list becomes the final decision accompanied by its *Obligations* (if any).

The "Only-one-applicable" combining algorithm ensures that only one *Policy* is applicable by virtue of its *Target*. The result of the combining algorithm is the result of evaluating the single applicable *Policy*. If more than one *Policy* is applicable, then the result is Indeterminate.

In the XACML v3 some other combining algorithms are also defined, such as Deny-unless-permit (which returns Deny only if no Permit result is encountered, Indeterminate or NotApplicable will never be a result), Permit-unless-deny (which returns Permit only if no Deny result is encountered, Indeterminate or NotApplicable will never be a result)

2.2.9.4 Evaluation of obligations in XACML model

In XACMLv2 a *Policy* or *PolicySet* may contain one or more *Obligations*. In XACML v3 a *Rule*, *Policy* or *PolicySet* may contain one or more obligation expressions which are evaluated to *Obligations* when such a *Rule*, *Policy* or *PolicySet* is evaluated. In both XACMLv2 and XACMLv3 an *Obligation* is passed to the next level of evaluation only if the effects of the *Rule* (for v3), *Policy* or *PolicySet* set being evaluated match the values of the FulfillOn attribute of the *Obligation*. An *Obligation* will not be returned to the PEP if the *Rule* (for v3), *Policy* or *PolicySet* from which it is drawn is not evaluated. On the other hand, XACML does not guarantee to evaluate all the Policies. For example, if the *Policy Combining Algorithm* is Permit-overrides the execution of *Policies* will return as soon as it gets a Permit decision without executing the rest of the *Policies*. The *Obligations* along with the path of returning having the same effect will be returned with the decision. If there is any other *Policy* among the unexecuted *Policies* that return an *Obligation* will not be returned to the final decision. Suppose that, there are two *Policies*: one from the controller saying to Permit access with an *Obligation* to log the event and another one from the data subject saying to Permit access with an *Obligation* to e-mail the

author, and they are combined with the "Permit-overrides" combining algorithm. Due to the *Policy* evaluation strategy of the XACML PDP, either one of the *Policies* will be executed and the *Obligation* with that *Policy* will be returned depending on the order *they* are written in the *PolicySet*. Li et al. (Li, Wang, et al. 2009) have also addressed this limitation with XACML *Obligations*.

2.2.9.5 Limitations of XACML model

The XACML model is being widely used as an authorisation model but it is not a complete model for all kinds of authorisation. The model has the following limitations.

1. The XACMLv2 standard proposes a functional component called the Policy Administration Point (PAP) which is responsible for creating the policies and making them available to the PDP through some back channel prior to the PDP making its decisions. The back channel could be, for example, an API to an integrated database, or a communications link to an external repository. However, using a back channel and previously prepared policies is too static for some use cases. Consider the privacy protection of personal data, where a user's privacy policy is stuck to her personal identifying information (PII) Mont et al. (Mont, Pearson and Bramhall 2003 A). In this case the policy needs to be passed dynamically along with the decision request to the PDP. The data subject may not have or should not necessarily need access to the PAP for writing that policy for his/her private data. However, the SAML profile of XACML allows passing XACML policies dynamically to the PDP. Later, in Chapter 5, our authorisation server will use this protocol to dynamically add policies to the PDP, and we proposed (and got accepted) an enhancement to the SAML-XACML profile to allow policies in any language to be passed.

2. It assumes that all the PDP policies are written in the same language (XACML). It does not provide a way of integrating policies written in different languages. In reality many other policy languages are also available such as PERMIS (Chadwick, et al. 2008), P3P (W3C 2002), Keynote (Blaze, Feigenbaum and Ioannidis 1999) etc. Our advanced authorisation model is proposed which allows PDPs that support different policy languages to be integrated together.

3. The obligations are presented as attribute assignments and in XACMLv2 must be evaluated simultaneously with the user's action. Nevertheless in reality there are some obligations that should be implemented before the user is given access and some that should be implemented after and finally some that should be implemented along with the user's actions. In our design, the "before" obligations are evaluated by the P-PAAS server before the user is granted access. (Note, these different types of obligations are now supported in XACMLv3 enhancements).

4. The combining algorithms are such that they do not evaluate all the nodes in a tree. For example, the deny-(/permit) overrides return a decision as soon as it gets a Deny or Permit decision without executing the rest of the rules/policies. Consequently the obligations attached with the rest of the policies are also ignored. More details are provided in Section 3.4.3.2.

5. XACML provides no support for the use of credentials. Indeed credentials are never

mentioned in the XACML standard, even though remote users must present credentials in order to gain access to a resource. Our design incorporates credentials into the model so that they are evaluated as part of the process of granting the user access to a resource.

2.2.10 Privilege and Role Management Infrastructure Standards (PERMIS)

PERMIS (Chadwick, et al. 2008) is an authorisation system that provides access control decisions. It also has some other functionalities, for example, it provides a way of managing privileges during the process of taking access control decisions, it has a credential validation system which verifies users' credentials, and it supports delegation of authority which allows delegations of credentials between users. PERMIS uses the Hierarchical RBAC (or ABAC) model, in which roles are used to model organisational roles and the roles or attributes may be organised in a partial hierarchy where a superior role inherits all the privileges allocated to its subordinate roles.

2.2.10.1 Access control elements of PERMIS policy

Each PERMIS policy is identified by a unique Object Identifier (OID). The SubjectPolicy element specifies the subject domains. Only the users from the specified subject domains can be authorised to access resources covered by the policy. The TargetPolicy element of a PERMIS policy identifies the resource domains that the policy will protect. Target domains can be defined as either LDAP resources or resources formatted as HTTP-like URLs. The ActionPolicy element determines the actions recognised by the policy. It is also possible to restrict the target domains on which an action can be applied by including references to TargetDomainSpec elements inside the Action element. The TargetAccessPolicy consists of a list of Target Access Rules (TAR) and defines the access control rules. A TAR is represented as a TargetAccess element in the PERMIS policy and it contains TargetList, RoleList and optional conditions and Obligations. The TargetList specifies the actions allowed on a specified target. The RoleList element of a TAR limits the applicability of the TAR by specifying the roles a subject should have in order to be granted the specified permission. It is possible to limit the applicability of the TAR further by attaching a condition to the TAR which may contain a number of Boolean operations (AND, OR, NOT). Some other elements of a PERMIS policy include: the RoleHierarchyPolicy element, which specifies the hierarchy among roles, the SOAPolicy element, that defines which Source Of Authorities (SOA)s are trusted to issue roles to subjects, and the RoleAssignmentPolicy element, which specifies the rules for what roles can be allocated to which subjects by which SOAs.

2.2.10.2 Decision making in PERMIS

The PERMIS Java API receives a request through four different parameters:
- Subject: contains the valid roles of the subject
- Action: contains the requested action
- Target: contains the request target resource
- Environment: contains the environmental variables that are available during decision making

The first stage of decision making in PERMIS checks whether
- None of the given Subject, Action and Target is missing. If one of them is missing, an

exception is
thrown "[Indeterminate. Subject|Action|Target] is missing".
- The Subject is valid. If it is not, an exception is thrown "[Indeterminate. Unrecognised Subject Object]".
- The Target belongs to the target domain of the policy. If it does not, an exception, "[Not Applicable, TargetOutOfDomainException]" is generated.
- The Action is specified in the policy (by name and parameters) and whether the requested Target belongs to the target domain associated with the action. If any of these is not satisfied, an exception, "[NotApplicable-ActionNotInPolicyException]" is thrown.

When a request passes the first stage of decision making, it processes each TAR. The evaluation of a TAR works as follows:
- First, it is determined whether the requested action matches one of the allowed actions for the TAR. If it is not a FALSE is returned.
- Next, it is determined whether the requested target belongs to the domain of the TAR. If it is not a FALSE is returned.
- Next, it is determined whether the TAR's roles are subordinate to the subject's roles. If they are not a FALSE is returned (because the subject's roles are not sufficient).
- Once it has been established that the action and target match and that the subject's roles are sufficient, the condition, if any, is evaluated. If the TAR does not have a condition, a TRUE is returned. If the TAR has a condition, the condition is evaluated as follows:
    - If the TAR's condition evaluates to TRUE, then TRUE is returned.
    - If the TAR's condition evaluates to FALSE, then FALSE is returned.
    - If there is a problem evaluating the TAR's condition, an indeterminate exception is thrown.

2.2.10.3 Blacklist and Whitelist policy and their decision combination

A PERMIS PDP typically executes faster than an XACML PDP with an equivalent set of rules, since its policies are monotonic. Therefore, when a TRUE response is found no further rules are processed. However, having rules that only grant access makes the writing of some policies very inefficient e.g. everyone except X is granted access. Therefore, PERMIS also provides a Blacklist/Whitelist PDP combination. A Whitelist PDP contains the monotonic list of granted permissions. A Blacklist PDP contains the monotonic list of denied permissions. The same list of permissions can have two opposite meanings (granted or denied) by changing the PDP that interpreters it (from whitelist to blacklist, or vice versa).

Table 2.4: Example of PERMIS Blacklist and Whitelist policies

TargetAccessPolicy example of a Whitelist Policy.	**TargetAccessPolicy example of a Blacklist Policy.**
< X.509 PMI RBAC Policy OID=" WhitelistPolicy" > . . . <TargetAccess> <RoleList> <RoleType="permisRole" Value="Student"/> </RoleList> <TargetList> <Target><TargetDomain ID ="Library"/ > <**AllowedAction** ID =" access "/ ></Target>	< X.509 PMI RBAC Policy OID=" BlacklistPolicy" **DenyBased=" true "**> . . . <TargetAccess> <RoleList> <RoleType="permisRole" Value="Student"/> </RoleList> <TargetList> <Target><TargetDomain ID ="Laboratory"/>

</TargetList> </TargetAccess>	<**DeniedAction** ID ="access "/></Target> </TargetList> </TargetAccess> . . .

The Whitelist policy PDP returns a Grant only if an access is specifically granted by the policy, otherwise it returns a Deny. This kind of decision mechanism is suitable for an organisation that requires all kinds of access to be denied by default unless specifically granted. By setting a special parameter "Enable Not Applicable" the default Deny decision can be stopped. In this case if all the TARs evaluate to false, instead of a Deny the Whitelist policy PDP will return a NotApplicable decision.

The Blacklist policy PDP behaves in the opposite way to the Whitelist Policy PDP. It returns a Deny if a TAR evaluates to true and returns a Grant if all the TARs evaluate to false. This is suitable for a security set up which defines the situations or conditions on which access should be denied and it allows access if all of these denying conditions are false. If the Blacklist policy PDP is configured with the "Enable Not Applicable" parameter it does not return a Grant when all the TARs evaluate to false, rather it returns NotApplicable. In PERMIS the Blacklist PDP is evaluated first and then the Whitelist PDP. How the decisions of Blacklist and Whitelist PDPs are combined is given in Table 2.5.

The Whitelist policy presented in Table 2.4 says, "All students can access the Library" and the Blacklist policy says, "All students cannot access the Laboratory". In this situation if a staff tries to access the Library, the Blacklist PDP will return either a NotApplicable or Grant and the Whitelist policy will reply either a NotApplicable or Deny depending whether the "Enable Not Applicable" option is set or not. According to Table 2.5 the final decision will be either NotApplicable or Deny. If on the other hand staff is defined as a superior role to student, then staff can enter the library and the final decision will be Grant. If a student tries to access the Library the Blacklist PDP will return a NotApplicable or Grant (depending on the "Enable Not Applicable" option is set or not) and the Whitelist PDP will reply a Grant and the final decision will be Grant. If a student tries to access the Laboratory, according to Table 2.5, the Blacklist PDP will return a Deny and this will become the final decision without calling the Whitelist PDP.

Table 2.5: Black List and White List PDP result combination for PERMIS

Black List PDP Response	White List PDP Response	Returned Result
Deny	Not called	Deny
Indeterminate	Not called	Indeterminate
Grant/Not Applicable	Grant	Grant
Grant/Not Applicable	Not Applicable	Not Applicable
Grant/Not Applicable	Indeterminate	Indeterminate
Not Applicable	Deny	Not Applicable
Grant	Deny	Deny

2.2.10.4 Conflict resolution in PERMIS

PERMIS does not explicitly support conflict resolution rules since it operates with the implicit rule of

1. "all actions are denied except those specifically allowed" (for a Whitelist PDP policy only). This is equivalent to a permit-overrides conflict resolution rule of XACML and
2. "all actions are allowed except those specifically denied" (for a Blacklist PDP policy only). This is equivalent to a deny-overrides conflict resolution rule of XACML.

2.2.10.5 Limitations of PERMIS

1. It does not allow any arbitrary attribute to be specified for resources and subjects in condition statements (IF clauses). Only the resource type and role are supported. Other resource/subject attributes have to be evaluated as environmental attributes in condition statements.
2. It does not have explicit conflict resolution rules as implicit ones are built in.

2.2.10.6 Advantages of PERMIS

1. Since it does not mix a grant and a deny rules together, its rules are monotonic. Thus when the first rule denies (for a blacklist PDP) or grants (for a whitelist PDP) access, processing can stop and a definitive reply can be given. This means it performs much faster than an equivalent XACML PDP (Chadwick, Su and Laborde 2008)
2. The monotonic nature of the policy rules enables administrators to inspect policies and understand which subjects are being granted access to which resources much more easily. This can be almost impossible to do with complex XACML policies where rules can contradict one another.

2.3 Privacy Requirements and Privacy Enhancing Technologies

While designing a privacy protecting authorisation system it is important to gain a clear understanding of the privacy requirements. In this section the requirements for a system providing privacy protection have been presented. Furthermore, a thorough comparative study of the privacy enhancing technologies has been performed to identify the strength and weakness of them in meeting the requirements.

2.3.1 Requirements for a system providing privacy protection

Privacy protection of personal data is an important Legal requirement for organisations handling electronic private data. Many organisations need to collect personal data for business, promotional, research and operational purposes. These organisations need to ensure the privacy of these data, as now a day people are more concerned about the privacy of their personal data. Many laws exist to support the protection of personal data (Fischer-Hubner 2001). OECD (OECD 1980) first introduced the guidelines for the protection of privacy of computerised data. The EU and UK data protection acts (Directive 95/46/EC 1995, Data protection act 1998) specifies the requirements for privacy protection, which are as follows:

1. Purpose specification: While collecting personal data the purposes for which the data are

being collected should be stated. Once collected they should be associated with the personal data to ensure that they are only used for those purposes.

2. Consent specification: There should be a mechanism for obtaining and associating consent with the personal data so that the consents can be checked while taking an authorisation decision for accessing the personal data.

3. Limited collection: Minimum data should be collected for a defined purpose.

4. Limited use and limited disclosure: Only the requests that are consistent with the purposes and consents associated with the personal data should be allowed.

5. Limited retention: When collecting personal data the data subject should be able to state how long the data will be kept, and the data should be removed after that.

6. Accuracy: The collected data should be stored accurately.

7. Safety: The collected personal data should be kept safely so that no leakage can occur.

8. Openness: The subject of personal data should be able to view the data about him/herself.

9. Compliance: It should be possible to verify the compliance of the privacy protection offered by the enterprise with the rules of law.

10. User's control: The user should have control over the information he/she provides such as the ability to request an update or blocking of processing or erasing the data.

11. Enforcing privacy obligation: Privacy obligations should be enforced such as notifying the data subject when his/her data are accessed.

12. **Privileged access:** The EU DPD has defined privileged access to personal data by certain parties for some purposes, e.g. Medical Professionals can access medical data and so on.

13. Transferring data: The EU DPD has defined certain restrictions on the cross border flow of personal data. An authorisation system should be able to address that issue.

14. **Contract based access to personal data:** The EU DPD has defined certain access rights based on the contract, and an authorisation system needs to deal with that condition.

Other than the above requirements we consider that some additional requirements would help to protect privacy, which are as follows:

15. Simple user interaction: A system will be useless if the users cannot use it comfortably. The privacy preservation mechanism should provide the users with easy tools so that the interactions remain very simple.

16. Multiple policy language support: When a data subject (or an issuer) of any personal data provides a policy written in one policy language and later the data, along with the policy are transferred to another controller's system, the full enforcement of the policy to protect access to that personal data will only be possible if the receiving system also supports the compatible policy language. If the received policy language is not supported by the receiving system then the receiving system has two options- it can either translate the policy into the language it understands or discard it. Translation of a policy from one language to another is a very daunting task and even if the policy can be converted the meaning of it can change due to conversion as the features of different policy languages are not always the same. In a distributed environment we cannot expect that all policies will be written using one policy language when so many are available and there is no ubiquitously accepted standard for privacy policy languages. Consequently we need an authorisation infrastructure that is capable of evaluating multiple policies written in multiple languages.

17. Distributed enforcement: The policies that are being used to protect a personal data item should also be used when the data are moved to another system.
18. Inclusion of policies from law with the highest priority: Law enforcement would be much easier if the authorisation system could include the authorisation policies from the law with the highest priority so that the rights provided by the law can always be executed.

2.3.2 Review of privacy enhancing technology

The UK Information Commissioner's office has defined Privacy Enhancing Technology (PET) as technologies that protect or enhance an individual's privacy under the Data Protection Act 1998 (Data protection act 1998). Previous research on privacy enhancing technology related to access control is reviewed here.

2.3.2.1 Privacy research of IBM

IBM has performed research on privacy protection of customers' data collected by enterprises (Karjoth, Schunter and Waidner 2002, Karjoth, Schunter and Waidner 2003, Karjoth and Schunter 2002, Nelson, McCullough and Bliss 2005, Schunter and Berghe 2006). The authors have mainly tried to ensure that the privacy that is promised to the customer while collecting data is actually implemented. They have used the sticky policy paradigm where personal data are associated with the privacy policy and passed together while exchanging data among enterprises. The IBM Enterprise Privacy Architecture (EPA) (Karjoth, Schunter and Waidner 2002) is a methodology for enterprises to provide privacy. The EPA presents privacy awareness and privacy services into enterprises in a structured way and supports data subject's consent management on a per person basis. The consented policies are associated with the collected data and thus support the sticky policy paradigm.

The Platform for Enterprise Privacy Practices (E-P3P) (Karjoth, Schunter and Waidner 2003) is a model for describing enterprises' access control policies to provide privacy to customers' data. The authors show how personal data are collected; and also introduce separation of duties of different policy administrators. The privacy officer is responsible for defining privacy policies while the security officer is responsible for defining access control policies. A consent management paradigm provides greater control for the customers with regards to their personal data.
Karjoth et al. (Karjoth and Schunter, A Privacy Policy Model for Enterprises 2002) describe a privacy policy model. By extending the Flexible Authorisation Framework the authors have created a privacy control language that includes user consent, obligations and distributed administration. In this model a group hierarchy is presented, where a group inherits all the permissions from its ancestors. Data are categorised depending on their linkage to the data subject. This model introduces a hierarchy of purposes along with conditions and obligations. The privacy policy model is described in the logical framework of authorisation specification languages. A solution to automatically translate the inner-enterprise privacy policy written in E-P3P to publishable policies for customers in P3P is provided in (Karjoth, Schunter and Herreweghen 2003). The language has been formalised and refined to form the IBM Enterprise Privacy Authorisation Language (EPAL) (Ni, et al. 2007).

Nelson et al. (Nelson, McCullough and Bliss 2005) have focused on the need for privacy

protection in distributed systems. The authors have described the various steps of collecting and using personal data in a distributed environment and discussed how to provide more trust in those steps. This paper only considers different issues but does not show any practical implementation or uniform solution. Schunter et al. (Schunter and Berghe 2006) have discussed different features of practical implementations and enforcement of the sticky policy paradigm. They have introduced the idea of privacy injector to add privacy enforcement to existing applications, which is comprised of two parts- the privacy meta-data tracking part and the privacy policy enforcement part. The first automatically assigns, preserves and updates a privacy policy and the second enforces the policy. They have described a life-cycle of personal data and a way to protect the privacy of personal data throughout its life-cycle.

The IBM research on privacy has not provided a way to accommodate different policy languages. In a distributed environment, it cannot always be assumed that all the PDPs will use the same language. They have not provided a secured way of transferring data between enterprises or a provision for a mechanism to verify a signed message in the system. Furthermore, the obligations they present are just activity names such as log, notify, getConsent etc. (Karjoth and Schunter 2002). Conditions are provided to start and end obligations (Karjoth, Schunter and Waidner 2003), however they have not provided a way to practically enforce these. Moreover, they did not consider including the Legal authorisation policies.

Review of the model:

- When data are accessed by a user to perform an operation for a particular purpose that purpose must match the conditions described in the consent associated with the data.
- The data subject is given opt-in and opt-out choices for the privacy policy that governs the usage of data.
- Does not ensure limited collection but assumes that the enterprise is trusted and so would minimise the data collection.
- Checks whether a request by a user for a data item is authorised by the data subject and thus it limits the uses and disclosure and the data subject gets control over the access.
- Does not specifically ensure that the data subject can access the data. As the data subject's policy is protecting the access to the personal data it can be assumed that the data subject can access his/her personal data by his/her policy.
- A future work is mentioned to store accounting information in the privacy metadata so that all the operations performed on a data can be observed and thus it will be possible to verify the protection offered.
- Obligations are returned as activity names.
- Allows distributed enforcement through sticky policies.

2.3.2.2 Privacy-Aware Role-Based Access Control (P-RBAC)

Ni et al. (Ni, et al. 2007) have defined the Privacy-Aware Role-Based Access Control (P-RBAC) model to support privacy related policies which are expressed as Permission Assignments (PA). Similar to RBAC in P-RBAC permissions are also assigned to roles and roles are assigned to users. However, the structure of a privacy permission in P-RBAC differs from the permission in RBAC. Along with the data and action it explicitly states the intended purpose, condition on

permission and obligation. The privacy policy is combined with the existing access control model and the formal definition of privacy-aware permission is also provided. Like RBAC the family of conceptual models of P-RBAC consist of core P-RBAC, hierarchical P-RBAC, and conditional P-RBAC. The hierarchical P-RBAC model introduces the notions of role hierarchy, object hierarchy and purpose hierarchy. The conditional P-RBAC model allows the policy writer to specify relations between different permission assignments; the notions of conflicting permission assignments and conflict detection algorithms are also provided. When a conflict is detected feedback is provided to the policy author to modify or discard the policy. Constrained natural language policies from policy authors are written with the SPARCLE policy workbench (which is a tool for authoring policy) and are transformed into P-RBAC permissions. Although these models theoretically associate data permissions with purpose, condition and obligation, the authors admit that they are too complex to implement practically. Qun Ni et al (Ni, Bertino and Lobo 2008) have defined the obligation model for P-RBAC, and have mentioned that the traditional policy based access control approach is not good enough for enforcing obligations as it can involve time interval, conditions as well as actions. The authors have also mentioned the features related to obligations. A language for specifying and handling obligations is proposed in (Li, Chen and Bertino 2012). These papers are discussed in detail later in Section 2.4.

Review of the model:

- Uses purposes with the permissions. The data subject can assign permissions and those are matched with the requests and thus the data subject gets control over the access.
- Shows a way to write natural language policy rules by the user using a tool, thus policy rules from the data subject can work as consents.
- Checks whether a request by a user for a data item is authorised by the permission and thus it limits the uses and disclosure.
- With the enforcement of obligation it is possible to ensure that the data are deleted after a certain event occurs.
- Does not address that data subjects can access the data. However, by assigning permissions it is possible to ensure that.
- For enforcing obligations an obligation language is specified.

2.3.2.3 Privacy research of HP

HP has also worked on providing privacy of personal identity information by enforcing obligations. Mont (Mont 2004) has listed the different aspects that are required to deal with privacy obligations such as: validity period, event that triggers the obligations, enforceability of the obligation, target of the obligations and so on. He identifies the important issues that need to be considered for the management and enforcement of privacy obligations such as: modelling and representing privacy obligations, association of obligations to data, mapping obligations to data to name a few. Furthermore, this paper gives a high level design of the obligation management systems and provides a way of transmitting encrypted confidential data with obligations to other parties. Nevertheless, it only describes the obligations related to privacy and does not provide any uniform solution for integrating access control policies of the organisation with the privacy policies (set by the data subject). Mont et al. (Mont, Pearson and

Bramhall 2003 A) have proposed a way to obfuscate personal information to protect its content. Obfuscation of data is done using the sticky policies as the IBE encryption keys. Alteration of that key makes it impossible to generate the decryption key. The sticky policies are then associated with the obfuscated data. The receiver of the personal data needs to get the decryption key from the trusted authority (TA) and provides information to the TA as required by the disclosure policy. The trusted authority issues the decryption key only if the requester acknowledges compliance to the disclosure policies. Mont et al. (Mont and Thyne 2006) have described a privacy policy enforcement model with the technical details. In this model the requester's intent is checked against the purpose provided by the owner of the personal data to enforce privacy aware access control. No direct access to the databases is allowed in this model and the queries are analysed before using them for accessing databases. A working prototype has been integrated with HP Select Access, software for providing policy-based authentication and authorisation to web-based applications and web services. Mont et al. (Mont and Beato 2007) have presented an obligation management system that automatically derives related obligation policies from the privacy preferences provided by the users. This paper is reviewed in detail in Section 2.4.

Review of the model:

- Uses purposes with the preferences and the purposes are checked for authorising access.
- Consent is taken in the form of preferences.
- Checks whether a request for a personal data item is authorised by the preferences of the data subject and thus it limits the uses and disclosure.
- Accommodates a mechanism for obligation enforcement which also ensures data deletion after the occurrence of certain event.
- Supports distributed enforcement by using a sticky policy paradigm.

2.3.2.4 Privacy research of EnCoRe

The Ensuring Consent and Revocation (EnCoRe) project (EnCoRe 2009) has worked towards the means for individuals to retain control over their personal data by providing and managing consent and revocation. By giving consent an individual gives permission to use personal data for specific purposes, under certain conditions. By performing revocation, on the other hand, s/he determines how personal data should be handled once it is disclosed. The key requirements and practical implications of handling consent and revocation are discussed in (Mont, et al. 2009). Consents define the privacy preferences and constraints on the personal data, how data should be used such as for a specific purpose and any obligation to be applied and so on. Revocation is a process by which an individual can modify or invalidate a previously given consent. A reference model for the management of consent and revocation is also provided. A consent and revocation policy is presented in (Agrafiotis, et al. 2010), which defines what consent preference a user can express and in what way he / she can revoke it. The privacy requirements and the related policies from different sources, such as legislation, organisation guidelines, user expectations and so on, are considered (Papanikolaou, et al. 2010). The authors have tried to find an intermediate representation of all the policies having different levels of abstraction so that it can embody all high level requirements and be translated into low level (machine executable) policies. Such a representation consists of syntax for conditions that need to be checked, syntax for immediate actions that need to be

performed if the conditions of a particular rule are met and syntax for obligations which the enterprise has if the given conditions are met. In (Mont, S. Pearson, et al. 2010) the authors have specified a hierarchy of policies based on the level of abstraction. Privacy policies of the laws and regulations constitute the top most layer since the requirements in these are presented in the most abstract form. In this hierarchy the low-level policies describe how privacy requirements are implemented; executable policy languages such as XACML belong to the lowest-level in the policy hierarchy. All the upper level policies eventually need to be translated into low-level policies for enforcing them. The preference of the data subject itself is treated as personal data. To protect access to the preferences of the data subject a complete separation of the decision making part of the authorisation system from the data access part is proposed in (Kounga, Mont and Bramhall 2010).

Review of the model:

- The user is given the option to modify or completely remove the previously given consent which gives more control to the data subject over his/her personal data.
- Provides a way for verifying the compliance of the policies with the requirements.
- Accommodates mechanism for obligation enforcement.
- Plans to provide a user friendly interface in future.
- Conceptually includes access control policies from laws and regulations. However, it is still lacking the practical implementation of the policies from law and they did not specify how these policies will be combined with the preferences of data subject and the organisation's policy.

2.3.2.5 Platform for Privacy Preferences (P3P)

P3P (W3C 2002) has defined a machine interpretable format for websites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by user agents. The notice and consent model of P3p allows websites to describe their privacy policies and users can read these and can choose to interact with the websites and thus provide their consent to the policies. P3P provides a human readable version of the policies, automated comparison of users' preferences with the policies, reports if there is a mismatch as well as users' preferences for viewing and changing policies. The limitation of P3P is that it just checks whether the users' preferences match with the organisation's stated privacy policies, it does not ensure whether the organisations actually enforce their stated privacy policies. It is notable that E-P3P (mentioned in Section 2.3.2.1) formalizes internal privacy practices of an enterprise while P3P formalizes advertised privacy promises.

Review of the model:

- Can use purposes with the preferences and the preferences along with purposes are checked with the organisation's stated privacy policies.
- Consent is taken in the form of preferences.
- Checks whether a user's preferences match with the organisation's stated privacy policies. The data subject can assign, modify or completely remove preferences, but this model does not ensure the enforcement of the policy inside the enterprise and so does not actually provide control to the user.
- Provides a way for verifying the compliance of the policies with user preferences. But does not provide a mechanism for verifying the compliance with the data protection requirements.

- Does not accommodate a mechanism for obligation enforcement.
- Provides a simple interface for obtaining users' preferences.

2.3.2.6 Primelife Policy Language (PPL)

PPL (Trabelsi, et al. 2010, Trabelsi, Sendor and Reinicke 2011, Ardagna, et al. 2009) has extended the XACML v2.0 language to provide a similar syntax to express preferences of the data subject, controller's policy and the sticky policy agreed upon by the data subject and controller. The data handling policy of the controller expresses how the information collected from the data subject is handled, the data handling preferences of the data subject specify how the he/she wants his/her data to be treated after they are collected. An applicable sticky policy is derived from the matched portion of the data handling policy and preferences and this represents the policy upon which both parties, i.e., the controller and the data subject, have agreed. It also extends the XACML model to have a credential handler. The architecture of PPL consists of the following three layers. The Presentation Layer consists of a policy editor and a matching handler, the Core Layer consists of a Policy Enforcement Point (PEP), Policy Decision Point (PDP), Credential Handler and Obligation Handler and the Persistence Layer handles the data and policy store and credential store. A new obligation handling mechanism including temporal constraints, pre-obligations, conditional obligations and repeating obligations is proposed. However, it does not support policies from multiple authorities in multiple languages, does not include policies from law with the highest priority, does not have dynamic conflict resolution strategy and credential or contract validation. Review of the model:

- Uses preferences from the data subject and purposes may be added in the preferences.
- Consent is taken in the form of preferences. The model checks the preferences of the data subject with the policy of the data controller.
- Checks whether a user's preferences match with the organisation's stated privacy policies. The agreed policy of the data subject and controller is used as a sticky policy and that is enforced; which limits the uses and disclosure of data and thus the users are provided a bit of control over the personal data. But whether the data subject can modify or completely remove preferences is not stated.
- Has separate storages for data, policy and credential but does not provide details of the mechanism for providing safe storage of them.
- Provides mechanism for obligation enforcement.
- Provides interface for getting user's preferences.

2.3.2.7 PuRBAC: Purpose-Aware Role-Based Access Control

PuRBAC (Masoumzadeh and Joshi 2008) has attempted to extend the RBAC model to include the *purpose* as a separate and central entity. It has increased the capability of the RBAC model to include privacy requirements while reducing the complexity of the P-RBAC (Ni, Trombetta, et al. 2007) model. The focus of the model is that the subject should specify the purpose of access in its request while accessing a piece of data. The hierarchy of role and purpose has also been considered. A user gets permission to access a piece of data only if the purpose of access is allowed for the currently active role. This model is mainly designed for an organisation where different levels of employees get access to the personal data suitable for their roles.

The model does not facilitate the data subject to specify for what purposes who he/she allows to access his/her data.

Review of the PuRBAC model:

- Uses a hierarchy of purposes and based on different active roles of a requester the purposes are allowed. The purpose specification is enterprise centric rather than user centric i.e. the organisation specifies what role in the organisation is allowed to access data for what purposes, and the data subject does not specify for what purposes who (a role or another organisation) can access the data.
- Only the authorised role can access the personal data for the authorised purposes and the user has to satisfy the conditions to get the permission to access. Hence the system provides limited use and disclosure.
- Allows limited retention through the use of obligation to delete the data after a certain amount of time.
- Has a mechanism to enforce pre and post obligations.

2.3.2.8 Privacy model of Al-Harabi and Osborn

Al-Harabi and Osborn (Al-Harbi and Osborn 2011) have provided a model based on the Role Graph Model (RGM). They include purpose with the permission assignment. Roles in this model consist of a role name and permissions which are arranged in a hierarchy called the Role-Graph. They provided a simple model to include purpose in the RGM model to provide privacy but their work does not accommodate many aspects of privacy.

Review of the model:

- Accommodates purposes with the permission assignments. Only the permission is activated (for a role) for which the access purpose is equal to or more general than the intended purpose. This purpose in a way is an indirect consent.
- Only the authorised role can access the personal data for the authorised purpose. Hence the system provides limited use and disclosure.
- How the model will attempt to ease the user interactions is not mentioned.

2.3.2.9 Access control model of Byun and Li

Byun and Li (Byun and Li 2008) have provided a purpose based access control model for privacy protection based on the RBAC model. This model allows multiple purposes to be associated with each data item and also accommodates explicit prohibitions i.e. the purposes for which the data should not be used. It also specifies different levels of granularity of data i.e. the units of data with which purposes can be associated in the context of relational databases.

Review of the model:

- Accommodates intended purpose i.e. the purposes for which access is allowed and the access purposes i.e. the purposes for which access is requested.
- Allows explicit prohibition, purpose compliance and granular association of purposes.
- Does not support distributed enforcement in its reviewed version.

2.3.2.10 Trust and privacy model of Smari et al.

Smari et al. (Smari, Zhu and Clemente 2009) (Zhu 2008) have proposed a model to incorporate trust and privacy in attribute based access control for collaborative environments. The access

control model is based on attributes of subjects and objects. They provide algorithms for calculating trust levels and assigning trust based subject attributes. They provide six aspects of privacy along with purpose matching.
Review of the model:

- Incorporates purpose matching associated with an object but does not mention how to obtain the purposes.
- Allows purpose compliance. Unauthorised users are prohibited from knowing the existence of an object; an authorised user can access only the minimum necessary views of the object.
- Plans to limit access time according to the necessity of a subject to fulfil a purpose.
- Aims to allow authorised user to disclose or know usage status of an object when necessary.
- Whether the user can change the purpose later, and how the purposes can be passed to another organisation are not specified.
- Has no mechanism for obligation enforcement in the reviewed version.

2.3.2.11 Privacy enhanced access control model of Xu et al.

Xu et al. (Xu, et al. 2009) (Xu, He, et al. 2010) have proposed an enhancement of the traditional access control matrix model to integrate privacy; as that only presents subjects and objects. The enhanced model has a third dimension representing a set of privacy-concerning subjects. A privacy access right is an entry in the matrix that indicates a particular access right along with conditions that determine whether the access right is authorised. The conditions presented are very simple which may not be appropriate for a complex scenario.
Review of the model:

- Does not specify purpose matching.
- Incorporates conditions for which the access right is authorised. One of such conditions is 'ask', which means the authorisation of the privacy concerning subject is needed. The model does not provide the semantics to obtain real time authorisation from the subject.
- Allows checking the conditions of the privacy concerning subjects and thus limits the disclosure and the subjects are given control over the access to the data.

2.3.2.12 Policy-based Privacy Authorisation System (PPAS)

In PPAS (Choi, Lee and Lee 2006) privacy is maintained in a policy based authorisation system. In this simple system two kinds of policy are maintained, one is for every data item in the system which is set by the administrator and another is set by the owner of the resource or data. A request for data access comes with three elements- subject (the requester), resource (consists of resource and the owner) and action (consists of access mode i.e. read, write, create, delete and purpose). The requests are written as an XACML request context. Policies are stored and retrieved in an Oracle 9i database, and a simple conflict resolution policy is also implemented. The system is suitable for a single organisation, not for a distributed environment. Review of the model:

- Uses the purpose as one of the elements of privacy policy.
- Accepts policies from the owners of personal data and those policies work as consents of the owners.

- Checks whether a request can be granted by the privacy policy. So it provides limited use and disclosure.
- Does not ensure limited retention specifically, but with an obligation it may be possible to provide it.
- Whether the data subject can modify or completely remove the given preferences is not stated.
- Includes obligations as a part of the privacy policy, although does not provide details on their enforcement.

2.3.2.13 Privacy policy enforcement system of Goyal et al.

Goyal et al. (Goyal, Deodia and Gupta 2007) have proposed a system for privacy aware access control with RBAC. The authors have implemented the system using the PHP scripting language. The privacy policy is enforced by associating intent with the access control policy. An authorisation table stores the policies and another table stores the roles. Policies are grouped into policy sets according to the purposes, and when an access request comes in, the policy set is extracted according to the purpose. Obligations for the policies are also handled. A policy is modelled to have fields of purpose, attribute, target type, requester type, condition, obligationID. An obligation is presented with fields of Obligation-ID, Obligation_Name, Obligation_function, Ob_Argument_list. The tables of the system have a fixed number of fields, hence are restricted for modifications. This model is not suitable for distributed environment.

Review of the model:

- Uses the purpose as one of the elements of privacy policy.
- Assumes that the user account and information management module are responsible for the consent management.
- Checks whether a request is granted by the access control policies. Thus it provides limited use and disclosure.
- Separates the data handler so that no direct query can be made to the table.
- Includes obligations as a part of the privacy policy but does not provide details on their enforcement. Each obligation has Obligation-ID, Name, Function and Parameter, but does not specify various types of obligations.
- Accepts policies from the users but does not give details on how these policies are obtained from the user.
- Does not support distributed enforcement of privacy policies.

2.4 Use of Obligations to Protect Privacy

Obligations are the actions that must be performed when an event occurs. When the event is an authorisation decision the obligations are the actions that must be performed before, after or along with the enforcement of the authorisation decision.

Enforcement of privacy obligations is an important part of ensuring the privacy of personal data. Systems that want to protect the privacy of personal data must provide a way for the proper enforcement of privacy obligations which a traditional access control system is unable to do. Some privacy related obligations can be, for example, sending e-mail to the subject of

the personal data when the data are accessed or deleting the personal data after a certain amount of time. Many research works have been published related to the handling of obligations and are discussed in this section.

Randic et al. (Randic, Kunstic and Blaskovic 2004) have described how the operations of PEP can be improved by using an "object by value" transfer mechanism for loading objects representing obligation policies. The authors describe the policy based management system of the IETF policy framework which is composed of a Policy Management Application (PMA), Policy Repository (PR), Policy Decision Point (PDP) and Policy Enforcement Point (PEP). It is assumed that the policies are stored in a LDAP directory as policy objects and are transferred from the repository to the PDP. The PDP receives policy requests and returns policy decisions to the PEP which enforces the policy i.e. it performs the action according to the decision returned by the PDP. Obligation policies are events triggered that specify what action a subject must perform on an object. State information is transferred from the PDP as a local creation of an object and with that the PEP creates the policy enforcement object. The authors have claimed that the *object by value* technology in policy based management architecture has satisfied the adaptability requirement of agents. They specified policies in the Ponder policy language; however, this language does not support the pre-obligations, which is equivalent to our temporal type "before" (Chadwick and Fatema 2009).

Ni et al. (Ni, Bertino and Lobo 2008) have defined an obligation model for Privacy- aware Role Based Access Control (P-RBAC). The authors have mentioned that the traditional policy based access control approach is not good enough for enforcing obligations as that can involve time intervals, conditions and actions. Obligations are defined as actions that some subjects have to fulfill during some time intervals and the time interval is specified by a temporal constraint component. They have defined the pre-obligation as one that should be fulfilled before an action and the post obligation as one that should be fulfilled after the action. If the condition of a conditional obligation cannot be fulfilled, the obligation will not be satisfied. In the repeated obligation, an obligation will be attempted to be fulfilled at most 'n' times. Obligations are presented in tuples of conditions, users, actions, objects and temporal constraints where the temporal constraint is a tuple of start time, end time and count (number of times the obligation is to be fulfilled). Obligations cascading is defined as a phenomenon where the execution of an obligation can trigger the execution of another obligation. The term "dominance" refers to the situation where an obligation dominates another obligation. It is stated that the model is not implemented yet due to the complexity of obligations in privacy policies, and they are working on it.

Mont (Mont 2004) has documented different aspects that are needed to deal with privacy obligations such as validity periods, trigger events, enforceability, target and so on. Important issues that need to be considered for the management and enforcement of privacy obligations are such things as modelling and representation, association and mapping to data and so on. Furthermore, this paper presents a high level design of the obligation management system. It has also provided a way of transmitting encrypted confidential data along with the obligations to other parties.

The obligation management system presented in (Mont and Beato 2007) automatically derives the related obligation policies from the privacy preferences provided by the users. Here an obligation policy template is introduced which is instantiated by replacing the place-holders with the actual privacy preference value to provide flexibility in defining the privacy obligation policies. Each piece of data is associated with one or more "instances" of obligation policies. In this approach, a large quantity of data needs to be dealt with and a parametric obligation policy is introduced to make the system scalable. The key feature of this is that the privacy preferences are stored separately from the policies which provide a way to apply the policy to a potentially large set of personal data. The explicit reference to the privacy preference is stored elsewhere. The parametric obligation policy is presented as a tuple of unique identifiers, target, parametric events that trigger an obligation, parametric action and parametric "on violation" actions which are executed to re-mediate any violation enforced by obligations and is represented as an XML formatted reactive rule. This paper has also described the Scalable Obligation Management System, which can manage both the parametric obligations and traditional obligations. In summary, this model mainly focuses on the scalability problem when dealing with a large amount of private data, each having different privileges and user consents.

In (Ananthanarayanan, Mohania and Gupta 2005) the detection and resolution of conflicts amongst conflicting obligations are discussed. An obligation is defined as a task, optionally associated with some rules specifying an action that needs to be mandatorily performed after the user's request is fulfilled. Obligations are classified as notification related and retention related and conflict arises when two or more defined on the same data item specify different actions and it is only possible to achieve one of them or enforcing two of them leaves one or more data items in an inconsistent state. The authors claim that static conflicts exist in rules and runtime conflicts arise based on user requests. If one data item is mapped to more than one obligation then a static conflict may arise. Policy level and obligation management level resolutions of static conflict are mentioned. As more than one obligation cannot be executed at the same time, in order to choose which obligation to execute, a ranking of the obligations is done and resolving conflicts for the runtime obligation is attempted by removing overridden obligations. A prototype for the automated enforcement of obligations related to privacy policies is implemented, and the authors have also described a prototype for the conflict management system.

Bettini (Bettini 2002) has defined provisions as actions to be performed before the decision is taken and obligations as actions to be performed after. Obligation monitoring is needed to ensure that a user has fulfilled the agreed obligations. The authors mention that the user's history of obligation fulfillment needs to be utilised for dealing with the user in the future. This can be done by assigning a numerical reliability rating similar to a credit rating. A fulfilling clause defines the actions to be taken when the obligation is fulfilled and a defaulting clause defines the actions to be taken when it is not by the user. The server is assumed to have a mechanism to monitor atomic obligations. An obligation, in a disjunctive normal form is said to be fulfilled if one of the obligations is achieved and in a conjunctive normal form if all are. The authors have proposed temporal reasoning support in order to monitor the obligations with timing constraints. The concept of "guarding" time has been defined as a time such that if no

event occurs until that time, the corresponding defaulting action will be performed at the next time instance. This paper mainly focuses on monitoring the user's action that is supposed to be completed by the user when s/he gains some permission, but the model focuses on limited types of obligations. This work did not focus much on implementing provisions i.e. the actions to be performed before taking decisions.

Irwin et al. (Irwin, Yu and Winsborough 2006) have described obligations as "a requirement for a subject to take some action at some time in the future" where it cannot be enforced by a system in a direct way. The authors say that these obligations are unenforceable but are monitor-able by the system. They describe a meta model where an obligation system is represented as a tuple of subject, action, object and time-frame and can be in four states: invalid, pending, fulfilled and violated. The authors say that a system is in an accountable state if all the users in the system have sufficient privileges and resources to carry out their obligations. A state transition in a system is said to be an obligation-abiding transition if after transitioning to the new state no pending obligation of the previous state becomes violated. They introduce the concept of system accountability and show how complex it is to determine whether it is accountable and further defined a simplified concrete model to determine this. If an obligation has been enforced by the system then it no longer remains an obligation according to their definition.

Demchenko et al. (Demchenko, et al. 2008) have enhanced the obligation handling mechanism of XACML where the obligations are returned by the PDP as these are written in the policy. Their obligation handling processes have three stages as follows: obligations are returned by the PDP in the form they are written in policy, next templates and instructions of the obligations are replaced by the real attributes and finally the obligations are actually enforced by the resource itself or by the trusted services managed by the resource.

Gama et al. (Gama and Ferreira 2005) have described a policy platform called Heimdall that provides a mechanism for the enforcement of different types of obligations. Heimdall separates the application development from the policy specification and enforcement, which provides a way for the policy administrator to define obligation policies independently from the application developer. For the enforcement of obligations an event is matched against the predefined policies set by the administrator. However, Heimdall does not provide any way for the user to define their own obligations which makes the model unsuitable for implementing privacy obligations. In our model we have a mechanism for supporting obligations residing in the administrator's policy and those coming with the access request, for example, with a sticky policy.

Katt et al. (Katt, et al. 2008) have proposed extensions of the Usage Control (UCON) (Park and Sandhu 2004) Model by adding post obligations into it which are not enforceable with the original core UCON model. It assumes that obligations are mandatory requirements a subject/system has to perform before or during a usage of an object. An obligation is viewed from four points- 1. Who must perform the action, 2. What object the obligation must be applied to, 3. When the obligation is to be performed (before/ during/ after a usage control session) and 4. For how long. Our work further considers application independent/ dependent

obligations and obligation combinations as an important addition to the privacy preserving authorisation system as described below.

In most of the work related to obligation handling they are left for the application developers to enforce, however, not all obligations are application specific. There are some obligations that do not vary for different applications (such as sending e-mail), and they can be processed in an application independent way. In our proposed model, the application independent obligations are designed to be handled by the system; so that the application developers do not need to worry about them. Furthermore, we have proposed to accommodate the policies and obligations from a number of authorities. The obligations returned by the policies from various authorities are combined if the policies have the same effect, unlike the presented works above, which do not address the idea of combining obligations.

2.5 Review of Conflict Resolution Strategies

The proposed privacy protecting advanced authorisation system, P-PAAS, is designed to combine policies written by a number of authorities in different policy languages. Conflicts may arise in the system when more than one access control policy is defined for the same set of subject, action and target. Different PDPs may return different access decisions, and the Master PDP needs to resolve the conflicts among them. The system needs to be provided with a conflict resolution policy which it uses to resolve conflicts among the policy decisions. In this section the various conflict resolution strategies taken by prior research are discussed.

The conflict resolution strategy of XACML is defined as a rule and policy combining algorithm as presented in Section 2.2.9.

Linington et al. (Linington, Milosevic and Raymond 1998) have mentioned three principles that are commonly used by legal frameworks in many countries which can be used for defining policy priority -
Lex specialis legi generali derogat- the specific overrides the general.
Lex superior legi inferiori derogat- higher authority overrules lower authorities.
Lex posterior legi auteriori derogat -new law overrides old law.

These principles are later used by Dunlop et al. (Dunlop, Indulska and Raymond 2003) for establishing precedence for resolving conflicts in policies. The authors have classified policy conflicts into four categories- Internal, External, Space and Role conflicts, and have proposed different strategies of when and how to resolve them. In the Pessimistic Conflict Resolution approach, it is assumed that all the potential conflicting states result in a conflict at some time and it takes preventive steps to resolve them at compile time. However, it is not always possible to detect all of them at compile time especially in distributed systems. The Optimistic Conflict Resolution Approach does not take any preventive measure but rather it detects and resolves conflicts at the run time. The Balanced Conflict Resolution approach assumes that the likelihood of conflicts occurring is high and so takes some preventive steps to resolve the static ones and if some potential conflicting states cannot be prevented they are monitored and resolved at run time. In the Individual Conflict Resolution approach each detected conflict is assessed separately to determine when to resolve it. Different strategies are described for

establishing precedence in order to resolve the conflicts. In the principle *Specific Overrides General* the policies applying to a more specific role gets precedence over those applying to a general role. In the *new law overrides old law* principle the creation date of both policies are examined to determine the precedence and in the *higher authority overrides lower authority* strategy a hierarchy of authority determines it. The assigning of explicit weights or priorities to policies is useful in a single domain but is difficult for a distributed domain. In the *negative/positive policy precedence* strategy either the negative or the positive policy takes precedence when conflicts arise. The authors also show that some strategies are valid for some situations and invalid for others and have suggested that only one principle is not suitable to resolve all the conflicts. In a single domain the local authority determines the resolution strategy and in a multi-domain it depends on the organisation's rules.

Ma et al. (Ma, Lu and Qiu 2009) have defined a way for the static and dynamic detection of policy conflicts by representing the policies in a graph. They note that the relationship between the policies needs to be characterised according to the conflict resolution requirements of the system. They have proposed a step-by-step process for conflict resolution where a rule of higher priority is selected at each step for resolving the remaining conflict until none remains.

Russello et al. (Russello, Dong and Dulay 2007) have mentioned that conflicts arise when multiple policies apply on the same subject, target and action. They have provided rules to define precedence between conflicting policies based on domain nesting which gives precedence to policies that apply to more specific instances of subjects or targets or both. Furthermore, they provide a solution for the situation where a global policy needs to override a more specific one and to identify such a policy, a special keyword 'final' is used in the definition of it. The strategies they use for conflict resolution are: firstly searching for policies with the keyword 'final', if more than one 'final' policy is found precedence is given to the negative authorisation policy. If no 'final' policy is found then searching is done for the most specific one and precedence in given to the negative one.

The policy conflict resolution techniques proposed by Syukur et al. (Syukur, Loke and Stanski 2005) for pervasive computing environment are: *role hierarchy overrides policy* and *obligation holds precedence over rights.* The first strategy is used when conflict occurs between the users of different roles and the user with a higher role gets precedence, and the later one is used when conflict occurs between the obligations and rights. They propose that a conflict can be resolved when it is detected or at the time when the potential conflict becomes actual.

For resolving conflict instead of identifying which policy prevails Charalambides et al. (Charalambides, et al. 2006) have proposed to provide conflict resolution policies in an application specific environment. These policies are triggered when the condition for conflict is satisfied and this paper only used policies related to Quality of Service (QoS) provisioning for Differentiated Service (DiffServ) networks.

Chanda (Chanda 2006) has provided conflict resolution for policies in military network management systems. The author has classified conflicts as application-independent and

application-specific conflicts. The application-independent conflicts are the *modality conflict* – which occurs when one policy requires that certain actions occur and other forbids the same set of actions; and the *redundancy conflict* occurs when two or more identical policies exist in the system. Among the application-specific conflicts the redundancy conflict is the same as the previous one; the *mutually exclusive configuration conflict* occurs when a parameter of the target is set to two different values by two different actions; and *the inconsistent configuration conflict* when multiple policies inconsistently configure a set of related parameters. The author notes that resolving conflicts of the mutually exclusive configuration depends on the intended semantics of whether to forbid another policy to set parameters or the most recently performed action needs to stay. For resolving the inconsistent configuration conflict, a constraint is inserted as a condition in the policy, to maintain the relationship of related parameters.

Lupu et al. (Lupu and Sloman 1999) mainly discuss the modality conflict which occurs when two or more policies with opposite modality signs refer to the same subjects, action and targets. They specify maintaining policy precedence relationships as a resolution of conflicts whereby the negative policy has priority. They also mention providing explicit priority values to policies whereby a more specific policy i.e. a policy applying to a sub-domain overrules the policy applying to a more general domain. However, this precedence rule fails when one conflicting policy has a more specific subject while the other has a more specific object.

Li et al. (Li, Wang, et al. 2009) have presented a Policy Combining Language (PCL) while identifying several problems of XACML. The authors have shown many strategies that cannot be specified in XACML, such as, weak-consensus: which permits (or denies) when at least one policy permits (or denies) while no other policy denies (or permits), strong-consensus: which permits (or denies) if all the policies permit (or deny) and the policy decision based on the majority of decisions. It has provided a general formalism of policy and obligation combination which is implemented and integrated with XACML and can be used by other languages.

Crampton et al. have introduced a resilient policy evaluation solution to protect the policy evaluation of a distributed computer system from failure if a sub-policy fails to be retrieved or evaluated (Crampton and Huth 2010 B). Three semantics have been presented for policy evaluation, two of which handle the exceptional situation by considering different possible outcomes. The authors have provided an interesting concept of having orchestration policies to combine decisions of "base" policies in a tree structure (Crampton and Huth 2010 A). In their demonstrated model a Policy Orchestration Point (POP) forwards requests to other PDP or POP for evaluation and combines the decisions according to the orchestration patterns defined for that POP. Our authorisation system component, Master PDP, (Chadwick and Fatema 2009), which is published earlier than the above, calls the conflict resolution policy and gets a decision combining algorithm (which can vary based on request context) and combines the decisions of relevant PDPs. However our authorisation system also considers combining the obligations returned by the PDPs which is missing in their model.

Mazzoleni et al. (Mazzoleni, et al. 2008) have proposed a solution for integrating the resource owner's and third party's policies based on the integration preferences chosen by the policy

authorities and the similarity measurement of those in a complex way. They also identify situations where integration is not possible. There is no such limitation in our work. Their work makes a suggestion about choosing a party to collaborate with. The focus of our work is different since we do not aim to provide such suggestions; rather we try to provide a way which always ensures that the policies of various authorities always get the right priority in terms of privacy protection of personal data. In our case, the relationship among authorities is hierarchical whereas it is peer-to-peer in theirs. With our system, the data subject's conflict resolution policy always gets certain privileges regardless of that of the controller and no other authorities' policies are allowed to override policies provided by the law. In such a case the data subject is assured of certain privileges and does not need to carry out the complex similarity matching of policies to choose a party to collaborate with. Our strategy assures that the privilege of certain parties is always preserved while choosing a dynamic conflict resolution algorithm based on the request contexts.

Mohan et al. (Mohan and Blough 2010) have argued that static conflict resolution may not be suitable for a dynamic environment where there is a need to adapt the policies dynamically. They proposed a dynamic conflict resolution strategy that chooses an applicable policy combining algorithm based on a set of environmental attributes. Although their strategy improves the applicability of the authorisation system for a dynamic environment, it does not yet solve the problem entirely. The limitation with their system is that the policy combining algorithm (PCA) rules have to be mutually exclusive which makes it unsuitable for a dynamic environment where individual parties are expected to provide independent policies. Moreover, the existence of more than one applicable PCA rule is treated as an error which halts the authorisation procedure. The protocol of our strategy is influenced by that of Mohan et al., but we have made the system suitable for a dynamic environment where all the parties can provide their own independent conflict resolution policies. Only one conflict resolution algorithm is chosen for a request context regardless of the number of conflict resolution policies defined for the same conditions; which resolves the problem of requiring the conflict resolution policies to be mutually exclusive.

2.6 Privacy Protection in the Cloud

Cloud computing has brought a new era of internet based data storage and processing power. The cloud offers enormous benefits to businesses such as reduced costs, since they no longer need to spend a large amount of capital on buying expensive application software or sophisticated hardware that they might never need. The key characteristics of Cloud include agility, low cost, device and location independence, multi-tenancy, high reliability, high scalability, security and sustainability (Gong, et al. 2010). However, despite all these benefits the cloud has to offer, privacy and security issues are still major challenges of cloud computing (Sabahi 2011). There are many security issues to consider, including fine grained access to cloud resources, privacy protection of data in the cloud, and auditing of cloud operations. Some threat models assume that the cloud provider cannot be trusted, and therefore propose storing only encrypted data. Others assume that the cloud provider can be trusted, and, that the threats come primarily from outside attackers and other cloud users. Given that most public cloud services are currently being run by large, trusted organisations such as Amazon, IBM, Microsoft and Google, we believe that the latter threat model is reasonable for many

users. Furthermore, organisations are now able to run their own private clouds, using open source software such as Eucalyptus (http://eucalyptus.com/). A private cloud implementation aims to avoid security risks by setting up the cloud inside the corporate firewall, within the organisation's infrastructure boundary. The trusted provider model is the most appropriate one for this scenario. For example, the National Grid Service in the UK is already experimenting with private clouds for use by the academic community, so consequently the trusted provider model is the one we adopt in this work.

Gellman's report at the World Privacy Forum (Gellman 2009) focuses on privacy issues and legal compliance of sharing data in the cloud. He mentions various legal issues such as the possibility of the cloud being in more than one location at the same time with different legal consequences and such legal uncertainty makes it difficult to assess the privacy protection available to users. We cater for this by allowing different policies from law to be configured into our authorisation system and allowing them to stop data being transferred to jurisdictions which do not have proper privacy protection.

Pearson (Pearson 2009) points out the key privacy requirements for clouds such as: giving notice to users before collecting their data, getting consent of the data subjects, safeguards to prevent unauthorised access and so on. She attempts to provide a set of guidelines for designing a cloud service with privacy protection. Pearson et al. (Pearson and Charlesworth 2009) propose to use accountability (created with a combination of privacy policy and contractual terms) to enhance privacy protection in the cloud. She identifies the key elements for provisioning accountability within the cloud which include transparency of data handling, assurance through privacy policy, control of access to data and compliance with laws. Pearson et al. (Pearson, Shen and Mowbray 2009) and Mowbray et al. (Mowbray and Pearson 2009) propose a privacy manager which can obfuscate personal data in the client site before sending it to the cloud service provider. This approach minimises the amount of sensitive data held within the cloud as only the client can obfuscate or de-obfuscate data with their chosen key. The problem with this approach is that the applications that are able to use the obfuscated data are limited, and this affects the services available to the user. Also the computation overhead of obfuscating and de-obfuscating is large and so it imposes constraints on the computing resources available to the user.

Lin et al. (Lin and Squicciarini 2010) propose a data protection model by which a potential cloud user can find and choose a cloud service provider based on a user based ranking of the service providers. The user can then integrate their privacy policy with that of the service provider and any sub-contractors, and this combined policy can be coupled with the data, rather like our sticky policies. The work is still at a very preliminary stage and more design and implementation is needed.

Itani et al. (Itani, Kayssi and Chehab 2009) present a security infrastructure for cloud providers to adopt. Privacy of the data in the cloud is ensured by the use of a secure cryptographic co-processor which provides a trusted and isolated execution environment in the computing cloud.

Wang et al. (Wang, et al. 2009) propose an anonymity based privacy preservation method in the cloud. They show a simple example for anonymisation of some data based on the background knowledge of the service provider but their work lacks the automation of such anonymisation and is suitable for very limited services.

We believe that the obfuscation or encryption of the data is not necessary for many cloud usage scenarios, especially private clouds. Our P-PASS can be used as a cloud authorisation service that can use privacy policies to protect access to the personal data at trusted cloud providers, who can independently encrypt/decrypt the data for storage in the cloud as required.

2.7 Previous Work on Obtaining Authorisation Rules from Legislation

Even though privacy laws have existed for quite a long time, the automatic enforcement of them is still lacking, and the process of converting them into automatically enforceable formats is considered to be tedious and complicated work (Papanikolaou, Pearson and Mont 2011). Much research has been performed on obtaining requirements from legislation which is mainly used as a guideline or for compliance checking.

The NEURONA project (Casellas, et al. 2010) has developed a data protection application based on the Spanish data protection requirements. It produces reports regarding the correct application of privacy measures to files containing personal data. If a file contains personal data but does not comply with an adequate level of security then it is classified as an erroneous file by their ontology. They provide a semi-automated way to determine whether any aspects of the current state of a company's personal data files might not comply with the established set of regulations.

Breaux et al. (Breaux and Anton 2005 B) have proposed a semantic parameterisation process to derive semantic models from privacy regulations. In a content analysis technique, goal-mining, the text is parsed to extract structured natural language statements expressed as goals. The goals that are obtained from privacy policy documents are re-stated to form Restricted Natural Language Statements (RNLS). By semantic parameterisation the RNLS are expressed as a semantic model where every component comprises at least an actor, action and object. Unfortunately, the semantic parameterisation process is not capable of parameterising many goals. For example, it cannot present the conditions which coincide with the conjunction "unless". In (Breaux and Anton 2005 A) the semantic parameterisation process is applied to the Health Insurance Portability and Accountability Act (HIPAA) Fact Sheet that has been prepared to define the rights (i.e. the permissions) and the obligations (i.e. what people and systems must do). The analysis procedure had three steps- 1. Identify a natural language statement that expresses rights, permissions or obligations 2. Derive a semantic model for actors, actions and objects for each statement 3. Derive rules with pre-conditions (if a pre-condition is true the corresponding effects must also be true) and effects built from temporal constraints. In (Breaux, Vail and Anton 2006) the methodology is presented for extracting and prioritising rights and obligations from regulations. The semantic parameterisation is developed from Grounded Theory which states that a theory that is obtained from a dataset is valid for that dataset (Charmaz 2003). The authors note that heuristics, used by their method, may be insufficient or inconsistent when analysing other regulations or policies. Breaux et al. (Breaux and Anton 2007) note that acquiring requirements from regulations is complex due to the intended and unintended ambiguities.

They have introduced a frame-based requirements analysis method (FBRAM) to get a semi-formal representation from regulations. However, there are limitations in their method such as relying upon a specific format of regulatory texts, and analyst's skills. The results from applying their methods on HIPAA privacy rules are presented in (Breaux and Anton 2008). However, the authors say that their methodology may not be applicable for other regulations. The manual process of identifying rights and obligations from regulatory texts are replaced by a tool called Centro framework in (Kiyavitskaya, et al. 2008). The Centro framework requires the manual construction of grammatical rules to identify basic concepts. They applied this method to the HIPAA regulations, but there is inadequate assessment of its applicability to other regulations.

Massacci et al. (Massacci, Prest and Zannone) present a case study of using Security-Enhanced Tropos, an agent based requirement engineering modeling and formal analysis methodology, to validate the compliance of the University of Trento to the Italian Data Protection Legislation. Their scope of focus is different from ours as they did not try to obtain access control policies from legislation.

Bkara et al. (Bekara and Laurent 2011) have presented a semantic information model that formalises legal requirements. This model can be used by the user to check automatically whether the service provider's request is compliant with the legislation. The semantic model associates the personal data type and service type with explicit legislative rules. For each service type, the data type allowed by the legislative framework for the service type is displayed. After selecting the data type for a service type the related privacy rules are displayed and the resulting privacy policy is then compared with the service provider's privacy policy to judge its compatibility with the legal requirements. The ontology can be a powerful tool to determine whether a service provider's privacy policy is compatible with legislation or not. However, the work does not focus on extracting all the access control rules from the legislative rules. In other words, not all the legislative rules were possible to express with this ontology. For example, "the data subject can access the personal data if there is no legal objection" is not possible to express with this model. The legislative based privacy rules are only capable of expressing the rules an enterprise can provide to the customer while collecting personal data.

Waterman (Waterman 2010) has identified that the translation of privacy laws and regulations into machine executable form is slow and difficult because of the unstructured form of legal text. In order to develop a policy aware accountable system that can compute compliance with data usage policy, they have worked on forming isomorphic intermediate representations of the US Privacy Act and Massachusetts Criminal Offender Records Law. The table based presentation of the intermediate representation was useful, but according to the authors it is not sufficient for the programmer to make them computer executable.

Mont et al. (M. Mont, S. Pearson, et al. 2010) say that translation of legislation/regulation to machine readable policies has proven to be very difficult. They have presented a hierarchy of policies based on a level of abstraction and consider that the legislation/ regulations present the most abstract policies and so those policies stay on the top of the hierarchy. Papanikolaou et al. (Papanikolaou, Pearson and Mont 2011) say that it is unreasonable to expect a computer

program to fully understand the legal or other policy text. However, even with the limited capabilities, the automation of Legal policy enforcement significantly reduces the effort required to ensure compliance. They have grouped previous research on the analysis of privacy and privacy related regulations and proposed to use techniques such as analysing natural language privacy texts, extraction of formalised rules and then their automatic enforcement. The work is still at an initial stage and hence the details of the procedure or findings are currently not available.

Wu et al. (Wu, Ahn and Hu 2012) propose a framework to determine whether the policies of a health care system are compliant with the Health Insurance Portability and Accountability Act (HIPAA) regulations. They first extract the policy patterns from both the HIPAA regulations and the local policies in a health care system to transform them into a formal representation. The formal representations are then transformed into a logic based representation using answer set programming (ASP). For compliance checking, they apply the same request to the two policies and if the effects are the same then the local policy is compliant. If the effect of the health care system policy is *allow* and the effect of the HIPAA regulation is *deny* then this non-compliance case is *less-constrained non-compliance*. If the effect of the health care system policy is *deny* and the effect of the HIPAA regulation is *allow* then this non-compliance case is *over-constrained non-compliance.*

Gopalan et al. (Gopalan, Anton and Doyle 2012) have presented a usage control model $UCON_{LEGAL}$ to express access and usage rules that they identified in HIPAA Privacy Rule. First a dataset is identified from the rules that govern access, use and disclosure of information. Then an inquiry driven approach is used to analyse the data set to identify the components of the Legal ACRs. Each statement of the data set is analysed by using inquiry questions to determine whether it grants or denies access, use or disclosure. The components in the Legal rules are identified that are not possible to present using the $UCON_{ABC}$ (Park and Sandhu 2004) model and hence they proposed the $UCON_{LEGAL}$ model that accommodates those components. The $UCON_{LEgAL}$ components are identified using seven sections of the HIPAA privacy rules. Further study is needed to ensure the reliability of the methodology so that it can be repeated by others. Their methodology is a bit similar to ours, which is published earlier than theirs (Fatema, Chadwick and Van Alsenoy 2011), in a way that it first separates the Legal rules that are related to access control. We also have used methods like inquiry question to identify whether the rule expresses a condition on accessing a data item. However, their work focuses on analysing the Legal rules to express in the UCON system and proposes a modification of $UCON_{ABC}$ to construct the $UCON_{LEGAL}$ system to express the Legal rules. On the contrary, our work focuses on obtaining attribute based enforceable access control rules, which we have implemented using XACML and PERMIS policy languages and installed into the P-PAAS authorisation system (using either a XACML or PERMIS PDP).

Xiao et al. (Xiao, et al. 2012) have provided a procedure for automatic extraction of XACML Access Control Policy (ACP) from Natural Language (NL) software documents. Their approach consists of three steps: i) Linguistic Analysis: parses NL software documents and annotates sentences with semantic meanings for words and phrases. ii) Model-Instance construction: identifies subject, action, resource elements and effects and constructs ACP model instances

based on that. Transformation: transformation of each ACP rule into XACML rule. Their approach is not suitable for extracting ACP from the EU DPD for two reasons: i) This approach works for software documents with ACP rules which follows a specific style [subject] [Can/can't/ is allowed to][action][resources] whereas the language of the EU DPD is ambiguous, includes implicit information and is highly dependent on human judgment and interpretation.
ii) Their approach identifies only the basic elements: subject, action, resource and those are transformed into XACML role, action-id and resource-id attributes and does not include any other type of attributes. Our constructed ACPs from the EU DPD are full of complex conditions on a number of different types of attributes which this system fails to incorporate in their ACP construction.

Deontic logic, the logic of expressing pertaining to the obligations, permissions and rights of agents, has been proposed in legal knowledge representation (Jones and Sergot 1992, Ortalo 1996). Deontic logic is concerned with the presentation and analysis of reasoning from the legal text to find the modal of a sentence i.e. whether it is an obligation, permission or right. However this complex analysis is not sufficient in identifying or formalising the enforceable conditions for which the agent is permitted or denied an action on an object. For example Deontic logic might be helpful in analysing whether Article 7. (c), "Personal data may be processed if processing is necessary for compliance with a legal obligation to which the controller is subject", represents an obligation, permission or right by exposing the ambiguity of the meaning of the presented text. Besides, it cannot help to identify enforceable conditions from the legal texts. Finding or constructing an enforceable condition from a complex legal document human judgment is essential.

In comparison, to all of the above our work is based on the analysis and extraction of rules from the EU DPD and converting these into executable policies (using both XACML and PERMIS PDPs) so that automated decisions can be obtained. To our knowledge no other work has been performed on obtaining all possible access control rules from the EU DPD to get automated access decisions on behalf of it.

2.8 Conclusion

From the literature review we conclude that none of the privacy protecting authorisation systems considered the need for having support for multiple policy languages, which our system does. Only one of the recent research projects (EnCoRe) has thought of having an access control policy from the law included in the system, but they did not mention how the policy from the law can be integrated with the other policies, or whether Legal policy will be given the highest priority or not. Moreover, they could not show how to obtain the policy from the law. Unlike P-PAAS no other previous privacy protecting authorisation system addresses the issues related to privileged access to personal data by individual parties, transfer of personal data based on certain conditions or access to personal data based on contract. Furthermore, none of the work considered combining obligations returned by various authorities. So far no other previous work has been done on obtaining access control rules from the EU DPD and other works related to the extraction of authorisation rules from legislation were not found suitable for our purpose.

Chapter 3

3 Design of the System

3.1 Requirements

An authorisation system provides decisions for who is allowed to perform what actions on what data. While designing an authorisation system that provides privacy of personal data we are aware of the following high level requirements:

- The need to protect the privacy of the data subject (obviously).
- The only data we are concerned about are personal data.
- Data protection legislation should be adhered to.
- The need to ensure the distributed enforcement of privacy policies so that when personal data are transferred between systems, the privacy policies can go with them.
- We should build on previous knowledge, know-how and tools for authorisation system construction wherever possible in order to minimise the cost of building a privacy preserving authorisation system.

When considering the legislative requirement, we need to decide which legislation should be adhered to, since we cannot assume that all countries have the same legislation. In this thesis, we have only concerned ourselves with the EU Data Protection Directive (Directive 95/46/EC 1995). Section 2.3.1 has already outlined the EU DPD requirements for a privacy protecting system. Applying these to the authorisation component of a system leads to the following requirements:

- Access should only be granted if it is in line with the original purpose of collection.
- The wishes of the data subject should be taken into account (informed consent).
- Data subjects should have access to their own personal data, except under certain conditions.
- Data subject should have control over their personal data, e.g. be allowed to object to the processing of personal data, request for update of data, ability to submit policy to protect access to his/her personal data.
- Under certain conditions, the data subject should be informed when his personal data are accessed.
- Certain privileged people should be given unrestricted access to personal data.
- There should be special rules for transferring personal data from the EU to other countries.
- Special access can be granted to personal data based on signed contracts between various parties.

More information about the above requirements is given in Chapter 4 where the EU DPD is discussed in detail.

Existing state of the art authorisation systems are policy based which enforce the policies of various stakeholders, providing they are all written in the same policy language. When conflicts arise between their policies, they have in-built conflict resolution mechanisms, which should provide a good basis for the construction of our privacy protecting authorisation system. However, they do not allow for policies specified in different languages, which we believe is essential as explained in Section 2.3.1. and 3.2. Most previous work has concentrated on adding user policies to the system, but no prior work has attempted to translate the EU DPD into an authorisation policy, and resolve conflicts between this and the user or controller policies. In addition, no previous work has attempted to integrate authorisation decision making with contract validation.

In this research work, an authorisation system has been designed that is capable of satisfying the above requirements. This chapter presents the design of the Privacy-Protecting Advanced Authorisation System (P-PAAS).

3.2 Assumptions

The conceptual model of P-PAAS is designed based on some assumptions which are listed here:

1. A one size fits all policy language is not suitable for constructing policies satisfying all types of requirements.

Justification: Today we have many examples of different policy languages e.g. XACMLv2 (XACMLv2 2005), XACMLv3 (XACMLv3 2013), PERMIS (Chadwick, et al. 2008), P3P (W3C 2002), Keynote (Blaze, Feigenbaum and Ioannidis 1999), EPAL (Ni, et al. 2007), DPAL (Barth, Mitchell and Rosenstein 2004) etc. and hence many different PDP implementations. For example, XACMLv2 does not support delegation of authority whilst XACMLv3 and PERMIS do. The XACML policy language assumes a stateless PDP and hence cannot support state based policy rules such as separation of duties (SoD), whilst PERMIS is state based and can support both dynamic and static SoD. Keynote, on the other hand, uses the same language to describe both credentials and policy rules whereas none of the other policy languages does this. P3P is designed specifically to express privacy policies, whereas most of the other policy languages are designed as access control or authorisation policy languages. EPAL was also specifically designed for privacy policies. EPAL uses sequential semantics that makes it consistent, but it does not provide guaranteed safety, local reasoning (i.e. the ability to express a statement that needs to be enforced by all policies) or logical combinations of policies whereas DPAL provides local reasoning and logical combination (Barth, Mitchell and Rosenstein 2004).

2. Recipient organisations have to be trusted to a certain extent to handle the received personal data.

Justification: If recipient organisations were not trusted at all to handle personal data, then we would need a DRM system or similar to pass the personal data to them. Even then, no DRM system has been invented that is hack proof and so ultimately they have to rely on legal measures to prosecute offenders who abuse the data given to them. The alternative model we adopt is that recipient organisations can be trusted to handle personal data, providing they

have the proper tools to help them. If a sender does not trust a recipient then we assume they will simply not send any personal data to them. The trust we place in the recipient organisation, is that it will evaluate any incoming sticky policy, and if it knows it can obey it, it will accept the personal data, otherwise it will refuse to accept the data (since it will not want to break the law by violating the policy). A sender trusts some entities and not others. For those it trusts, it only trusts them to give an honest answer as to whether it can or is able to enforce the sticky policy. If it can it will accept the data, and if not it will refuse to accept the data.

3. We assume that the law is paramount, and that it must always be obeyed. Thus legal policies must override all other conflicting policies.

Justification: We assume that no organisation wants to be prosecuted for breaking the law as this will badly affect its reputation and ultimately its bottom line.

4. We assume that implementers will want a simple way of integrating a privacy protecting authorisation system into their applications, and that most will not want to be locked into any commercial software supplier via proprietary interfaces or protocols.

Justification. If a privacy protecting authorisation system remains independent from any application and can communicate using a standard protocol it eases the process of integrating the authorisation system with applications. This will inspire the organisations to adopt the authorisation system with their already existing applications.

5. We assume that developers are willing to develop their applications to conform to appropriate standards.

Justification. Having an application that conforms to standards makes it easier to mix and match components, and avoids lock in to proprietary interfaces. Consequently we will design our system to conform to appropriate standards where at all possible.

6. We assume that applications will uniquely identify their protected resources.

Justification. This is a reasonable assumption to make, since applications need to be able to differentiate between their protected resources.

3.3 Conceptual Model

At the highest level of abstraction the P-PAAS can be seen to comprise an application independent conceptual component, namely the authorisation service (see Figure 3.1) and an application dependent component called the Policy Enforcement Point (PEP) that calls the authorisation service. Even though the authorisation service comprises a complex set of functional components, it remains a black box to the application PEP. The P-PAAS further comprises a newly added component, the Contract Validation Service (CVS) for aiding the

validation of contract based access to personal data. In this section the overall structure of the P-PAAS system is described.

3.3.1 Application dependent PEP / PEP

According to the ISO standard (ISO 1996) the PEP (which is named as the AEF in (ISO 1996)) is application dependent. Similar to the ISO standard, in P-PAAS it receives the access request (step 0 of Figure 3.1) and passes it to the authorisation service for an access decision (step 2 of Figure 3.1). The PEP also optionally calls the Contract Validation Service (ConVS) when it receives a request with a contract (discussed in Section 3.3.3). It also retrieves the resource attributes for a requested resource and adds those attributes to the request context before passing that to the authorisation service (step 1 of Figure 3.1). More about the resource attributes is discussed in Chapter 4. When the PEP receives a decision from the authorisation service it enforces that decision by either allowing the user to access the requested resource (when the decision is a Grant) or by denying the access otherwise. When a response along with an obligation is received, it executes the application dependent obligation by calling the obligations service.

The PEP acts as an interface between the user and the protected resource. A privacy protecting authorisation system essentially deals with personal data and policies for protecting the personal data. It is not expected that the user is aware of the available policy languages or can write a policy with those languages. Therefore, a person needs to be provided with an easily usable interface to input his/her preferences or choices and these preferences need to be converted into the actual machine executable policies. The preference options provided (via an interface) to the user depend on the data the system handles and the functionalities of the application. For example, a hospital deals with medical data and may provide an interface for the user to choose the Medical Professional/s s/he wants to share his/her data with, whether s/he wants to allow medical students to read his/her data and so on. The provided options would be different for an organisation dealing with other types of personal data, such as personal blogs or CVs. This is due to the fact that the potential requester for accessing that data would be different. The application handling the policy preference interface converts the user's chosen options into machine executable policies and the user's request (provided via the interface) into a standard format which the authorisation service will understand. It then sends those to the authorisation service via the PEP. Some examples of converting the preference options into executable policies are given in Chapter 5.

3.3.2 Authorisation service

At the highest level of abstraction, we propose a conceptual model that comprises a standalone authorisation web service that makes authorisation decisions for remote applications. The authorisation web service should talk a standardised protocol so that different applications can interact with different authorisation services from different suppliers. The authorisation service should support many different policy languages, since there is no single ubiquitous authorisation policy language (see assumption 1).

The authorisation service should provide support to applications for sticky policies, by analysing any incoming sticky policies and informing the application whether the policies can

be enforced or not. This allows the application (owner) to conform to our trust assumption (see assumption 2). The authorisation service should relieve applications from the burden of having to store and retrieve sticky policies and should do it on their behalf, since this will make it easier to integrate sticky policies into applications (see assumption 4). The authorisation service should return all applicable sticky policies to the application when the associated data are to be transferred to another location.

To provide privacy of personal data, the authorisation service needs to integrate the policies of all the authorities who have any control over the data. Therefore in this system we have considered four different types of authorities for any personal data –

a. **Law:** Access to any personal data item needs to be protected by the relevant legislative rules. Consequently, the legislative rules related to accessing personal data need to be converted into machine executable authorisation rules and are then considered as authorisation rules from the "law".

b. **Data subject:** The data subject of any personal data is the person about whom the data reveal information. The data subject may have his/her own policy.

c. **Issuer:** The issuer of any personal data is the entity who issues the data. For example, a doctor is the issuer of a medical record, the university authority is the issuer of a university degree and a person is the issuer of personal choice or statement such as his/her favourite drink. The issuer can also be identified by a role instead of a specific identity of a person.

d. **Controller:** The system that is holding the personal data and controlling the flow of the personal data is the controller. The policies of the controller are, in fact, the policies of the administrator of a traditional authorisation system and these also need to be integrated with the policies from the other authorities. A data processor processes data on behalf of the controller and does not control the flow of data like a controller. Hence the processor is not considered as a policy authority.

In our system the policies of the different authors remain independent of each other and are independently evaluated. This separation helps to enforce them in a distributed system, when they are transferred as "sticky policies" to other systems along with the data they control. The receiving system does not need to employ complex processing algorithms in order to create an integrated complex "super" policy from the received policies and the organisation's own policy; it only needs to start an independent PDP (Fatema and Chadwick 2014). Furthermore, Having a separate Legal PDP offers some advantages, including: a) it helps to make sure that the legal policy always gets preference over any other policies; b) when personal data move from one legal domain to another, this separation will help to integrate the new legal rules just by replacing the legal PDP without manually integrating the policies of the new domain.

Finally, the authorisation service should have a well-defined dynamic conflict resolution strategy in order to determine the ultimate authorisation decision to be returned to the application, should multiple policies that give conflicting decisions apply to any authorisation decision request. This is further described in Section 3.4.

At the next level of abstraction we look inside the authorisation service black box to determine

its conceptual set of functional components and the mechanism for sticky policy enforcement and distributed enforcement of policy.

3.3.2.1 Application Independent PEP (AIPEP)

Since the PEP and authorisation service are remote from each other, there needs to be a protocol handler that can receive the protocol message from the PEP and:

- extract any attached sticky policy and put it in a sticky policy store,
- ensure that there is an embedded PDP that can evaluate this sticky policy (i.e. supports the particular policy language.) If not, an "unsupported policy language" error message will need to be sent to the application,
- extract the user's credentials and pass them to a credential validation service for validating,
- extract the authorisation decision request and give it to an embedded PDP for an authorisation decision.

We have called this protocol handling component the application independent PEP (AIPEP) since it acts on behalf of all applications and calls the embedded PDP(s) on behalf of the application.
From the PEP's point of view, the AIPEP is a remote PDP, accessible via a standard secure protocol, and from the embedded PDP's point of view, the AIPEP is a normal (local) PEP.

In the privacy preserving scenario where the data subject should provide his/her own privacy policy, the policy needs to be passed dynamically along with the decision request to the authorisation server. The data subject, or the PEP acting on her behalf, should not need to access the policy store (or PAP) of the authorisation server to do this. Instead, a standard protocol for communicating with the authorisation service should ensure that it is capable of carrying this policy along with the request context. The implementation options for this are discussed in Chapter 5.

The AIPEP is intended to reduce the burden on the application developers by handling as many application independent functions as possible. Consequently, it performs the application independent obligations (step 11 of Figure 3.1) so that the application developers remain free from the responsibility of enforcing those obligations.

The application programmable interface (API) between the AIPEP and embedded PDPs should be standardised if possible, so that different PDPs that support different policy languages can be embedded into the authorisation server. Different implementations of the same policy language PDP can also be embedded with this design, as some implementations may be quicker or less resource intensive than others. In chapter 5 we look at the different alternatives for this standard API. Suffice to note here that since we propose to use a standardised PDP API, then it either cannot pass a policy (since there are numerous different policy languages and no ubiquitous standard policy language) or it must have an extensible way of referring to them, for example, by using the ASN.1 "any" or XML "string" construct. We propose to adopt the former approach, so that a policy will never be carried across the AIPEP-PDP (Master PDP) API. Therefore another function of the AIPEP is to ensure that the embedded PDPs are initialised with the correct policies that are needed to handle the current

authorisation decision request before the request is passed to them. Since there are a number of APIs available for the chosen interface these are discussed in Chapter 5.

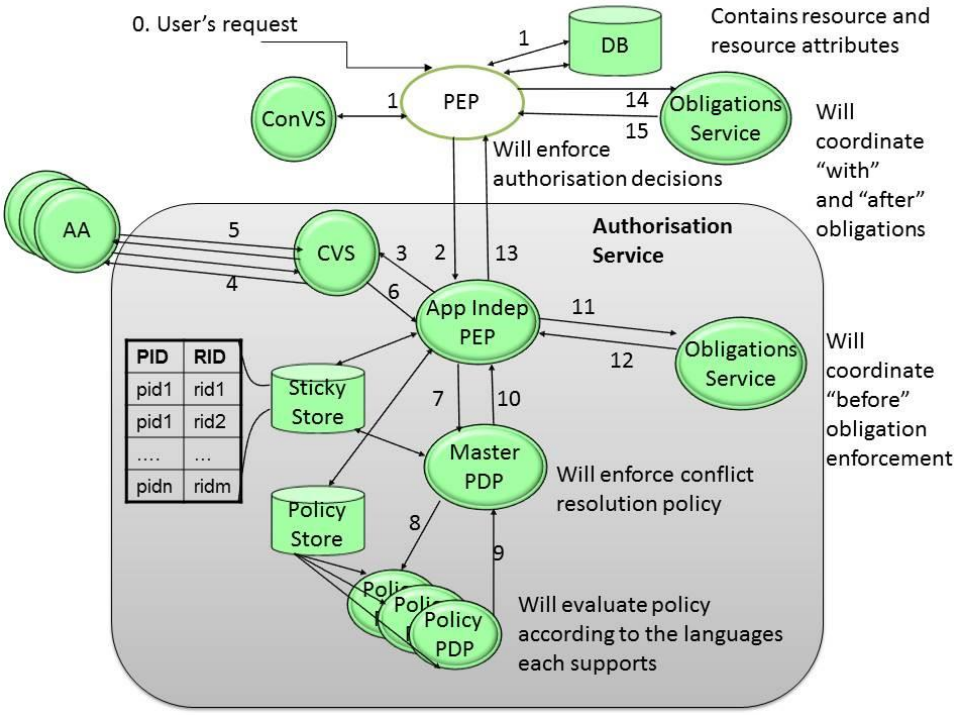

Figure 3-1. P-PAAS infrastructure

3.3.2.2 Master PDP

An authorisation decision request may require a number of different policies to be evaluated, possibly written in different policy languages by different authorities, so this raises the necessity to have a well-defined conflict resolution strategy. Since the dynamic conflict resolution strategy of the system demands a broader description it is described fully in a separate section, in Section 3.4. Suffice to say here, that we model this in terms of a Master PDP that dynamically determines the conflict resolution rule to use for the current authorisation decision request, and then sequentially calls multiple PDPs, each being capable of working with one specific policy language (step 8 of Figure 3-1), obtains their authorisation decisions (step 9 of Figure 3-1), and then resolves any conflicts, before returning the overall authorisation decision and any resulting obligations to the AIPEP (in step 10 of Figure 3-1). Consequently, the AIPEP will only interface with the Master PDP, and will delegate the task of multiple policy evaluation to it. The API for the Master PDP – PDP communication will be the same as the API for the AIPEP-Master PDP interface.

When the Master PDP returns the final authorisation decision to the AIPEP, this decision may be accompanied by a number of obligations. The AIPEP passes these obligations to an obligations service for evaluation. Any outstanding obligations that have not been evaluated by the obligations service will be passed back to the application along with the authorisation

decision response, for it to evaluate them. These obligations may be written in a number of different policy languages since they will have been generated by different PDPs. Providing each obligation clearly indicates its policy language, the appropriate obligation handler can be called. The Master PDP further provides a mechanism for obligation co-ordination which allows integrating obligations returned by multiple PDPs. The details are provided in Section 3.4.3.2.

3.3.2.3 Credential Validation Service (CVS)

The CVS is a component responsible for validating a set of credentials of a subject according to a credential validation policy which is usually provided by the Policy Administration Point (PAP). This policy tells it which credentials are valid, in terms of who the trusted attribute authorities (AAs) are, the maximum set of attributes each can issue and which attributes each is trusted to issue to which groups of users. The credentials are issued by multiple dynamic attribute authorities of different domains. The functionality of the CVS is like the XACML component Policy Information Point (PIP) which acts as a "source of attribute values". The idea of the CVS was introduced in the PERMIS project prior to the start of the research presented in this thesis and is described in detail in (Chadwick, et al. 2008, Chadwick and Su 2009). For a given set of validation policy and credentials the CVS returns a set of locally valid attributes.

The CVS is called by the AIPEP (step3 of Figure 3.1) for validating credentials. The CVS can work in pull, push or push-and-pull mode. Pull mode means that the requester does not have any credentials and requires the CVS to pull the credentials from the remote authorities. Push mode means that the WS-Trust request message contains the full set of credentials which are to be validated by the CVS. Push-and-pull mode, as its name implies, requests the CVS to validate the credentials in the request message and pull any further credentials that it can find for the subject of the authorisation decision query.

3.3.2.4 Obligations service

One important mechanism for ensuring privacy is to provide a way for enforcing obligations associated to a privacy policy. Obligations are the actions that must be performed when a certain event occurs; therefore when the event is an authorisation decision, the obligations are actions that must be performed before, after, or along with the enforcement of the authorisation decision, as defined in (Chadwick, Su and Laborde 2008). An example of *before* obligation can be to increase the amount of logging before a user is given access to monitor what s/he is doing or to ask for the consent of the data subject before granting access to personal data. An example of *with* obligation is to decrement the user's balance simultaneously with the user's access to the system for withdrawing money. An example of *after* obligation can be to send an email to the data subject notifying his/her data have been accessed.

Many obligations are application specific, such as, an obligation to charge the customer a fee if s/he goes overdrawn on his/her current account. However, some obligations can be implemented in an application independent way, for example, recording the authorisation decision in a secure audit trail, or emailing the security administrator that a certain decision has

been made. Each obligation should have a unique identity. At the time of construction, the obligations service is configured with the set of obligation IDs it can support. When passed a set of obligations by the AIPEP, the obligations service walks through this list and calls the appropriate application independent obligation service. If all obligations are processed successfully, a success result is returned. Each of the application independent obligations must be of temporal type *before*; otherwise they cannot be enacted by the AIPEP. This is because the AIPEP only gets invoked in advance of the PEP enforcing its decision and is inactive after the actual enforcement has taken place.

3.3.2.5 Sticky policy enforcement

To enforce privacy based on the policies of various authorities we need to make sure that the policies are stuck with the data within the system so that the relevant policies are enforced while making an authorisation decision to access a resource. Hence there is a need to maintain a link between the data and the policies.

Policy store: The policy store is the location where policies can be safely stored and retrieved. If the store is trusted then policies can be placed there in an unsecured manner. Otherwise policies need to be protected e.g. digitally signed and/or encrypted, to ensure that they remain confidential and are not tampered with. Each policy has an identity, the policy identity (PID). The AIPEP can use the PID when asking either the PDP/CVS factory to spawn a new PDP/CVS or the sticky store (described in the next section) to stick this policy to some personal data. This design clearly separates the implementation details of the policy store from the rest of the infrastructure, and allows different types of policy store to be constructed e.g. built on an LDAP directory or RDBMS.

Sticky store: The sticky store holds the mapping between the policies and the resources to which they are stuck. This is a many to many mapping so that one policy can apply to many resources, and one resource can have many sticky policies applied to it. The design requires that each PID is globally unique so that when a sticky policy is moved from one system to another, the receiver can determine if it needs to analyse each policy or not. Already known PIDs do not need to be analysed, whereas unknown PIDs need to be evaluated to ensure that they can be supported, otherwise the incoming data and sticky policy are rejected. Each resource protected by the system has a locally unique resource identity (RID). When the PEP passes a request for storing a resource to the AIPEP, it also passes the RID along with the policy. The AIPEP stores the policy in the policy store and the RID along with the PID in the sticky store. While passing an authorisation request for accessing a resource, the PEP passes the RID of the requested resource to the AIPEP (for it to retrieve the PID stuck with the RID).

Sticky policy contents: To combine the privacy policy and the private data together the concept of stickyPAD is proposed which is a combination of a **sticky p**olicy (or a set of sticky policy) **a**nd the **d**ata to which the policy applies. The schema of the stickyPAD is presented in Chapter 5. Different kinds of sticky policy are defined in (Chadwick and Fatema 2009).

Sticky transfer mechanism: Prior research focussed on different methods for enforcing sticky polices (Chadwick and Lievens 2008). Three methods for enforcing sticky policy have been proposed as- Application Protocol Enhancement Model, Encapsulating Security Layer Model

and Back Channel Model. Some applications, for example S/MIME, are already able to attach data and policy, and hence are capable of forming a StickyPAD. When the PEP receives a message to be sent to a remote PEP, it parses the message and sends the extracted information to the AIPEP who passes the request to PDP. If the PDP grants the request it optionally returns an obligation to the AIPEP which tells the AIPEP what policy should be sent with the outgoing message. The AIPEP enforces the obligation and makes the policy packet to send with the outgoing message and passes to the PEP with the decision returned by the PDP, the PEP then attaches the policy packet with the data in an application dependent way and sends to the remote PEP, where the remote PEP can parse the received packet and extract policy and message to pass to the AIPEP and the AIPEP may also update the PDP policy to carry out the same policy.

3.3.2.6 Distributed enforcement of policies

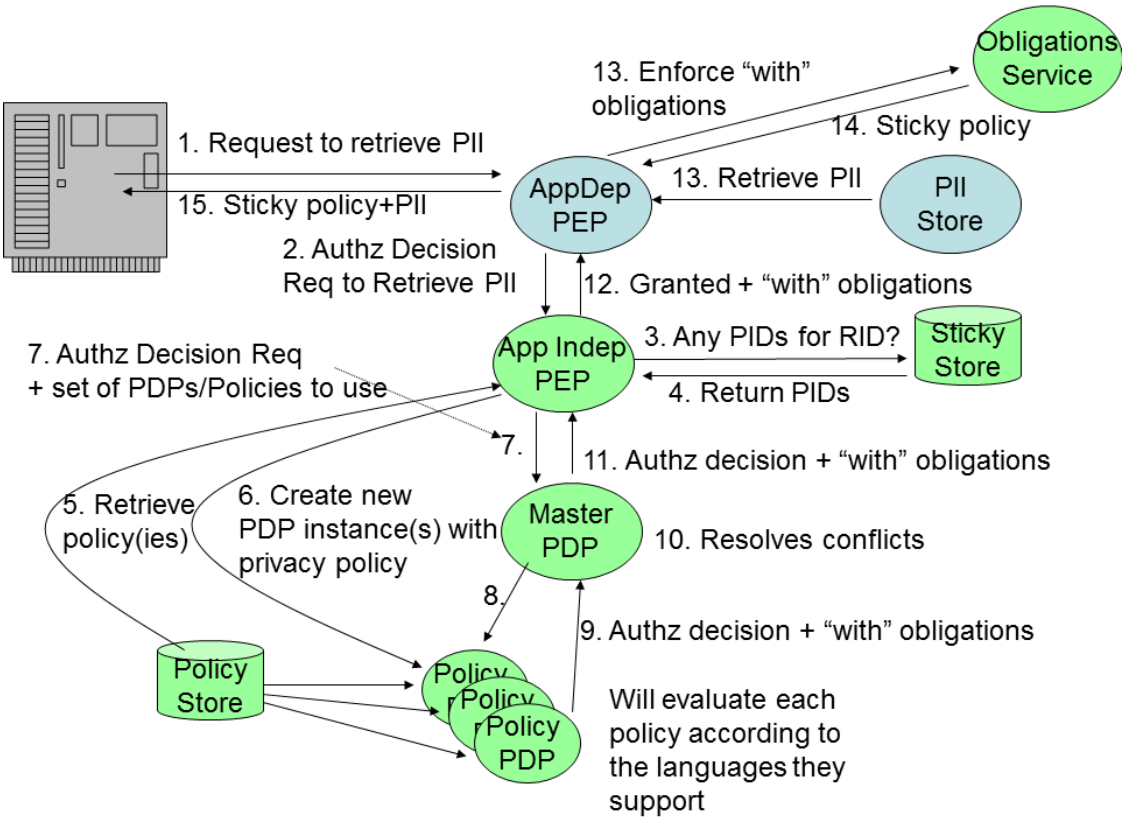

Figure 3-2. Flow of data when a remote third party requests access to a resource

Prior research on distributed enforcement of sticky policy of Mont et al. (Mont et al 2003 A, Mont et al. 2003 B) has focused on the problem of ensuring that personal data are only disclosed to trusted remote parties. This is achieved by encrypting a sticky policy along with the personal data before sending that to the other parties, and only distributing the decryption keys if they were trusted. However, the authors have admitted that their solution only helps to mitigate against the risk of unauthorised access to the data whilst it is shared encrypted to the remote party, but does nothing to ensure enforcement of the sticky policy. Once the data are disclosed it is not possible to control the misuse of that. We would argue that the complex

procedure of encrypted sticky policy distribution is both unnecessary and insufficient, and does not solve the problem of the recipient obeying the policy of the sender. If the remote site is trusted, encryption is not necessary, since it can be trusted to store its own data securely in whatever way it wants to. If it is not trusted, then it may ignore the sticky policy after the decryption has taken place, since their procedure does not provide any way to make sure the obtained sticky policy is enforced. Instead, we have proposed an alternative mechanism that ensures that a trusted recipient obeys the sender's sticky policy. To do this, we use an obligation based protocol for the distributed enforcement of sticky policy. In our mechanism, if the remote system is not trusted, and by this we mean is not running a P-PAAS conformant system, then it is not be sent anything. If it is trusted, then the sticky policy is sent to it (unencrypted) along with an obligation to start a PDP. The authorisation system only allows the receipt and storing of the personal data if the obligation to start a PDP with the received policy can be enforced i.e. it knows that it supports the policy language, has a PDP for this, and also has implemented the obligations. If the receiving authorisation system knows that it cannot enforce the incoming policy, then it denies the incoming request and never receives the private data. Our protocol for distributed enforcement of sticky policy is described below.

When an application wishes to transfer a user's personal data, the holding application service intercepts the request and makes an authorisation decision query to the authorisation service asking if this application has permission to transfer (or other access mode depending upon the client's request) the data identified by its unique RID. The authorisation decision request for transferring personal data becomes "Does this third party have permission to retrieve the data identified by this unique RID?" When the subordinate PDPs are asked if the third party is allowed to retrieve the personal data, then along with each Grant decision there is a "before" obligation which instructs the authorisation system to retrieve the PDP's policy from the sticky store and return it to the application service, together with a "with" obligation which requires the PEP to put this sticky policy in the relevant application protocol field along with the personal data. Once the third party's authorisation system receives the response, the application extracts the sticky policy(ies) and passes this/these to the authorisation infrastructure service along with the authorisation decision request "can this third party receive this data item into its data store, using this policy(ies) in conjunction with the existing policies". Eventually, all the relevant policies (including the data subject's one) are evaluated. The data subject's policy (at least) requires the third party to store and enforce the subject's policy, and this causes a "before" obligation to be returned to the AIPEP, which ensures that the sticky policy is safely stored in the authorisation infrastructure before returning Grant to the receiving application. If the receiving authorisation infrastructure is not able to enforce the sticky policy then the application is denied permission to receive the personal data.

3.3.3 Contract Validation Service

A component Contract Validation Service (ConVS) is added to the P-PAAS in order to validate a digital contract, which is an agreement between the parties and is signed by the agreeing parties. The concept of contract came from the idea of enforcement of some Legal rules of the EU DPD (Directive 95/46/EC 1995) which allows access to personal data when it is necessary for the performance of a contract and the data subject is a party of the contract (Article 7 (b)). It also allows transfer of personal data to a non-EU country or a country that does not have an

adequate level of protection to personal data if there is a contract between the data subject and the controller or between the controller and a third party which is concluded in the interests of the data subject (Article 26.1 (b) and (c)). Nowhere in the literature have we seen to address this important issue to automate the contract based access to personal data which the law has specified. Here we present the ConVS component which helps to automate the contract based access to personal data and we show how to integrate it with the authorisation system. We have shown a way to automate the authorisation of contract based access to personal data with the help of the ConVS component and the Legal PDP rules (presented in Chapter 4) which allow access to personal data based on contracts. The validation of such authorisation scenarios are presented in Chapter 5 along with the request contexts presented in the Appendix 5.

Here are some examples of the use cases of contract based access to personal data.

1. Data subject has a contract with a third party, e.g., Mr. M signs a contract with a health insurance company (HIC) to let HIC access the summary of treatment information and the billing information at X Medical Centre. Both Mr. M and HIC are the parties of the contract and sign it (a representative officer actually signs on behalf of HIC). The contract also mentions the data (treatment information or billing information) that is allowed to be processed due to the contract and also mentions the employee of HIC as an authorised requester so that only the employee can access the personal data.

2. Data subject has a contract with the data controller, e.g., Mr. M signs a membership contract with a gym GetFit to pay £25 monthly for the next 12 months and provides his name, address and bank details. The gym has appointed a financial service provider company SafeCollect for collecting the membership fees monthly from the members' accounts. The membership contract of Mr. M with GetFit mentions that the employee of the organisation SafeCollect is an authorised requester to let the employee of SafeCollect access Mr. M's personal data (name, address and bank details) to ensure timely collection of payments. The company SafeCollect is passed a copy of the contract or the contract's unique identifier which the employee can use while requesting access to personal data mentioned in the contract. In this case, the controller itself is a party so it can store the signed contract or information from the validated contract in its repository.

3. Data controller has a contract with a third party, e.g. there is a contract between a health care centre and each of the registered pharmacies to access the prescriptions of patients for providing medicines. The contract indicates what type of data (prescription) is allowed to be accessed by the authorised requester (employee of the registered pharmacy) by this contract.

The validation of the use cases for the contract based access to personal data are presented in Chapter 5.

3.3.3.1 Construction of the signed contract

A contract is an agreement between parties. Therefore, a contract needs to define the terms and conditions on which the parties have agreed, which we call the ***contract document***, and the signatures of the parties as a proof of their agreement on the defined terms and condition.

All the parties have to sign on the same ***contract document***. The ***contract document*** and signatures of the parties over that form the full ***signed contract***. Each contract needs to have a globally unique identity which also needs to be mentioned in the signed contract and the signatures, in order to verify the signatures have been done on the same ***contract document***. In order to provide globally unique identify for the contract document a URL can be used since it not only provides a unique identity to the contract document but also provides a way to access a document remotely. All the parties sign over the contents of the ***contract document*** with their own keys. The URL and the digest of the ***contract document*** become a part of the digital signatures and are used while validating the contract. The URL of the ***contract document*** and the signatures by all the parties over the contract document together form the ***signed contract.*** The process of constructing the ***signed contract*** is described in Figure 3-3. All the parties have access to the signed contract which they can use for validation while accessing data based on that contract. The schema of the signed contract is given in Chapter 5. Various implementation options for the contract document and the signature methods are also discussed in Chapter 5.

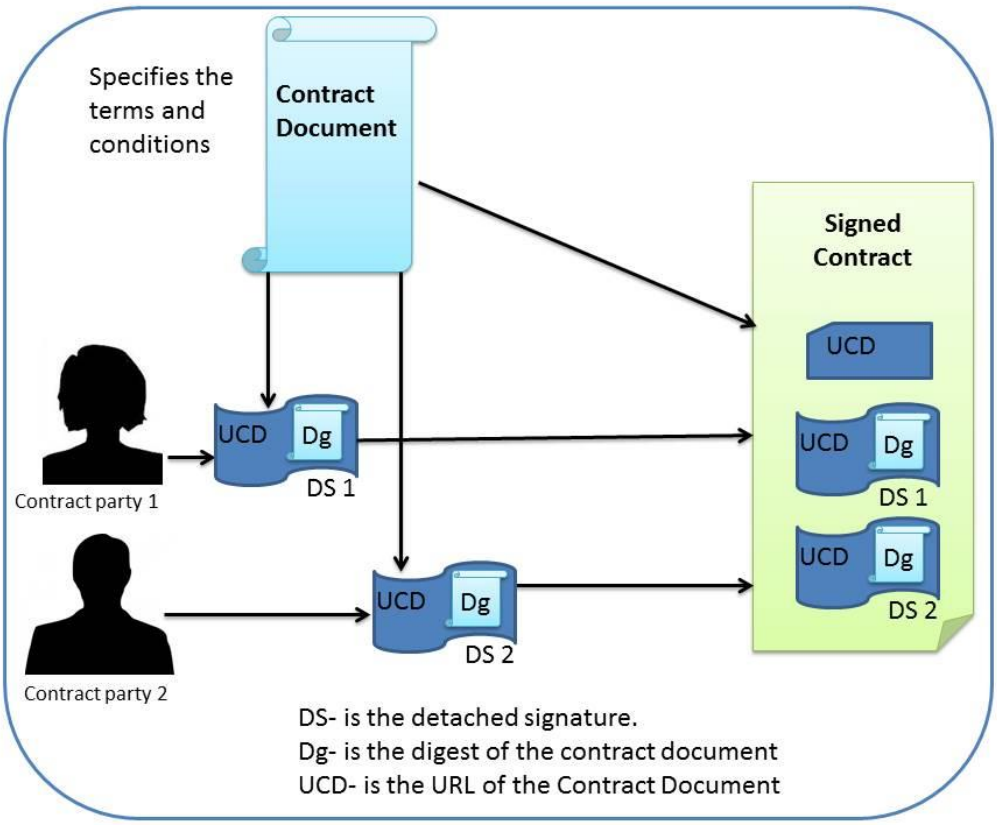

Figure 3-3. Construction of the signed contract

3.3.3.2 Construction of the contract document

Before defining the schema of the ***contract document*** we need to analyse the components needed in a contract document. Here we rationalise the attributes that we need for the ***contract document***.

The ***contract document*** ideally needs to identify who is allowed to access what data based on

the contract. Hence the ***contract document*** needs to identify the resource. Using a locally unique resource ID (RID, as mentioned in 3.3.2.5) would not be ideal in this case. To allow it to identify a personal data regardless of local identity of the resource the ***ResourceType*** (which identifies the type of the resource that is being allowed to be processed by this contract) and the ***SubjectOfContract*** (which identifies the subject whose data are being allows to be processed by the contract) are used. For example, to identify Mr. M's account information in a system holding different types of personal data, it firstly needs to determine which person's data is being identified and then the type of data. Hence the ***SubjectOfContract*** (e.g. the identifying attributes of Mr. M) and the ***ResourceType*** (e.g. account information) can identify a particular personal data without knowing the local resource identifier (RID) of the data. Note that, as ***IdentifyingAttributes*** of a person a set of attributes can be used, such as {{name} and {address}}, {NHS number}, {{role} and {organisation}}, {e-mail address}. If the ***SubjectOfContract*** element is absent in the contract document, the contract allows access to the type of resource mentioned in ***ResourceType*** in general.

The contract needs to identify who is authorised to access the resource by the contract. Hence an attribute named ***AuthorisedRequester*** is used which contains the identifying attributes of the persons who are allowed to process data due to the contract. The resource type mentioned in the contract is allowed to be processed only by the ***AuthorisedRequester***. This makes sure that no other person can access a personal data item just by presenting a contract, to access the mentioned data the requester has to be an ***AuthorisedRequester***.

The digital signature only tells who individually has signed the contract. To identify the parties of the contract, i.e. who are authorised to sign the contract, the identities of the parties also need to be mentioned in the contract document so that no one can simply present a contract signed by him/herself and get access to the data. If only signature verification is used anyone could sign over a contract and would get access to the data. Therefore, an element named ***PartyOfContract*** is used in the contract document that contains the identifying information of the parties, which consists of ***SignerOfContract*** and ***IdentifyingAttributesOfParty.*** Since a digital signature can only contain the ***Distinguishing Name (DN)*** of the signer the contract should contain the DN which is used to check if that of the contract and the signature matches to prove that the signer mentioned in the contract has signed it. Hence the ***SignerOfContract*** element contains a ***DN*** element which is of string type but its format should be of X.500 format. The signer of the contract can be the party himself (e.g. the data subject himself can be a party) or the signer can be an authorised person signing on behalf of an organisation (e.g. the authority of a bank can sign on behalf of the bank). The ***SignerOfContract*** element can contain an optional ***OtherIdentifyingAttributesOfSigner*** to indicate the identifying attributes of the signer other than the DN. For example, if the data subject is identified by his/her NHS number to a health care system and has signed a contract with the health insurance company the data subject needs to mention the NHS number as an ***OtherIdentifyingAttributesOfSigner*** element. The ConVS matches the DN mentioned in the contract's ***DN*** element with that of the digital signature to authenticate the person as a ***SignerOfContract,*** and the NHS number mentioned as the ***OtherIdentifyingAttributesOfSigner*** is used to identify the person as a data subject in the health care system.

The other element, ***IdentifyingAttributesOfParty*** is an optional element which contains the

attributes of the party in cases where the party is an organisation, and the signer signed the contract on their behalf. For example, an authority of a bank signs on behalf of the bank. In this case the ***DN*** element of the ***SignerOfContract*** has the DN of the signer (i.e. the DN of the bank authority), ***OtherIdentifyingAttributesOfSigner*** has the other identifying attributes of the signer such as the role and organisation of the authority (i.e. the role of the authority who signed the contract and the name of the bank as the organisation) and the ***IdentifyingAttributesOfParty*** contains the identity of the organisation (e.g. the name and address of the bank).

Last but not the least each contract should mention a validity time. Therefore, each contract document has a ***ValidityTime*** which comprises a ***StartDate*** and an ***EndDate***. The contract is not valid before the ***StartDate*** or beyond the ***EndDate***. The schema of the contract document is presented in Chapter 5.

3.3.3.3 Validation of contracts

While requesting access to personal data based on a contract, the requester can either provide the signed contract or the unique identifier if the system already has the contract in its repository (e.g., when the controller itself is a party or when a previous request has been made with the same contract the system should already have a copy). The system can validate a signed contract with the ConVS when it receives that for the first time and can store the retrieved information from the ConVS in the repository. Next time when a request comes with the same contract identifier it does not need to validate it again and can use the information from the repository. When the controller is a party of the contract, it also validates the signed copy to make sure the other parties have signed before storing the information in the repository.
The **signed contract** containing the URL of the contract document and the signatures of the parties over the contract document is passed to the ConVS. The validation process of the ConVS consists of four steps as mentioned in Figure 3-4. These are as follows-

i) **Validation of URL:** The contract document is presented in a URL reference. All the signing parties sign the contents referred to by the URL and the URL is also mentioned in the signature element (W3CXMLSignature 2008). The ConVS first checks whether the signatures are done on the same content i.e. all the signatures have the same URL mentioned in it. If the URLs mentioned in the signatures are not the same, it would mean that the signatures are not done on the same content and so an error message is returned by the ConVS.

ii) **Validation of schema:** The schema of the contract document obtained from the URL reference is matched against the schema of the contract document. If they don't match an error message is returned by the ConVS since it would mean all the essential elements of the contract document are not present.

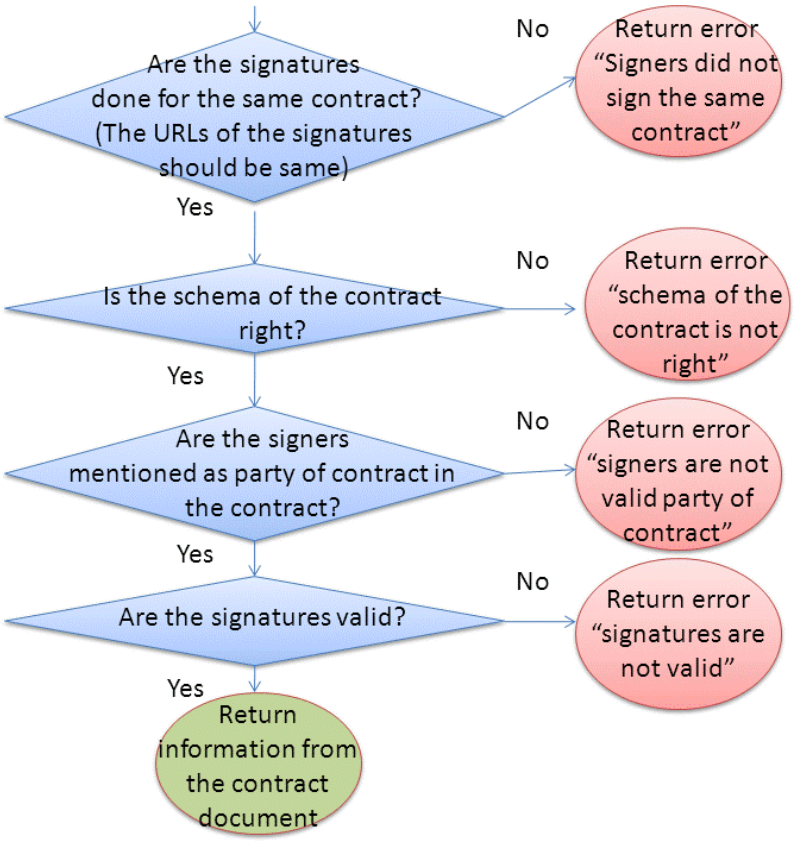

Figure 3-4. Flow chart of ConVS

iii) **Validation of parties:** The digital signature only tells who individually has signed the contract. To identify the parties of the contract, i.e. who are authorised to sign the contract, the identities of the parties are mentioned in the contract document and the ConVS checks the signer of the contract is in fact authorised to sign the contract. This checks if the DN of each of the signer is also mentioned in the ***SignerOfContract*** (as a part of the ***PartyOfContract***) of the contract document. The digital signature's DN element is matched with the DN of the signer of contract ***SignerOfContract***. If any of the signers is not mentioned in a ***SignerOfContract*** element then an error message will be returned by the ConVS.

iv) Validation of signatures: This portion determines the core validation of signature in the following two steps as mentioned in the W3C standard (W3CXMLSignature 2008).

Reference/ Contract document validation

a. Obtains the content from the URL of the actual contract document.
b. Digest the resulting data object using the digest method specified in the signature.
c. Compare the generated digest value against digest value present in the signature; if there is any mismatch, validation fails and returns error message.

Signature validation

a. Obtains the keying information element of the signature.

b. The signature is verified using the signature method and the keying information to confirm the signature is done over contract document.

If the validation passes all the above steps the ConVS returns the validity time, resource type and the identifying attributes of the subject of contract, authorised requesters, and all the parties of the contract. The PEP can format the received information according to its need and can store for a later use when sending to the authorisation service for authorisation.

3.3.3.4 How a request based on a contract is validated

The PEP has a contract repository containing information obtained from the validated contracts. When the PEP gets an access request with a signed contract or a unique identifier it first checks with the contract identifier whether it has already been validated and the information from the contract is present in the contract repository. If the contract has not been validated before (i.e. the information is not present in the repository) and it is passed the copy of the signed contract it calls the ConVS for validating the contract. If it validates, the ConVS passes the information from the contract back to the PEP, which stores it in the repository. The PEP then formats this information as environment attributes and adds this to the request context as described in Section 5.2 and appendix 7.

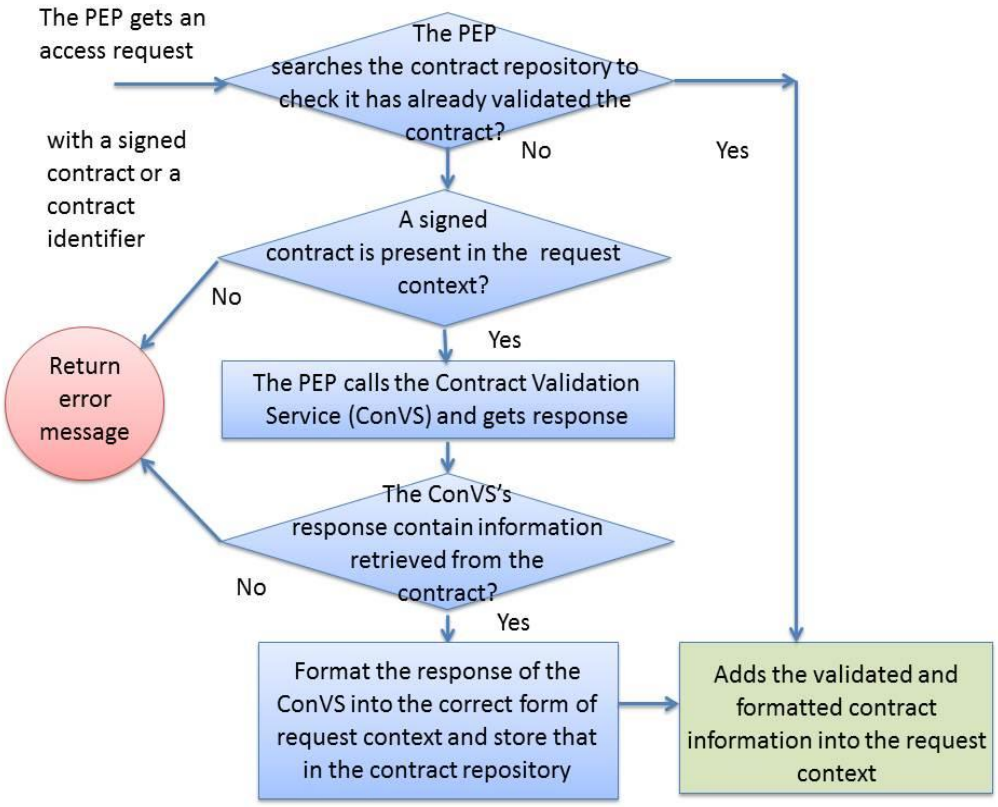

Figure 3-5. How PEP works with ConVS to validate a contract based access request

If the contract has not been validated before (i.e. the contract information is not present in the repository) and it is passed only a contract identifier and not a signed copy it just passes the

request context to the AIPEP ignoring the contract and does not add the contract information. The flow chart of Figure 3-5 shows how the PEP uses the ConVS for the validation of a request to access a personal data item based on a contract.

If it passes all the four validation steps of the ConVS as mentioned above the PEP receives information such as validity time, resource type, identifying attributes of the subject of contract, authorised requesters and the parties of the contract. The PEP then purses the response provided by the ConVS and forms attributes of the standard request element for authorisation service to understand and adds that to the original request context. The PEP also adds the resource attributes of the requested resource to the request context. The resource attributes contain information such as resource type, identifying attributes of the data subject and so on. It also passes the identifying attributes of the authorised signer (discussed in Section 3.3.3.6) of the controller as environment attributes to the request context before sending it to the AIPEP. Based on the request context the PDPs provide decisions and these are combined into one decision by the Master PDP. However, if an error is returned by the ConVS the PEP simply passes the request context to the next level (i.e. to AIPEP) ignoring the contract.

3.3.3.5 How to determine the data subject is a party of a contract

The EU DPD allows access to personal data based on a contract which is performed either with data subject or with the controller (as can be seen from the noted articles in 3.3.3). When the data subject is a party of a contract and has signed it, the DN of the data subject is mentioned as a ***DN*** element in the ***SignerOfContract*** element (as a part of the ***PartyOfContract***) of the contract document. The other identifying attributes of the data subject (e.g. NHS number) are mentioned as the ***OtherIdentifyingAttributesOfSigner***. For each personal data item the identifying attributes of the data subject are stored as resource attributes and when a request for a data item is received all the identifying attributes of the data subject are passed to the request context by the PEP. The policies in the PDP (specifically the Legal PDP) match the attributes mentioned in the ***OtherIdentifyingAttributesOfSigner*** with the data subject's identifying attributes passed with the request context to identify him/her as a party of the contract.

3.3.3.6 How to determine the controller is a party of a contract

A controller is usually an organisation that holds the personal data and controls the flow of that data. When the controller is a party of a contract an authority of the controller signs the contract with his/her DN on behalf of the controller and that DN is mentioned as a ***DN*** element in the ***SignerOfContract*** (as a part of ***PartyOfContract***) of the contract document. Any other optional identifying attributes of the signer are mentioned in the ***OtherIdentifyingAttributes-OfSigner*** element (as a part of ***PartyOfContract***). The ***IdentifyingAttributesOfParty*** is only used when the signer has signed the contract on behalf of an organisation and it contains the identifying attributes of the organisation such as the name and address. The PEP of the controller has a list of identifying attributes of the authorities of the controller who are authorised to sign a contract on behalf of the controller organisation. These identifying attributes of the authorised signer of the controller are passed as environment attributes with the request context by the PEP and are matched with the identifying attributes mentioned in

the **OtherIdentifyingAttributesOfSigner** elements to identify the controller as a party of the contract.

3.4 Dynamic Conflict Resolution

The P-PAAS infrastructure includes policies from multiple authorities such as the law, the issuer, data subject and controller (who is currently holding and controlling the flow of the data). A strategy that combines all these policies into a single 'super' policy may not be suitable for all situations, and such an example scenario is presented here. We propose a dynamic conflict resolution strategy that can integrate the decisions of the policy decision points (PDPs) evaluating the policies provided by the various parties. Using XACML (XACMLv2 2005, XACMLv3 2013), due to its static policy / policy set combining algorithm, it is not always sufficient to satisfy the policy needs of an organisation where multiple parties provide their own individual policies. Different conflict resolution strategies are often required for different situations. Thus combining one or more sets of policies into a single XACML 'super policy' that is evaluated by a single policy decision point (PDP), cannot always provide the correct authorisation decision, due to the static conflict resolution algorithms that have to be built in. We therefore, propose a dynamic conflict resolution strategy that chooses different conflict resolution algorithms based on the authorisation request context. The proposed system receives individual and independent policies, as well as conflict resolution rules, from different policy authors, but instead of combining these into one super policy with static conflict resolution rules, each policy is evaluated separately and the conflicts among their authorisation decisions are dynamically resolved using the conflict resolution algorithm that best matches the authorisation decision request. It further combines the obligations of independent policies returning similar decisions which XACML cannot do while keeping each author's policy intact.

3.4.1 Use case

Here we show the necessity for a dynamic conflict resolution strategy with a use case example, we consider a university which awards degrees and scholarships and maintains a profile for each student containing various personal data such as degree certificates, transcripts, and awarded scholarships. For personal data like degree certificates or transcripts, the university may want to deny access to anyone unless the data subject (i.e. the graduate) has specifically granted an access to the requester (for example, s/he can authorise a potential employer in his/her policy to access the degree certificates). For the scholarship awards, the university may want to publish this on its web site for marketing purposes, unless the data subject (i.e. the student) has specifically requested that the public be denied access to it. Since scholarships are usually regarded as achievements by students most of them will usually like to be honoured in this way, (and the university might also make it a condition of the scholarship that the award can be published except in exceptional circumstances). In the degree certificate case the conflict resolution rule will be GrantOverrides, since the issuer's (university's) policy denies access but the subject's policy may override this with a Grant decision. In the scholarship case, the conflict resolution rule will be DenyOverrides, since the issuer's (university's) policy grants access but the subject's policy may deny access. It is therefore not possible to effectively combine the issuer's policy for both types of resources with a subject's policy since two different conflict resolution rules are required, whilst XACML

only allows one conflict resolution rule to be applied to a set of policies. In order to implement this scenario in a single XACML super policy, both the subject's and issuer's policies would need to be dissected into their separate rules for each type or subtype of resources and then combined together per type or subtype of resource with separate conflict resolution rules. Depending upon the number of types or subtypes of a resource covered in any policy, this splitting and merging could get very complex. Furthermore it might be envisaged that different conflict resolution rules are needed for different actions or subjects (requesters) on the same resource, which would make the splitting even more complex. We conclude that in a single organisation, there may be the need for various policy conflict resolution strategies which are not possible to satisfy with one static XACML policy, without sacrificing the integrity of the individual policies provided by the different authorities. We therefore, propose a solution where each author's policy remains intact and is evaluated as it is, but the conflicts between policies are dynamically resolved based on a dynamically determined conflict resolution rule. In this motivating example, this means that if the access request is to read a degree certificate, the conflict resolution rule chosen by the issuer is GrantOverrides, but if the access request is to read the scholarship awards, the conflict resolution rule chosen by the same is DenyOverrides.

3.4.2 Conflict resolution strategy

Our authorisation system P-PAAS includes many different PDPs each with policies from different authorities and possibly written in different policy languages. As a consequence, a mechanism is needed to combine the decisions returned by these PDPs resolving any conflicts among them. Each PDP can return five different results – Grant (Permit), Deny, BTG, NotApplicable and Indeterminate. Where **Grant** (Permit) means the request is granted and **Deny** means the request is denied. **NotApplicable** means that the PDP has no policy covering the authorisation request. **Indeterminate** means that the request context is either mal-formed e.g. a String value is found in place of an Integer, or is missing some vital information so that the PDP does not currently know the answer. **BTG** (Break the Glass) (Ferreira, et al. 2009) means that the requester is currently not allowed access but can break the glass to gain access to the resource if s/he so wishes. In this case her/his activity is monitored and s/he is accountable for her/his actions. BTG provides a facility for emergency access.

We have introduced a component, the Master PDP, which is responsible for combining the decisions returned by the subordinate PDPs and resolving conflicts among their decisions. The system is enriched with an automated, sophisticated and dynamic conflict resolution strategy. The strategy is dynamic as it can choose the conflict resolution strategy dynamically depending on the request context. Each authority of a personal data item can provide a Conflict Resolution Policy (CRP) along with the Access Control Policy (ACP). Each CRP consists of Conflict Resolution Rules (CRRs). Each conflict resolution rule (CRR) comprises:

- a condition, which defines some logical relations among some attributes and values. The attributes and values presented in the request context are compared according to the relations defined in the conditions of the CRR. The evaluation of a condition is true if the attributes and values presented in the request context are evaluated to true according to the relations of those presented in the condition. The evaluation of condition of CRR is similar to the evaluation of condition element of XACML policy, to see if the attached decision combining

algorithm should be used,

- a decision combining algorithm (DCA),
- optionally, an ordering of policy authors (to be used by FirstApplicable DCA),
- an author and
- a time of creation.

A DCA can take one of five values: FirstApplicable, DenyOverrides, GrantOverrides, SpecificOverrides or MajorityWins which applies to the decisions returned by the subordinate PDPs. In other words the DCA (to be discussed shortly) is applied to the decisions returned by the PDPs of different authorities and a single final decision is obtained.

The system receives independent access control policies and conflict resolution policies from all policy authors i. e. from the law, issuer, data subject and controller. The Master PDP is also configured with a default CRP. The CRR of the default CRP is set by the administrator of the system during the initialisation time of the authorisation system and has no condition mentioned (so that it matches any request context) but only a DCA. The CRPs from different authorities are ordered based on a precedence rule and the law or Legal CRP has the highest priority so that no other authority can override the rights provided by the legislation. The next higher priority is given to the issuer of the personal data. For example, the university is the issuer of degree certificates, the bank of credit cards, and the person of personal data like personal diaries or personal choices like favourite drinks or personal information such as CVs. When the issuer withdraws the data, even the data subject (in cases where the data subject is different from the issuer) cannot access the personal data. In that sense the issuer is given the 2nd highest priority. The next higher priority is given to the data subject of the personal data and finally the controller of the data so that the controller cannot override the choice made by the issuer or the data subject. Here it is notable that only the CRPs of the different authorities are ordered according to this priority order. All the access control policies of all the authorities (expect for the FirstApplicable DCA) are evaluated and the conflicts among the decisions returned by the ACPs are resolved using the DCA of the chosen CRR. The default CRP and the fixed CRPs (i.e. the Legal CRP and the controller's CRP that do not change due to requests for various data) are read at program initialisation time, and additional CRPs are dynamically obtained from the subjects' and issuers' sticky policies.

Figure 3-6 shows the simplified form of the P-PAAS structure. Here the Application Dependent Policy Enforcement Point and the Application Independent Policy Enforcement Point are mentioned together as the Policy Enforcement Point (PEP) for simplification. When an access request is received by the PEP, (step 1 of Figure 3-6) it passes the request to the Master PDP (step2 of Figure 3-6). As there are a number of PDPs in the system the Master PDP needs to call only the ones related to the current request it is handling. The Master PDP finds the related PDPs searching with the RID (obtained from the request context) in the sticky store. After getting the request context and the information of the PDPs the Master PDPs first gets the CRR following the steps of Figure 3-7.

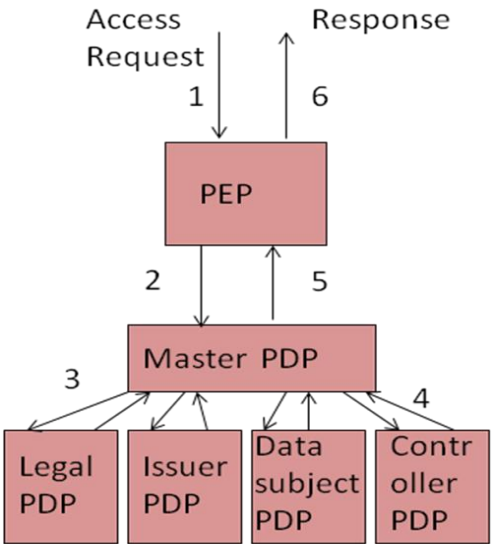

Figure 3-6. The P-PAAS system in a simplified form

The Master PDP has all the CRPs defined by different authorities as well as a default one. The CRPs are ordered according to the hierarchy of authorities (which is law, issuer, data subject and controller in order). For the same authority the CRRs are ordered according to the ToC (Time of Creation) so that the CRR with the latest creation time always comes first in the ordered list. For the default CRR the authorities and ToC contains no value and is placed at the end of the ordered list.

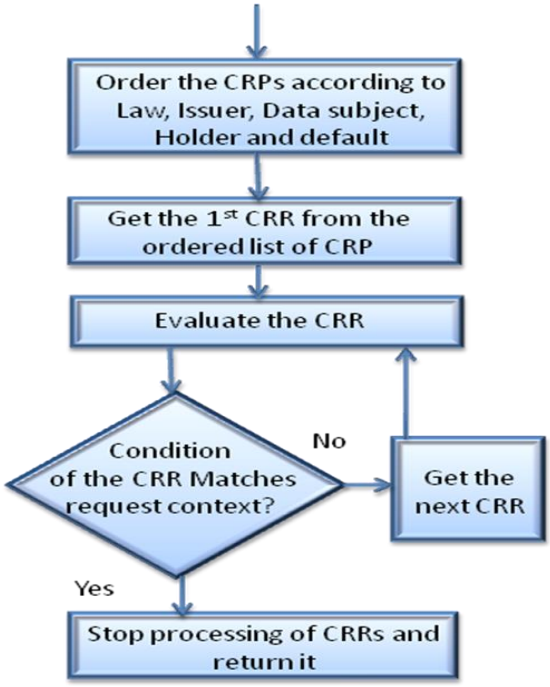

Figure 3-7. The process of selecting CRR by the Master PDP

All the conditions of a CRR need to match the request context for it to be applicable. The CRRs from the ordered queue are tested one by one against the request context. If a CRR's

conditions do not match the request context the next one from the queue is tested. If a CRR's conditions match the request context that one is chosen and so is the DCA of that CRR. After obtaining the DCA the PDPs are called (step 3 of Figure 3-6) and Master PDP gathers the responses from all the PDPs (step 4 of Figure 3-6). For different DCA the PDPs are called differently, for example, the FirstApplicable DCA requires the PDPs to be called in a provided order while other DCAs require no ordering, and the results returned by the PDPs are combined differently based on the value of the DCA.

We have proposed a merging strategy for combining obligations of the returned decisions. When a final decision returned by the Master PDP is Grant (or Deny) the obligations of all the PDPs returning a Grant (or Deny) result are merged to form the final set of obligations. The rational for combining obligations in such a way will be clearer with the examples provided in Section 3.4.3.2.

In our system, the policies by different authorities remain independent of each other and that also helps to enforce them when they are transferred to other systems along with the data. The receiving system does not need any extra effort to write an integrated complex policy combining the received policies with their organisation's policy; it only needs to start the independent PDPs. Initially, the system has the law and controller PDPs running as these two are common for all request contexts and, based on the request context the issuer and the data subject's PDP may be started.

Next we describe the Decision Combining Algorithms (DCAs).

3.4.2.1 FirstApplicable

If DCA=FirstApplicable the Master PDP calls each subordinate PDP in order and stops processing when the first Grant or Deny decision is obtained. If DCA=FirstApplicable the CRR is accompanied by a precedence rule (OrderOfAuthors of the CRR) which says the order in which to call the PDPs. For example, a CRR specifying a FirstApplicable DCA can be specified as follows, if (ResourceType=PII, requester=data subject) DCA=FirstApplicable, OrderOfAuthor=law, data subject, controller. The Master PDP calls each subordinate PDP in order (according to the order of authors), and stops processing when the first Grant or Deny decision is obtained. If no order of author is mentioned in the CRR the default order (the law, issuer, data subject and controller in order) is used. The flow chart for First applicable is given in Figure 3-8.

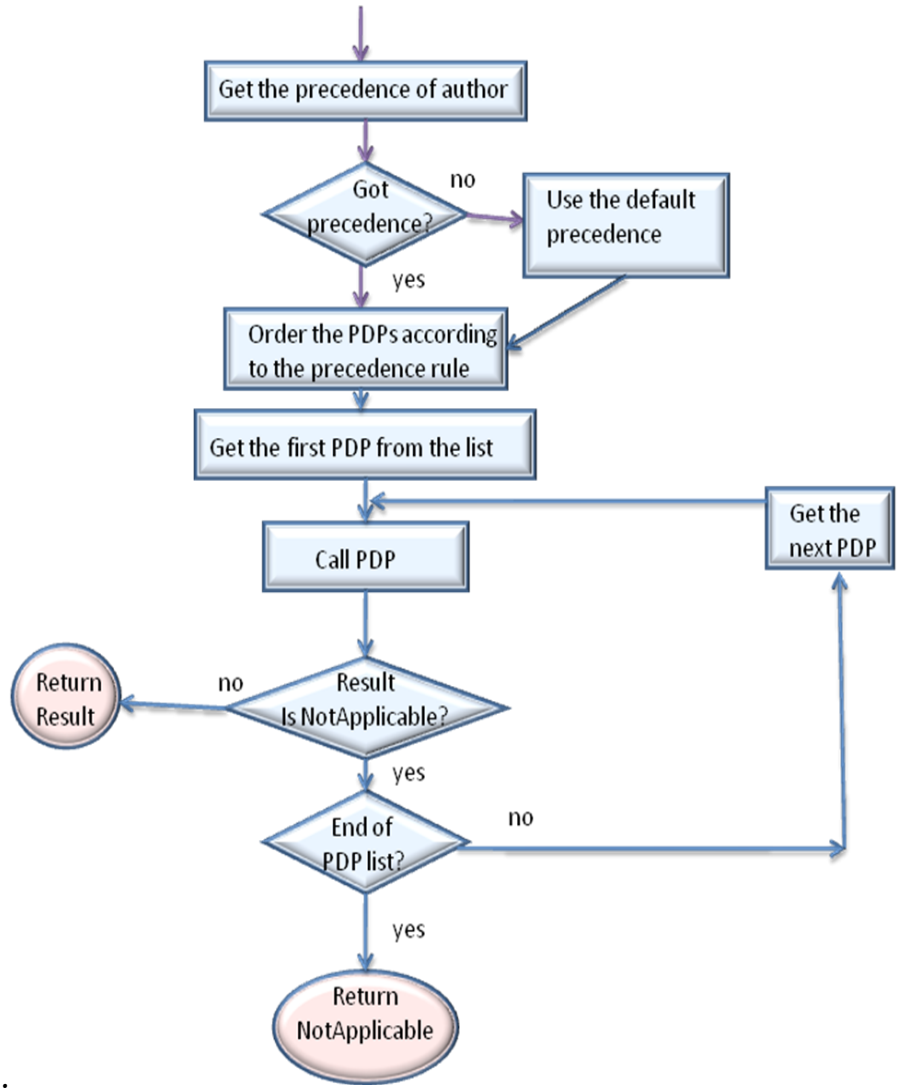

Figure 3-8. Flow chart of First Applicable

3.4.2.2 SpecificOverrides

For SpecificOverrides the decision returned by the most specific policy gets preference. A policy is defined to be the most specific if it is assigned to the most specific resource. An element x is more specific than y if all the members of x are also members of y but not vice versa (di Vimercati, Samarati and Jajodia 2005). When the resources are organised according to the containment hierarchy a resource x is more specific than y if the path of y is the prefix of that of x.

All the resources in the system, for example, can have their RID formatted using the tree based file structure such as Kent/CS/Module1/Result where the elements of the RID are separated by '/'. Each policy applicable to a resource is linked by its PID to that RID. In this containment model of resources, policies applied to a less specific resource are also applied to the more specific one, but not vice versa. If multiple most specific policies exist, all of those are evaluated and a Deny result gets precedence; in other words DenyOverrides is applied when SpecificOverrides cannot differentiate between multiple policies. It should be mentioned that the SpecificOverrides can only be applicable when the RIDs of the resources are formatted

using the containment model of resources.

3.4.2.3. DenyOverrides

For DenyOverrides the Master PDP calls all the subordinate PDPs. The flow chart of Deny Overrides is given in Figure 3-9. The precedence of decisions for DenyOverride is Deny>Indeterminate>BTG>Grant>NotApplicable.

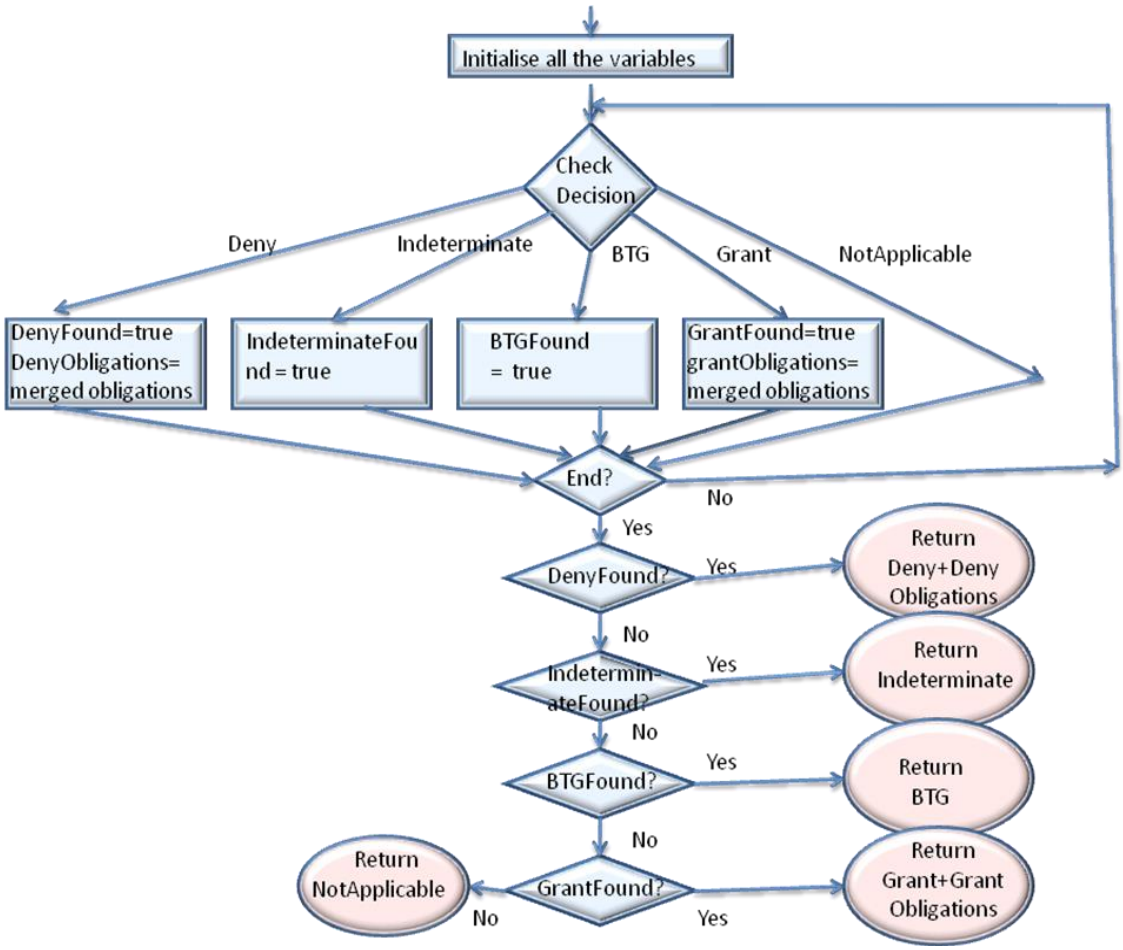

Figure 3-9. Flow chart of DenyOverrides

This means that, the Deny decision overrides all other decisions. If any PDP returns a Deny then it becomes the final decision returned by the Master PDP. If no other PDP returns a Deny and any PDP returns an Indeterminate then Indeterminate is the final decision returned. Similarly if no other PDP returns a Deny or Indeterminate and if any PDP returns a BTG then BTG is the final answer and if no other PDP returns Deny or Indeterminate or BTG and if any PDP returns Grant then Grant is the final decision. NotApplicable is given only if no PDP returns any decision. The rationale behind the precedence of decision for DenyOverrides is that it tries to provide the highest possible restriction on any access request. With the Deny decision no one can access the resource so it gets the highest priority for this conflict resolution strategy. The Indeterminate decision means that there is a problem with the request and no one gets access to the resource for this decision hence this gets second higher priority. The BTG decision does

not give access to the resource directly but it allows the requester to Break the Glass to gain access. So this is more restricted than Grant but less restricted than Indeterminate (since Indeterminate does not give access at all). Hence BTG stays between Grant and Indeterminate.

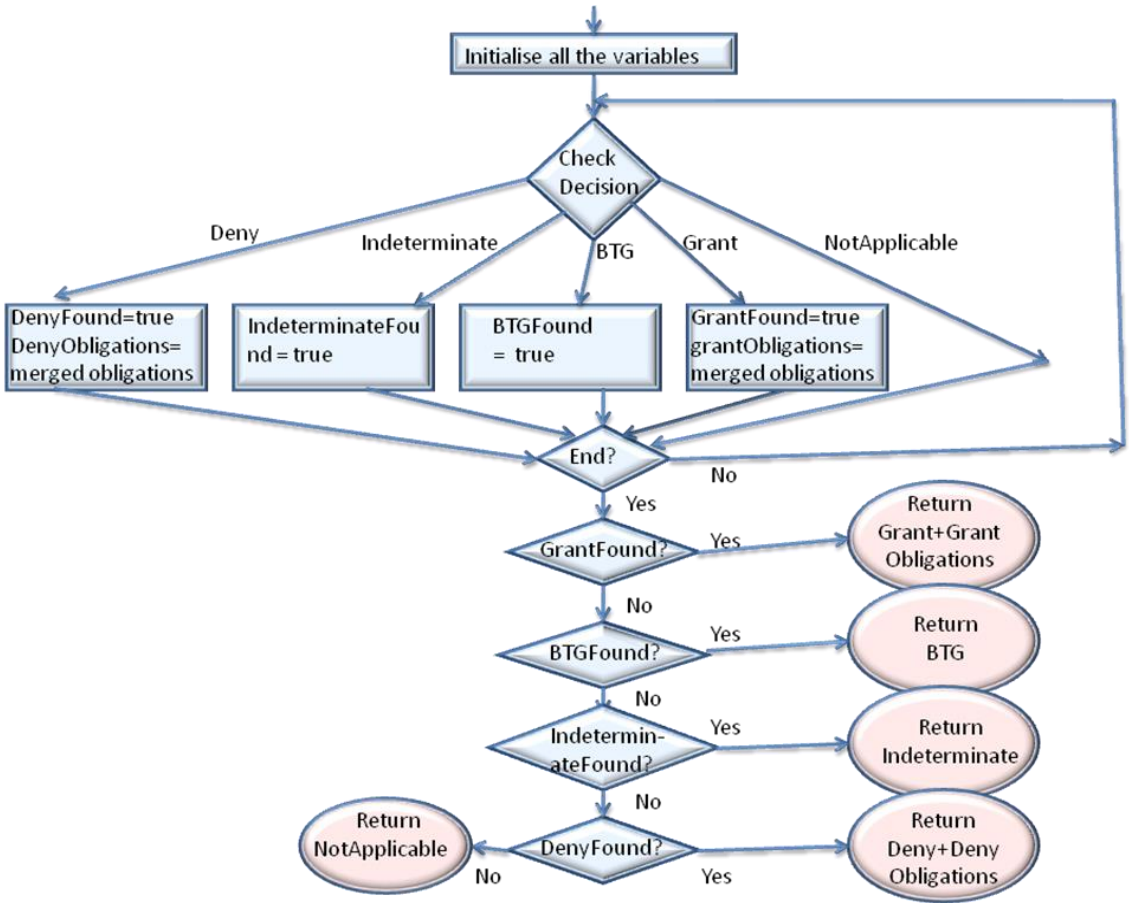

Figure 3-10. Flow chart of GrantOverrides

3.4.2.4 GrantOverrides

For GrantOverrides the Master PDP calls all the subordinate PDPs and the flow chart of GrantOverrides is given in Figure 3-10. The precedence of decisions for GrantOverrides is Grant>BTG>Indeterminate>Deny>NotApplicable and a Grant result overrides all other results. If any PDP returns Grant then it is the final decision returned by the Master PDP, but if no other PDP returns Grant and any PDP returns BTG then BTG is the final answer. Similarly if no other PDP returns Grant or BTG and if any PDP returns Indeterminate then Indeterminate is the final decision. If no other PDP returns Grant or BTG or Indeterminate and if any PDP returns Deny then Deny is the final decision. NotApplicable is returned only if no PDP returns any decision. The rationale behind the precedence of decision for GrantOverrides is that it tries to provide the highest possible flexibility on any request to access a resource. With the Grant decision the access to the resource is allowed, therefore, it gets the highest priority for this conflict resolution strategy. The BTG decision does not give access to the resource directly, but it allows the requester to Break the Glass to gain access. So this is less permissible than Grant but less restricted that Indeterminate (since Indeterminate does not give access at all). Hence BTG

stays between Grant and Indeterminate. The Indeterminate decision means that there is a problem with the request, and no one gets access to the resource for this decision but the access may be granted if the request is improved. With the Deny decision no one can access the resource and so it stays at the bottom of priority for the GrantOverrides DCA.

3.4.2.5 MajorityWins

For MajorityWins all the PDPs are called and the numbers of Grant, Deny and BTG responses are counted. The final decision (Grant/Deny/BTG) is the same as that returned by the majority number of PDPs. If there is no majority decision and the same numbers of PDPs return Grant, Deny or BTG then the precedence order is Deny>BTG>Grant. NotApplicable is returned if no other decision is returned by any other PDP. So, if the same numbers of PDP return Grant and BTG the final answer would be BTG. Similarly if they return Grant and Deny the outcome would be Deny. Finally if they return Deny and BTG the final answer would be Deny. If none of them return Grant/Deny/BTG but return Indeterminate, then that would be the final answer.

3.4.3 Comparison with XACML

In this section, the conflict resolution strategy of XACML (presented in Section 2.2.9) is compared with that of our system.

3.4.3.1 Policy creation and integration strategy

We first look into the strategy that can be taken to convert the previous example use case scenario into policies for our system and then the same into one XACML 'super' policy. While forming the conflict resolution policy the conflict resolution rules (CRRs) obtained from legislation (Fatema, Chadwick and Van Alsenoy 2011) come first, then come the CRRs from the issuer, then from the data subject and then from the controller and finally the default one.

For the scenario of our example the issuer of the student profile (the university) has 2 different CRRs to contribute to the conflict resolution policy. The CRRs of issuer are 1. If ResourceType=scholarship_info, DCA=DenyOverrides 2. If ResourceType=degree_certificate, DCA=GrantOverrides.
The issuer has policy saying "for ResourceType=scholarship_info, effect= Permit and for ResourceType=degree_certificate, effect = Deny"
Suppose that the data subject has a policy saying "for ResourceType=scholarship_info, ScholarshipType=hardship assistance, effect= Deny"
The data subject can have any conflict resolution rule and that CRR comes after the one of issuer on the ordered list of CRRs and so it is only evaluated if there is no CRR from that law or issuer matching the request context.
So the CRP has the CRRs in the following order

1. CRRs from the law (presented in Chapter 4)
2. CRRs from the issuer (CRR no. 1. If ResourceType=scholarship_info, DCA=DenyOverrides CRRs no. 2. If ResourceType=degree_certificate, DCA=GrantOverrides.)
3. CRRs from the data subject
4. CRRs from the controller

When a request to view a person's scholarship information arrives, the Master PDP evaluates the ordered list of CRRs. The law has no CRR regarding this. The next CRRs on the list are from the Issuer, thus the CRR no. 1 of issuer CRR matches the request context for which DCA=DenyOverrides. So the DCA=DenyOverrides is chosen by the Master PDP. Then the Master PDPs calls the independent PDPs of the law, issuer, data subject and controller. In this case the Legal PDP returns NotApplicable, the issuer PDP returns Grant. The data subject PDP returns Deny when the ScholarshipType=hardship assistance otherwise it returns NotApplicable. There is no controller PDP for this use case scenario as the controller and issuer are the same for this use case (i.e. the university). So in examples of ScholarshipType= hardship assistance the final result is Deny (according to the DenyOverrides DCA). For other types of scholarship information, the DCA remains the same i.e. the DenyOverrides. The Legal PDP returns NotApplicable, the issuer PDP returns Grant, data subject's PDP returns NotApplicable, and therefore, the final decision becomes Grant.

When a request to view a person's degree certificate arrives, the Master PDP evaluates the ordered list of CRRs. The law has no CRR regarding this (see Chapter 4) and the next CRRs are the ones from the issuer. Issuer CRR no 2 matches the request context which has a DCA=GrantOverrides, so it chooses the DCA=GrantOverrides. Then the Master PDPs calls the independent PDPs of law, issuer, data subject and controller. In this case the Legal PDP returns NotApplicable, the issuer PDP returns Deny and the data Subject PDP returns NotApplicable. So according to the GrantOverrides DCA the final result is Deny (unless the person specifically grants the access in his/ her policy)

Now if we try to combine the policies from the law, issuer, and data subject under one XACML PDP policy the policies from different authorities cannot remain independent anymore. In order to make sure that the law always has the highest priority the top policy combining algorithm cannot be Deny / GrantOverrides as the decision of Legal policy can be overridden by the other authorities. To give the Legal policy the highest priority the top combining algorithm needs to be first applicable with the Legal policy coming first. To implement our example policies the policies from the issuer and data subject need to be combined under DenyOverrides algorithm for one case (for ResourceType=scholarship_info) and under GrantOverrides algorithm for another (ResourceType=degree_certificate). That means it requires splitting the policies of different authors and then combining them and this may require manual interpretation and implementation depending on the needs of an organisation. In contrast, our system keeps the policies written by different authorities independent of each other, and makes it easier to travel to a new system and be configurable. The integration of policies from different authorities in one PDP will require a policy administrator to manually integrate them according to needs and it damages the integrity of the individual policies by splitting them.

3.4.3.2 Integration of Obligations

Each XACML policy document contains one Policy or PolicySet element as a root XML tag. A PolicySet can contain a number of Policies or PolicySets. A Policy represents a single access control policy, expressed through a set of Rules. Each Policy or PolicySet or Rule (for v3 only) element can define Obligations which can contain a number of Obligation elements. Each

Obligation element has an Obligation ID and a FulfillOn attribute. XACML's obligation combination strategy can be viewed as a vertical procedure where the Obligations of a contained Rule/Policy/PolicySet are combined with the Obligations of the containing Policy/PolicySet. An Obligation associated with a Rule or Policy is returned with a decision only if the effect of the Rule or Policy being evaluated is the same as the FulfillOn attribute of the Obligation. If the Policy is contained in a PolicySet, the Obligations associated with the PolicySet having a FulfillOn attribute value matching the effect of the PolicySet are combined with the returned Obligations of the contained Policy. For example, if PolicySet A has obligation 01 and it contains Policy A with obligation 02 and Policy B with obligation 03 then the final obligations returned could be 01 and 02 or 01 and 03 (assuming they all have the same FulfillOn attribute) depending upon the combining algorithm (see below) and the order in which Policy A and B are evaluated. This procedure continues recursively.

The limitations of the XACML obligations combining algorithm is that if a rule, policy or policy set is not evaluated then no obligations from them are returned to the PEP (XACMLv2 2005 (p 87), XACMLv3 2013 (p 82)). With XACML's GrantOverrides / DenyOverrides combining algorithms as soon as a Grant/ Deny decision is encountered the Grant / Deny is returned without evaluating the rest of the policies. Also with the FirstApplicable combining algorithm as soon as a decision (Grant/Deny) is obtained it is returned. This strategy of obligation combination may result in losing important obligations that ought to be returned. For example, if the controller's policy says that every time a Grant decision is returned, there is an obligation to "log the request" whilst the data subject's policy has a similar requirement that when a Grant decision is returned there is an obligation to "e-mail the data subject"; then if these policies are combined in a single XACML PolicySet with a GrantOverrides combining algorithm, then one of these obligations is always lost. In light of the above one can see that the integration of policies from different authorities into one 'super' XACML policy is more complex than simply splitting the policies of the different authors and then combining them together into PolicySets based on the resource type, as this may result in the loss of obligations.
In contrast to XACML, our system's policy evaluation and obligation combination strategy can be viewed as a horizontal procedure. In our system for both GrantOverrides and DenyOverrides all the PDPs are evaluated, and the final decision is chosen based on the DCA. The obligations that are returned by all the PDPs that have a decision equal to the final decision are combined. For example if the controller PDP returns a decision Grant with an obligation to "log the request" and the data subject's PDP returns a decision Grant with an obligation to "e-mail the data subject" and the final decision is Grant; then in our system the returned obligations are the combination of the obligations attached to the Grant decisions. If the policies are implemented in a single XACML PDP with either a GrantOverrides or DenyOverrides combining algorithm, the returned obligation(s) is only the obligation(s) attached to the policy that is encountered first and that contributed to the final decision.

3.5 How a Request is Processed in P-PAAS

The P-PAAS receives user's (or application's) requests via an application dependent interface. The format of the interface depends on the application and we do not discuss this here. The application handling user's interface converts the application request into a standard format

before sending the request context to the AIPEP via the PEP. However, we require that all resources are identified by a Resource identifier (RID) by the PEP. It is assumed that either the application requesting to access a resource specifies the RID or the PEP maps the requested resource to a particular RID in the system e.g. by database lookup to find the unique key of an entry. The application also maintains a database for storing the identifying attributes of a resource (e.g. name, RID, owner, location, type etc.) which are discussed in detail in Chapter 4. The application passes the resource attributes which are appropriate to the requested access, to the AIPEP in the request context via the PEP.

After receiving the request context from the PEP, the AIPEP first validates any subject credentials in the request, before passing the (possibly modified) request to the Master PDP. If the request context contains any sticky policy the AIPEP stores it in the policy store (discussed in Section 3.3.2.5) and stores the policy identifier (PID) along with the related information of the policy such as the policy author, language, type (authorisation policy / conflict resolution policy) and so on in the sticky-store before sending the request to the Master PDP. A request context containing sticky policies is presented in Appendix 5.
The Master PDP retrieves the relevant policies and policy related information from the sticky store with the RID from the request context. It then evaluates the conflict resolution policies (identified from the policy type information attached) in the sequence of law, issuer, data subject, controller and the default one (input at configuration time) and gets a decision combining algorithm (DCA) (explained in Section 3.4.2).
The Master PDP enforces all the authorisation policies retrieved with the RID (if any), and the law and controller policies which are specified during the configuration of the system, by calling the appropriate PDP object for each policy based on the policy language information specified. If the PEP's authorisation request does not contain a RID, then the Master PDP only calls the law and controller PDPs that the system is configured with, and does not call any sticky policy PDPs. The evaluation of each policy in a PDP depends on the policy engine it is configured with. How the PDP does this is of no concern to the Master PDP, since it assumes that every PDP knows precisely how to evaluate the policies that it has been written for (otherwise it would not be fit for purpose). The only requirement that is placed on a PDP is that it can support a standard request and response format. The Master PDP gathers the decisions and obligations returned by the PDPs. It then resolves any conflicts in these decisions, based on the previously selected DCR and returns all the obligations that are associated with the final decision to the AIPEP.
The AIPEP gets the decision with obligations and sees if it can enforce any of the "before" type ones (as discussed in Section 3.3.2.4). The AIPEP enforces an obligation if it has been configured to understand it; otherwise it returns that to the PEP. Since each obligation contains a globally unique ID, then it is trivial for the obligation engine to determine which ones it can enforce. The AIPEP can enforce application independent obligations such as sending an email to someone, attaching a sticky policy to the decision returned by the Master PDP, and writing information to a secure audit trail. How the PEP enforces the remaining obligations depends on the application.

3.6 Conclusion

In this chapter, the design of the P-PAAS is discussed. P-PAAS provides a generic infrastructure for providing privacy for various types of personal data based on privacy policies written by a number of authorities, possibly in different policy languages. Access to any data or resource of an organisation can be authorised by this system. The system is capable of storing policies and maintaining a link between the policy and the resource ID so that when a request for a resource (by the RID) is received all the policies related to that RID are consulted for making an authorisation decision. The system allows co–ordination of policies from multiple authorities (law, issuer, data subject and controller) written in different languages and resolving conflicts among them with a sophisticated, automated, dynamic conflict resolution strategy. The proposed strategy allows our system to choose a different combing algorithm based on the request context while allowing the policies written by different authorities to be separate and independent of each other. The separation of policies facilitates the easy integration of policies in a new system while the personal data are transferred to a different system along with them. Furthermore the proposed system allows us to integrate obligations (as a part of policies) written by different authorities which the current XACML policy combining strategy is unable to do. The system is integrated with a new component ConVS that facilitates the contract based access to personal data. Using this component some of the Legal data access rights mentioned in the EU DPD is made automated. The system also provides distributed enforcement of the privacy policies by transferring them along with the data and starting PDP with them so that they can be enforced while taking a decision on accessing that data. The system also enforces the application independent 'before' obligations (such as e-mail, secure audit trail) so that the application developer does not need to enforce them and thus it reduces the burden on them. Many other systems might have some of these features but to our knowledge no other system has a combination of all of these features. We do not claim to make each of the components better than other similar one (of fewer) system components, rather we claim that the combinations of all such features are not available in any other one system.

Chapter 4

4 Extraction of Machine Executable Rules From the EU DPD

4.1 Introduction

In order to ensure the enforcement of Legal rules our designed authorisation system, Privacy-Protecting Advanced Authorisation System (P-PAAS), has included a separate Legal PDP (Policy Decision Point) in its architecture. This Legal PDP contains all the authorisation rules obtained from legislation. Having a Legal PDP that is capable of giving automated authorisation decisions helps to enforce the legislative rights automatically. Furthermore, the separation of the Legal PDP helps to prioritise the enforcement of Legal rules so that no other PDP decisions can override the rights mentioned in the Legal PDP. In this chapter we present the research that has been performed on the extraction of the authorisation rules from one particular legislation, the EU DPD (European Union Data Protection Directive) (Directive 95/46/EC 1995), so that a PDP can evaluate those rules to provide automated decisions on behalf of the legislation. We conducted a manual examination of the EU DPD following a structured methodology which is described below, in order to obtain the automated access control or authorisation rules[2] from it. In this chapter, the methodology that we have used to obtain the authorisation rules from the EU DPD is described.

For the example of obtaining access control rules from legislation, we considered the EU DPD (Directive 95/46/EC 1995) as it contains all the important privacy features. Privacy legislation, on the other hand, of other regions such as the USA is not as strong as the EU DPD. In fact, in the USA there is neither comprehensive privacy law for the private sectors nor a supervisory authority to monitor the privacy protection which the EU DPD has (Fischer-Hubner 2001).

The strengths of the EU DPD are discussed in the review of the directive provided to the Information Commissioner's Office (Robinson, et al. 2009) and some of those strengths are:

- sets out the basic framework of data protection in a comprehensive manner.
- sets standards which are widely regarded as "high".
- sets out important and usable rights of humans.
- its principles are proven to be stable over time.

• is largely neutral in terms of technology.
• has helped to harmonise data protection rules across the European Union and provides an international reference model for good practice.

4.2 The Basics of the EU DPD

4.2.1 Historical background

The very first Data Protection Act has been adopted by the Parliament of the West German state Hessen which also serves as a basis for similar laws by the other German states (Fischer-Hubner 2001). The first national Data Protection Act is Sweden's Data Act which is passed in 1973. Privacy is first acknowledged as a fundamental human right by the Universal Declaration of Human Rights of the United Nations in 1974 (Assembly 1948). The privacy right is brought into a legal framework by the US privacy Act in 1974 (PublicLaw93-579 1974). The Organisation for Economic Cooperation and Development (OECD) first provides the *Guidelines on the Protection of Privacy and Trans border flow of Personal Data* with a view to helping harmonisation of national privacy legislation. They represent a consensus on the basic principles which can serve as a basis for national privacy legislation. The guidelines, in the form of a Recommendation by the Council of the OECD, are adopted in 1980. The council of Europe adopted the *Convention for the Protection of Individuals with regard to Automatic Processing of Personal Data* in 1981. The aim of the convention is to secure the individual's fundamental freedoms and right to privacy, with regard to automatic processing of personal data relating to him/her (EC 1981). The Council of Europe Convention and the OECD guidelines form the core of many data protection laws (Fischer-Hubner 2001). The European Community issued the first draft of the Directive on Personal Data Protection in order to protect the privacy of individuals. This is revised later and the final EU Directive 95/46/EC was adopted by the European Council in 1995[3]. The Directive mandates the member states to bring into force the laws, regulations and administrative provisions necessary to comply with this Directive.

4.2.2 Principles of the EU DPD

The EU DPD has the following data protection principles:

1. Personal data shall be processed fairly and lawfully.
2. Personal data shall be obtained only for one or more specified and lawful purposes, and shall only be processed for those purposes unless there is an informed consent from the data subject.
3. Personal data shall be adequate, relevant and not excessive in relation to the purpose or purposes for which they are processed.
4. Personal data shall be accurate and, where necessary, kept up to date.
5. Personal data processed for any purpose or purposes shall not be kept for longer than is necessary for that purpose or those purposes.
6. The data subject has right to access, to get notification and right to correction, erasure or blocking of incorrect or illegally obtained personal data.

[2] In this chapter the term "access control" and "authorisation" are used interchangeably.
[3] The EU DPD is currently being revised but the final revised version is not yet available. Furthermore,

7. Appropriate technical and organisational measures shall be taken against unauthorised or unlawful processing of personal data and against accidental loss or destruction of, or damage to, personal data.
8. Personal data shall not be transferred to a country or territory outside the European Economic Area unless that country or territory ensures an adequate level of protection for the rights and freedoms of data subjects in relation to the processing of personal data.

4.2.3 Structure of the EU DPD

The EU DPD is composed of seven chapters.

Chapter 1 (General Provisions) defines the objects and scopes of the directive and the definitions of the terms used.

Chapter 2 (General rules on the lawfulness of the processing of personal data) sets out the privacy principles such as the rules for purpose specification, keeping data accurate, criteria for making data processing legitimate, information to be given to the data subject, rights of a data subject and exemptions to the rights.

Chapter 3 (Judicial remedies, liability and sanctions) ensures the right of a person to receive compensation if s/he has suffered damage due to any processing incompatible with the national provisions adopted pursuant to the Directive.

Chapter 4 (Transfer of personal data to third countries) states the principles for transferring personal data; such as personal data can be transferred to the EU countries or to the countries having an adequate level of protection to personal data or when the data subject has unambiguously given consent to the transfer or the transfer is necessary to protect the vital interest of the data subject and so on.

Chapter 5 (Codes of conduct) provides instructions for drawing up a code of conduct for the proper implementation of the national provisions adopted by the member states pursuant to the Directive.

Chapter 6 (Supervisory authority and working party on the protection of individuals with regard to the processing of personal data) states the instruction for the appointment of one or more supervisory authorities and also mentions the duties, responsibilities and rights of the supervisory authority.

Chapter 7 (Community implementing measures) instructs the member states to ensure compliance with the Directive.

4.3 Methodology

The processing steps have been followed to extract the machine executable Legal access control rules from the EU DPD are:

Step1. *List the Legal provisions* that are directly related to authorisation. This step ensures that the legal rules which are not related to access control/ authorisation are eliminated from our consideration.

Step2. *Analyse the Legal provisions* obtained from step 1 with the aid of a legal expert to see if

no drafts were available at the time this research was performed.

they can form enforceable access control / authorisation rules (in natural language). In this step the Legal provisions are examined one by one in order to form a set of enforceable access control rules / authorisation rules from each of them. The Legal rules that are not capable of giving automated, independent decisions are discarded at this stage (see 4.5.2).

Step3. *Refine the natural language rules* by grouping similar rules together and ordering them in terms of their exceptions which need to be evaluated before the ones without. For example, data subjects are allowed unconditional access to their personal data that are held by a data controller, but not if law enforcement would be jeopardised by this. Consequently the rule which concerns law enforcement must be evaluated before the rule which grants the data subject unconditional access.

Step4. *Convert the natural language rules into a controlled natural language (CNL)* that is machine process-able. The grammar of the CNL is specified in Section 4.4. It allows an Access Control Rule (ACR) to be specified, which comprises a set of conditions on Subjects, Actions, Resources and the Environment, an Effect (Grant/Deny/BTG) and an optional set of Obligations. Alternatively a Conflict Resolution Rule (CRR) may be specified, which has the same terms as an ACR except that the Effect is always a Permit, and the Obligation is to always use a specific Decision Combining Algorithm (DCA). The rule conditions are specified in terms of attributes, and attribute names can contain any combination of String. Nevertheless, the conversion has to ensure that the same attributes are used in both request contexts and the specified rules (otherwise matching would never occur). A set of guidelines for attribute determination is given in Section 4.5.5.

Step5. Convert the controlled natural language rules into executable rules in XML[4].

Step6. Validate the obtained Legal rules in a PDP.

4.4 The Controlled Natural Language Grammar

The grammar for the natural language that is used in step 2 is described in Augmented Backus-Naur Form[5] (ABNF) below:

rule-definition=("ACR" wp rule-id wp ":" wp rule-statement ".") | ("CRR" wp rule-id wp ":" wp crr-statement ".") ;
crr-statement = "If" wp conditions wp "then" wp "DCR=" DCR;

rule-id = STRING;

*rule-statement = "If" wp conditions wp "then" wp GrantOrDeny wp *prep wp article wp actions *(wp prep wp) *(wp article wp) *(wp ResourceType wp) *(wp "with obligations to" wp obligations) ;*

conditions = (condition wp operator wp conditions) | (condition wp operator wp "(" wp conditions

[4]The actual schema that is used for the XML will depend upon the PDP that is used in the validation, so different programs will be needed for different PDP policy languages.

[5] http://www.ietf.org/rfc/rfc2234.txt

*wp ")" wp *(conditions)) | (condition);*

condition = (article attributes wp relationalOperator wp article attributes) | (wp "there is" article booleanAttributes) / (wp "there is no" wp booleanAttributes);

attributes = attribute | values ;

attribute = category ":" name ":" type;

category = "Subject"|"Resource"|"Action"|"Environment";

name= STRING;

type = "string"|"boolean"|"integer"|"double"|"time"| "date"|"dateTime";

article= (wp "a" wp) | (wp "an" wp) | (wp "the" wp) | (wp);

values= (value wp "|" wp values) | (value);

value= DQUOTE STRING DQUOTE;

relationalOperator= "is equal to" | "is "| "is not equal to"| "is not" | "is greater than" | "is less than";

operator= "AND"|"OR";

*actions = action *(wp "|" wp action);*

action = word;

ResourceType = word;

prep = "to"| "on"| "at" | "for";

booleanAttributes= (booleanAttribute wp "|" wp booleanAttributes) | (booleanAttribute);

booleanAttribute= category ":" name ":boolean" ;

obligations= (obligation wp "," wp obligations) | (obligation);

obligation=STRING;

DCR= "DenyOverrides" | "GrantOverrides" | "FirstApplicable" | "SpecificOverrides"| "MajorityWins";

GrantOrDeny = "Deny" | "Grant" |"BreakTheGlass";

word=(%x41-5A|%x61-7A|%x30-39);*

*STRING= *(%x41-5A|%x61-7A|%x30-39|%x20|%x2D|%x27|%x91|%x92);*

*wp = *(%x20 | %x09 | (%x0D %x0A));*

DQUOTE = %x22;

Where the notation "|" indicates alternate values and ***** indicates 0 or more repetitions.

4.5 Extracting the Rules From the EU DPD

4.5.1 Step 1. Listing the Legal provisions of the EU DPD related to authorisation

The EU DPD consists of seven chapters and 34 articles. Each article contains a number of paragraphs and sub paragraphs (which are referred to as Legal rules in this chapter). For obtaining the authorisation rules from the EU DPD the first step is to sort out the Legal rules that are related to authorisation.

A rule is directly related to authorisation if it pertains directly to the processing, prohibiting, accessing, collecting, blocking or transferring of personal data, i.e. it mentions an action on personal data. Each rule of each article of the EU DPD is examined one by one in step1 to see if it mentions an action on personal data or if it points to any other rule that mentions an action on personal data. If it does then the rule is kept for processing at the next step otherwise it is discarded. For example, Article 8.4 states that "Subject to the provision of suitable safeguards, Member States may, for reasons of substantial public interest, lay down exemptions in addition to those laid down in paragraph 2 either by national law or by decision of the supervisory authority." This rule does not mention any action on a personal data item but paragraph 2 that is pointed here mentions some actions on the personal data and so this rule is kept for processing at the next step.

4.5.2 Step 2. Analysing the Legal provisions

The nature of the EU DPD is such that the outcomes of rules are subject to explanations, built on "it all depends" upon context, and human interpretation at the point of application (M. Mont, S. Pearson, et al. 2010). It requires human skills and interpretation to obtain deterministic PDP rules from the EU DPD. Consequently in this step the rules obtained from step 1 are carefully examined one by one with the help of a legal expert, in order to try to convert them into automated authorisation rules. A rule can form an authorisation rule if it can satisfy either of the following conditions - i) the rule can express who is or not permitted to do what action on personal data under what condition/s or ii) the rule can express on what conditions an action can or cannot be performed on personal data. A provision is discarded if no enforceable access control/authorisation rules can be extracted for it or if all the extracted rules require human judgment in order to be enforceable i.e. they cannot be easily translated into a deterministic rule, so that a fully automated enforcement is essentially precluded. If an alternative set of rules can be obtained from the same provision then the set that provides the

most privacy protecting deterministic rules is chosen (see below). If a provision provides a mixed set of deterministic and human judgement rules, then the human judgement rules are discarded. If a rule does not form an automated authorisation rule the reason behind that is also discussed at this section. This step is the most time consuming and difficult in the methodology, and it cannot be done accurately or effectively without the help of a legal expert, since the rules often require interpretation and a nuanced understanding of their semantics.

The conversion of some rules is very complex. For example, the conversion of the Legal policy "Article 8.1: Member States shall prohibit the processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, trade-union membership, and the processing of data concerning health or sex life" can lead to three different possibilities-

Option 1: It can form a rule saying "if the request is for sensitive personal data (i.e. the data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, trade-union membership, and the data concerning health or sex life) then deny the request " (and a DCA for the same conditions is DenyOverrides). However, this is only appropriate if all the rules that state exceptions to this rule can be fully implemented. The Legal PDP's decision has the highest priority for any authorisation request. So a straight Deny decision for a request to access sensitive personal data is not appropriate if there are legal exemptions to this rule which could not be implemented in the Legal PDP.

Option 2: As the rule provides restrictions on processing sensitive personal data this can form a conflict resolution rule with a DenyOverrides DCA but no access control rule (i.e. the decision of the Legal access control PDP is NotApplicable) when a request comes for accessing sensitive personal data. The assumption here is twofold: either another PDP may deny the request and the DCA ensures that this takes precedence, or if all PDPs do not have a rule, then NotApplicable from all PDPs are converted into a Deny by the PEP. The problem with this approach is that if the data subject provides a Grant decision and the controller (or the issuer) says Deny the final decision becomes Deny. For example, a data subject provides a Grant decision to allow access to his medical record by his friend and the controller (or the issuer) may Deny it resulting in the friend being rejected. This may only be in conformity to the EU DPD depending upon national laws, since it says that national law can prohibit an access to personal data even when the data subject has given consent - "Article 8.2.(a): Paragraph 1 (Article 8.1) shall not apply where: the data subject has given his explicit consent to the processing of those data, except where the laws of the Member State provide that the prohibition referred to in paragraph 1 may not be lifted by the data subject's giving his consent;". Rejecting a third party even when the data subject has given consent might therefore not be illegal (depending on the national law).

Option 3: A final option is not to convert this Legal rule into any access control rule, and to leave the decision up to the other authorities. The problem with this is that the issuer can grant access in defiance of the EU DPD regardless of the subject's or controller's policies. Furthermore the controller can grant access if the subject does not have a policy.

Discussion: The first option is not suitable because all of the legal exceptions cannot be

implemented by this research, especially considering the fact that national legislation can produce more exceptions. Also the data subject's consent is one of the exceptions that need to be included in this rule, and this cannot be implemented (see below).

Option 3 is not suitable since it allows the issuer and controller to override the EU DPD.

Thus we are left with Option 2. The only negative effect of this option is that the data subject cannot always grant access to third parties, but the EU DPD does not say that the data subject should have this ability. Furthermore this option does empower the data subject always to be able to deny access.

The conversion of the Legal rule "Article 8.2.(a): Paragraph 1 (Article 8.1) shall not apply where: the data subject has given his explicit consent to the processing of those data, except where the laws of the Member State provide that the prohibition referred to in paragraph 1 may not be lifted by the data subject's giving his consent;" is also problematical, in that the data subject's consent in our system is captured in the PDP of the data subject. All the PDPs in our system are separate from each other and provide their decisions independently but in a set precedence order, where the Legal PDP comes first. Hence the Legal PDP cannot know the consent of the data subject when making its decision since the data subject's PDP is evaluated afterwards. Therefore, the conversion of the 1st part of this Legal rule can lead to the following two options.

Option 1: The 1st part of the Legal rule of article 8.2. (a) could be converted into an access control rule- "if the request is for sensitive personal data then the decision is Deny" along with a conflict resolution rule of SubjectOverridesDeny. The DCA=SubjectOverridesDeny means that only the data subject's PDP can override the Deny decision of the Legal PDP. So if the subject's PDP returns Grant the final decision is Grant otherwise the final decision remains Deny. The issuer's or controller's PDP is either never executed or their decisions are ignored.

Option 2: The 1st part of this Legal rule does not form any access control rule or conflict resolution rule. In this case the assumption is that the data subject provides conflict resolution rules and thus the data subject's choices override the controller's choices. Consequently the access to personal data is protected by the data subject's consent when there is no decision or DCA from the Legal PDP. However when there is no DCA returned by the Legal PDPs the issuer's conflict resolution rule is checked next (before the data subject's conflict resolution rule) and the issuer may be able to override the data subject's decision.

Discussion: The first option is not suitable as it ignores any local laws of Member States that may forbid the data subject from giving his consent. It also ignores the issuer's and controller's PDPs completely so they are not able to grant access to sensitive personal data such as for administrative reasons. The complete ignorance of these rules would not be appropriate.

The slight problem with option 2 is that the issuer can override the data subject. However, if we assume that the issuer of personal data is either the data subject (e.g. for personal statements) or a trusted authority such as a Medical Professional (for medical data) then these authorities can be trusted to provide appropriate priority to the data subject's choices or consent. Consequently we have chosen the second option. In Chapter 3 some use cases have

been provided where an issuer (a University authority) chooses different DCAs and decisions for different personal data and still takes care of the data subject's consent in an appropriate manner.

The results of examination of each of the rules that are obtained in step1 are presented in Table A1.1 in Appendix 1. The natural language rules that are obtained after completing processing of step 2 are presented in Table A1.2 in Appendix 1.

4.5.3 Step 3. Refining the natural language rule

In the previous step the Legal rules were analyzed sequentially. In the EU DPD the legal texts are written in a way that it first states the rule and then it states the exceptions to that rule. For example, the rule "the data subject is allowed to access his/her personal data" is stated first and then the exemptions to the rules are stated which include "the data subject is denied to access his/her personal data when there is a national security issue". These rules have matching subject, action and resource, whilst the conditions (i.e. having national security issue for the 2nd one) and effects (i.e. allow/ deny) might be different. The PDP, on the other hand, executes rules sequentially and as soon as it gets a decision (at least in the case of the "first applicable" decision combining algorithm) it returns that and does not execute the rest of the rules. If the above rules are kept in the order they appear in the legal texts, the PDP might allow access to personal data by the data subject even when the restrictions should apply. Consequently, the order in which the rules are written in the policy is very important for getting a correct decision. Therefore, we need to make sure that similar rules, i.e. rules having matching subject/ action/ resource, are correctly ordered. For the above example if the 1st rule allowing unconditional access to personal data to the data subject is placed at the bottom of this group, this ordering makes sure that the data subject is allowed access to his/her personal data only when no specific condition denying access is true.

Therefore, the natural language rules obtained from step 2 are refined during this step by grouping together similar rules and then ordering them to make sure the most general rule is placed at the bottom of the group. The ordering of rules is important only among similar rules. For example, if a rule says "the data subject can access his/her personal data" and another rule says "the treating Medical Professional can access the medical data" the ordering between these two rules will have no effect in the decision as the subject, action and resource are different for these two rules. Therefore, in this step we first group together only the similar rules. Similar rules are rules that have matching subject, action and resource. The term matching means the subject, action and resource of the two rules are the same or they contain the value "any" for subject, action or resource. This process is similar to the target matching of XACML. For example, if a rule says the data subject is allowed to transfer personal data and another rules says that any person is denied to transfer personal data, the two rules are similar according to the definition; as they have matching subject (one is data subject and another is "any"), action (both having the same action, transfer) and resource (both having same resource, personal data). Therefore, the rules that allow or deny access to personal data by the data subject (rules having the same subject i.e. data subject, action and resource; rules 14 and 16 of Table A1.2 of Appendix 1) are grouped together and the ones that allow/deny the transfer of personal data (rules 18-25 of Table A1.2 of Appendix 1) are grouped together. These

rules have various subjects with one rule having "any" for subject, the same action (i.e. transfer) and resource (i.e. personal data). The rules within the group are then ordered according to the criteria mentioned above i.e. placing the rule that mentions no condition at the bottom of the group. When there are a number of conditions among the rules, the rules should be organized according to the specificity of conditions, with the most specific coming first. The conditions in a rule 'A' are more specific than the conditions of a rule 'B', if the conditions (i.e. the name, value pairs) of 'B' are contained in the conditions of 'A'.

4.5.4 Step 4. Convert into a Controlled Natural Language (CNL)

At this step each of the natural language rules (except one that forms only a CRR, rule 10 of Table A 1. 2 of Appendix 1) is formalised in the form of an Access Control Rule (ACR) and a Conflict Resolution Rule (CRR) as follows:

ACR – If *condition* (under what conditions) then *effect* (Permit, Deny or BTG) the *action* on the *ResourceType* with an optional *obligation* (subject to these actions being carried out);

CRR – If *condition* (under what conditions) then *effect* (Permit) the *action* on the *ResourceType* with an *Obligation* (Decision Combining Algorithm (DCA) to be returned).

Each Legal ACR rule is also converted into a matching CRR to make sure that the Legal rule gets precedence over any other authority's rules. The difference between the CRR and its corresponding ACR is that the effect of the CRR is always a Permit and the obligation always returns the DCA that is applicable. If an ACR has an effect of Deny, the corresponding DCA is DenyOverrides and if an ACR has an effect of Permit the corresponding DCA is GrantOverrides. If the ACR has an effect of BTG, the CRR's DCA is GrantOverrides, since a Grant from another PDP should not require the requester to first break the glass before gaining access.

The *condition* element of the CNL rule defines the relationship of *attributes* with other *attributes* or *values*. Each *attribute* is defined with a combination of *category* (i.e. the kind of attributes: Subject, Resource, Action, Environment), *name* (i.e. the name of the attribute) and *type* (e.g. string, boolean, integer, double, time, date and dateTime). A guideline for determining *attributes* for converting the NL rules into CNL is given in Section 4.5.5. The converted CNL rules are presented in Table A1.3 of Appendix 1.

4.5.5 Guidelines for attribute determination for policies

For the automated execution of rules in an attribute based access control (ABAC) system we need to determine the attributes of each of the elements in the rules. Four different types of attribute are used in constructing the policy rules. 1. Subject attributes 2. Action attributes 3. Resource attributes 4. Environment attributes. Subject attributes identify the users who are to be granted or denied access. Action attributes describe the actions that are being controlled. Resource attributes describe the features of the protected resources (i.e. the personal data), such as: resource type, data issuer, data subject, date of creation etc. Environmental attributes describe the context in which the rule applies, such as the time of a day, location etc. These four types of attribute are also used to describe a user's request to access a resource, and are passed to the PDP in the request context by the application. PDPs compare the attributes of

the user's request with those of the rules to determine whether an access should be granted or not. Each resource in the system is identified by a unique RID (Resource identifier) (see Section 3.3.2.5). Each resource has some attributes associated with it and these attributes are stored safely with the resource (the personal data) and added by the PEP (Policy Enforcement Point) to the request context and passed to the PDP when a request for accessing it is received.

o **The data subject** is determined based on his/her set of identifying attributes (such as name and address, e-mail address, NI number, NHS Number etc.) given at the time the personal data are submitted or during the registration of the subject with the controller for a service. These identifying attributes are stored as resource's attributes of the resource to which the data subject is related. The Legal PDP checks if these identifying attributes match with those of the requester (passed with the request context as subject attributes) to determine whether the requester is the data subject or not for the requested resource. In the current implementation of Legal rules the following sets of uniquely identifying attributes are used: {{name} and {address}}, {e-mail address}, or {NHS Number}, but these sets are configurable and can be changed and extended as needed by the application. The data subject should be able to choose any of these to identify her/himself.

o **ResourceType** is a resource attribute that holds the type of the data, such as medical data, and is placed as an attribute of the resource by the issuer of the data. Only the issuer can modify the ResourceType of that data. An ontology is needed to classify the different types of personal data, and an ontology mapping server (e.g. as described in (Fatema, Chadwick and Lievens 2011)) may be used to hold it and be able to determine whether a resource type is a type of personal data or not. The ontology server may also determine the relationship among these data types; for example all the medical data types are subclasses of personal data.

o **PurposeOfCollection** (mentioned in rule 1 of Table A1.2 and A1.3 of Appendix 1) is another resource attribute that states the set of purposes for which the data were collected from the data subject. It is set by the application when the data are first collected from the data subject. The Legal PDP matches this set with the purposes stated by the requester in the request context.

o **ValidityTime** (mentioned in rule 3 of Table A1.2 and A1.3.) is another resource attribute collected from the issuer or data subject. A default value can also be set by the controller if the issuer or data subject does not provide a value for it. The controller needs to mention the default validity time of the data when collecting them. The Legal PDP matches the time of the access request (passed as an environment attribute of the request context) with the ValidityTime of the requested data.

o **Treating Medical Professional** (mentioned in rule 12 of Table A1.2 and rule 11 of Table A1.3) is identified by an identifying attribute (e.g. PhysicianID) stored in the medical record of the patient (as a resource attribute). The value of this attribute must match that of the equivalent attribute of the requester, in order for the requester to be identified as the Treating Medical Professional. The name of this attribute is configurable in the Legal policy.

o **Social Security Authority, Medical Professional/ Supervisory Authority** are Role attributes (mentioned in rules 8, 9, 17, 23, 26 and 27 of Table A1.2 and rules 8, 9, 15, 17 and 18 of

Table A1.3) provided by the trusted Attribute Authorities. Who are the trusted authorities for which roles depends upon the application, and these are configurable values in the Legal policy.

o **LegalObjection, MedicalObjection, NationalSecurityIssue and Economic/ FinancialIssue** (mentioned in the rule 16 of Table A1.2 and rule 14 of Table A1.3) are Boolean attributes of a resource which are used to flag the personal data that are not accessible to the data subject due to the national legislation which contains an exception to the data subject's right of access. For example, a doctor may have the ability to invoke a MedicalObjection to prevent the patient from accessing certain information; or there is legal issue such that seeing the data by the data subject may harm a legal process such as a legal investigation. These attributes can only be issued by the designated (trusted) authorities. Note: If only one attribute (e.g. LegalObjection) was used there might be a situation that a LegalObjection set by one authority for one purpose would be overridden by another authority for a different purpose.

o **DataAccessMandate/ DataTransferMandate** (mentioned in rules 7 and 22 of Table A1.2 and rules 7 and 16e of Table A1.3) are credentials which can only be obtained by a requester following the appropriate legal procedure. Conceptually these are treated as subject attributes in the policy, so that if a requester possesses the appropriate mandate attribute he/she inherits the permissions assigned to the mandate (the DataAccessMandate is assigned for allowing access to personal data and the DataTransferMandate is assigned for allowing the transfer of personal data). These legal mandates are issued by various trusted Attribute Authorities and both the authorities and mandate types are configurable to suit the application. The requester (or the Attribute Authority) presents the Mandate to the application which either verifies it using a Credential Validation Service and passes the valid attribute to the PDP as a subject attribute, or passes it to the PDP as a subject credential for the latter to verify.

o **PartyOfContract and SubjectOfContract, AuthorisedRequester** (mentioned in rules 5,19 and 20 of Table A1.2 and rules 5 and 16 of Table A1.3) are attributes of a contract. A contract is defined to be a digitally signed XML document which has an element called PartyOfContract containing the identifying attributes (IDAs) of the people who have signed it and information of the organisation who are parties of the contract, and an element called SubjectOfContract containing the IDAs of the data subject. AuthorisedRequester contains the IDAs of the persons who are allowed to access the data due to a contract. When a requester wants to access a personal data item for a purpose related to the performance of a contract, s/he presents the contract or the unique contract identifier to the system if the system already has that (in cases where the controller is also a party of the contract then the controller's system should have the contract in its repository). For validating contracts, a trusted component called the Contract Validation Service (ConVS) is added to the system (discussed in detail in Chapter 3). If a contract is valid the ConVS passes information such as validity time, resource type and the identifying attributes of the authorised requester, parties and subject of the contract. This information is used by the Legal PDP while authorising an access request based on a contract.

o **SubjectConsentsToTransferTo** (mentioned in rule 18 of Table A1.2 and rule 16 of Table A1.3) is an environment attribute set by the application to the IDA of the requester when the data subject consents to transfer his/her personal data to a requester. A requester can send a request for consent (via the application) to the data subject for transferring his/her personal data. If the data subject agrees to the transfer s/he can give his/her consent via the application (e.g. by clicking a button or ticking a box). This consent is stored by the application and when the requester requests the data this consent (in the form of SubjectConsentsToTransferTo environment attribute) is appended to the request context by the application.

o **SubjectRequestedToProcess and AllowedPartyToProcessData** (mentioned in rule 6 of Table A1.2 and rule 6 of Table A1.3) these two attributes are used to indicate that the data subject of a personal data has allowed a person or party to process a personal data item prior to entering into a contract. These two attributes can be obtained from a digitally signed document by the data subject or can be obtained by the application (via a process similar to obtaining the previous attribute). The attribute SubjectRequestedToProcess contains the type of data the data subject is allowing to process and this is matched against the requested resource type. The attribute AllowedPartyToProcessData contains the IDA of the person or party the data subject is allowing to process the data for the purpose of entering into a contract. This value is matched against the IDA presented by the requester.

o **SubjectRequestedToTransfer and AllowedPartyToTransferData** (mentioned in rule 21 of Table A1.2 and rule 16 of Table A1.3) these two attributes are used to indicate that the data subject of a personal data has allowed a person or party to transfer a personal data item for the implementation of the pre contractual measure. These two attributes can be obtained from a digitally signed document by the data subject or can be obtained by the application (via a process similar to obtaining the previous attribute). The attribute SubjectRequestedToTransfer contains the type of data the data subject is allowing to process and this is matched against the requested resource type. The attribute AllowedPartyToTransferData contains the IDA of the person or party the data subject is allowing transferring the data for the purpose of entering into a contract. This value is matched against the IDA presented by the requester.

o **AdequateSafeguard** (mentioned in rule 25 of Table A1.2 and rule 16 of Table A1.3) is another contextual attribute used by the current controller to indicate that the destination controller has agreed to provide adequate safeguards to the personal data when the transfer of personal data is being made to a country not having an adequate level of protection. For example, the current controller may have a contract with the destination controller stating that adequate safeguards will be provided.

4.5.6 Step 5. Converting CNL to PDP rules

Legal Access Control Policy and Legal Conflict Resolution Policy have been converted into machine executable policies using both the XACML and PERMIS policy languages manually following the steps presented in Section 5.3.1, 5.3.2 and 5.3.4. The Rules have also been converted into XACML automatically by a Java program named *XACMLConverter* which is written following the steps presented in Section 5.3.3.

4.5.7 Step 6. Validation

A) Validation of the PDP rules

This procedure ensures that all the manually converted XACML rules produce correct answers from the PDP, according to human judgement. See Section 5.4 and Appendix 2.

B) Validation of the automatic conversion to XACML

The manually produced and validated (by the above step) XACML rules are compared automatically with a program line by line with the XACML rules produced by the *XACMLConverter* to make sure that they are an exact match.

4.5 Conclusion and Discussion

We have presented a semi-automated procedure to construct automated access control/ authorisation rules from the EU DPD. The rules balance the right of a data subject to access his/her personal information, as well as, the rights of others to forbid this access or to gain access to the personal data for a legitimate purpose. The Legal conflict resolution rules help to ensure the precedence of the Legal access control rules over that of any other stakeholders. The Legal PDP, however, does not completely capture all the legal constraints. Some conditions are extremely complex to automate and some decisions of judgment are highly dependent on human intervention.

From the 53 rules of the EU DPD that were considered for analysis in step 2 (since they mentioned some actions on personal data) 27 of them could contribute to the construction of enforceable authorisation rules. However, 14 rules among these 53 are found to be guidelines or instructions only and there are no means to have authorisation rules from them. For example, Article 6.1.(a) states that personal data must be processed fairly and lawfully, Article 17.1 says that controller must implement appropriate technical and organisational measure to protect personal data against accidental or unlawful destruction or accidental loss, while Article 25.2 instructs about what to consider for determining whether a third country has adequate level of protections. These guidelines set out the overall requirements for privacy protection. Access control is one of the important aspects of privacy protection but it does not cover all such aspects. For example, Access control mechanisms are not capable of protecting against the accidental loss of data. This thesis attempts to cover the areas of privacy protection only within the boundary of access control. However three other rules can be supported by the system design, which are:

1. The requirement that only data that is 'adequate and relevant' is released, can be ensured by making the granularity of the RID appropriate for the application, as access is granted only for the requested RID by our system.

2. Consent of the data subject can be obtained by having sticky policies from him/her, and

3. The requirement for providing notification to the data subject can be fulfilled by the use of an appropriate obligation in the subject's sticky policy.

The remaining 9 rules are found to be too dependent on other laws or human judgement to be turned into access control rules by themselves, for example, Article 8.2(b) requires employment law to be examined to determine the conditions for data release. Whilst Article 13.2 requires human judgement to determine whether the data are being used for "taking decisions regarding the particular individual". Article 7(f) "processing of personal data for legitimate interest are allowed except where such interests are overridden by the fundamental rights and freedom of data subject" presents an extremely complex condition where the balance of interests are not feasible to be presented in an access control policy.

The proposed methodology is a proof of concept based on the EU DPD. Several of its directives may require far more complex access control rules to be formed due to divergences in national or sector specific law (e.g. more specific rules are available for medical data). Nevertheless we believe our methodology can be employed with these national or sector specific laws to produce enforceable access control rules.

From the literature, it is evident that no other previous work attempted to obtain access control rules from the EU DPD. Although some other previous works attempted to get requirements from legal texts, the processes are highly dependent on the nature of the representation of the legislative language (Breaux and Anton 2008). The main limitation of our extraction process is that due to the nature of the EU DPD the analysis of the rules highly depends on the knowledge of the human expert. Even with the limitations of our methodology given above, having enforceable access control rules from legislation will reduce the effort of ensuring compliance (Papanikolaou, Pearson and Mont 2011). Although the full EU DPD cannot be covered with access control rules, the obtained machine executable rules will help legal compliance to a great extent.

Chapter 5

5 Implementation, Validation and Testing

5.1 Introduction

This chapter describes the implementation of the authorisation system, and the Legal access control and conflict resolution policies. It further presents the validation tests of the same. For the validation tests of the Legal policies, a set of test cases has been constructed and for each of the requests the outcomes of the Legal access control PDP and conflict resolution PDP are compared with the expected ones determined by human judgement. The validation tests of the system have been conducted based on various use case scenarios which involves policies from the multiple authorities. Finally, the performance of the system has also been measured, and the results are presented here.

5.2 Implementation of the System

5.2.1 Implementation and configuration of the authorisation system

The high level conceptual mode described in Chapter 3 treats the authorisation service as a black box with a set of defined functionalities. The most important component at this level is the specification of the standard protocol that will be used to communicate with the authorisation service. The application developers will use this protocol for sending the request to the authorisation system (assumption 5 of Section 3.2). The policy needs to be passed dynamically along with the decision request to the authorisation service so that the data subject, or the application acting on her behalf, does not need to access the PAP for storing the policy prior to the authorisation decision. The chosen protocol should be able to pass policies in any policy language (assumption 1 of Section 3.2) along with the request context.

The inability of the XACML request/response context to pass policies makes this OASIS standard (XACMLv2 2005, XACMLv3 2013) unsuitable for this purpose. However, the SAML 2.0 Profile of XACML, Version 2.0, Committee Specification 01, 10 August 2010 (SAML2.0 2010) does allow policies and credentials to be securely transferred with a request context, so this was chosen as the best fit for our requirements. However, this protocol has one major drawback in that it only supports policies written in the XACML policy language, and we required support for multiple policy languages. Professor Chadwick, who is a member of OASIS, therefore entered into discussions with the OASIS XACML working group about adding support for multiple policy languages to the SAML 2.0 Profile. After much discussion the

working group agreed in 2011 to add this support to the SAML 2.0 profile by adding an extension to it (see Table 5.1). The revised Committee Specification was eventually published in August 2014 (SAMLv2.0 2014).

Table 5.1: Schema for SAML 2.0 elements used for passing policy to AIPEP (2014 version)

http://docs.oasis-open.org/xacml/3.0/xacml-3.0-profile-saml2.0-v2-schema-protocol-wd-13.xsd

```
<element name="XACMLAuthzDecisionQuery" xsi:type="xacml-samlp:XACMLAuthzDecisionQueryType"/>
<complexType name="XACMLAuthzDecisionQueryType">
<complexContent>
<extension base="samlp:RequestAbstractType">
<sequence>
<element ref="xacml-context:Request"/>
<element ref="xacml-samlp:AdditionalAttributes" minOccurs="0" maxOccurs="1"/>
<element ref="xacml:Policy" minOccurs="0" maxOccurs="unbounded"/>
<element ref="xacml:PolicySet" minOccurs="0" maxOccurs="unbounded"/>
<element ref="xacml-saml:ReferencedPolicies" minOccurs="0" maxOccurs="1"/>
<element ref="xacml-samlp:Extensions" minOccurs="0"/>
</sequence>
<attribute name="InputContextOnly" type="boolean" use="optional" default="false"/>
<attribute name="ReturnContext" type="boolean" use="optional" default="false"/>
<attribute name="CombinePolicies" type="boolean" use="optional" default="true"/>
</extension>
</complexContent>
</complexType>

<element name="Extensions" xsi:type="xacml-samlp:ExtensionsType"/>
<complexType name="ExtensionsType">
<sequence>
<any namespace="##any" processContents="strict" minOccurs="0" maxOccurs="unbounded"/>
</sequence>
</complexType>
```

In the SAML profile of XACML a new element called XACMLAuthzDecisionQuery has been defined which allows the PEP to combine the XACML authorisation decision request with the policy that is to be used by the PDP. This facility is helpful for enforcing sticky privacy policies where the PDP needs to be told which privacy policy to use for taking the access decision. The new Extensions element, shown in bold in Table 5.1, allows policies in languages other than XACML to be passed along with the request context. The SAML assertion XACMLAuthzDecisionStatementType (not shown) allows the XACML response context to optionally contain obligations and to be returned as a SAML response. This also facilitates the enforcement of privacy by including obligations with the access decision.

Table 5.2 shows an example request context for passing a sticky policy in the PERMIS policy language to the authorisation service along with the authorisation request. The resource attribute "rid" defines the RID of the resource to be access controlled (as defined in Section 3.3.2.5). The value "SUBMIT" for action attribute "action-id" indicates that this request is for submitting some new personal data (identified by the RID) along with a sticky policy to the authorisation service. The <Extensions> element contains the sticky policy as a <StickyPolicy> element. In the <StickyPolicy> element the PolicyID attribute is the PID (as defined in Section 3.3.2.5) and the PolicyLanguage attribute defines the policy language to use whereas the

<PolicyContents> contains the actual policy written in the specified language.

Table 5.2: An example request context for passing a PERMIS policy to the AIPEP

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<XACMLAuthzDecisionQuery
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
ID="A2010-12-13T12.58.12"
Version="2.0"
IssueInstant="2010-12-13T12:58:12.209Z">
<xacml-context:Request xmlns:xacml-context="X">
<xacml-context:Subject xmlns="X">
</xacml-context:Subject>
<xacml-context:Resource xmlns="X">
<xacml-context:Attribute AttributeId="rid"
DataType="Y#string">
<xacml-context:AttributeValue>LC01</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action xmlns="X">
<xacml-context:Attribute AttributeId="action-id"
DataType="Y#string">
<xacml-context:AttributeValue>SUBMIT</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="X"/>
</xacml-context:Request>
<Extensions><sp:StickyPolicy
PolicyID="PID123"
PolicyLanguage="PERMIS"
PolicyType="Authorization"
TimeOfCreation="2010-08-09T00:00:00Z"
xmlns:sp="http://sec.cs.kent.ac.uk/stickypolicy">
<sp:PolicyAuthor>
<sp:AuthorType>DataSubject</sp:AuthorType>
</sp:PolicyAuthor>
<sp:PolicyResourceTypes>
<sp:ResourceType>personal:preferences</sp:ResourceType>
</sp:PolicyResourceTypes>
<sp:PolicyContents>
<SubjectPolicy>
<SubjectDomainSpec ID="everywhere">
<Include LDAPDN=""/>
</SubjectDomainSpec>
</SubjectPolicy>
<RoleHierarchyPolicy>
<RoleSpec Type="permisRole" OID="1.2.826.0.1.3344810.1.1.14">
<SupRole Value="UNSPECIFIED"/>
</RoleSpec>
</RoleHierarchyPolicy>
<SOAPolicy>
<SOASpec ID="anyone" LDAPDN=""/>
</SOAPolicy>
<RoleAssignmentPolicy>
<RoleAssignment>
<SubjectDomain ID="everywhere"/>
<RoleList>
<Role Type="permisRole"/>
</RoleList>
```

```
<Delegate Depth="0"/>
<SOA ID="anyone"/>
<Validity/>
</RoleAssignment>
</RoleAssignmentPolicy>
<TargetPolicy>
<TargetDomainSpec ID="UK">
<Include LDAPDN="c=gb"/>
</TargetDomainSpec>
</TargetPolicy>
<ActionPolicy>
<Action Name="TRANSFER" ID="TRANSFER"/>
</ActionPolicy>
<TargetAccessPolicy>
<TargetAccess>
<RoleList/>
<TargetList>
<Target>
<TargetDomain ID="UK"/>
<AllowedAction ID="TRANSFER"/>
</Target>
</TargetList>
<Obligations>
<Obligation ObligationId=”http://sec.cs.kent.ac.uk/obligations/AttachStickyPolicy” FulfillOn="Permit"/>
</Obligations>
</TargetAccess>
</TargetAccessPolicy>
</X.509_PMI_RBAC_Policy>
</sp:PolicyContents>
</sp:StickyPolicy>
</Extensions>
</XACMLAuthzDecisionQuery>
</soapenv:Body>
</soapenv:Envelope >
X= urn:oasis:names:tc:xacml:2.0:context:schema:os
Y= http://www.w3.org/2001/XMLSchema
```

In order to transfer the data and policy together between the user's client and the application, or application to application, we defined the StickyPAD schema, which is presented in Appendix 9. For the implementation of the sticky transfer mechanism (Chadwick and Lievens 2008) we used the Application Protocol Enhancement Model, whereby the PEP augments the existing application protocol to make it capable of sending data with policies. The authoritative source combines the data/resources and policies together and signs the combined elements. The digital signature could be the XML signature defined in the StickyPAD or an externally provided one e.g. by using SSL/TLS to transfer it.

The protocol used for communicating between the AIPEP and the CVS is based on WS-TRUST (OASIS 2007) and SAMLv2 (SAMLv2.0 2010) and is documented in the OGF specification (Chadwick and Su 2009). Once the CVS has finished validating the subject's credentials, these are returned to the AIPEP as XACML formatted attributes and passed to the Master PDP.

We needed to choose a standard API between the AIPEP and the Master PDP. Since policies are being passed with the SAML-XACML protocol and the AIPEP is storing the policy in the policy store, then we do not need to consider passing the policy via the API. There are a

number of PDP APIs available, for example, the C authorisation API (Authorization C API Developer Reference 2003), the XACML Java API (Sun's XACML Implementation 2006) and the PERMIS Java API (Otenko, Chadwick and Thornton 2002). We chose to use the XAMCL API as it is only API that has come from a recognized standards organisation, OASIS.

Due to the deadline of the EU TAS3 project (an IST FP7 funded Integrated Project) which funded this research, the complex design needed to be implemented in a short period of time and that required help from other project team members. The majority of the implementation of the authorization service was done by Dr. Stijn Lievens and can be found on (PERMIS Standalone Authorisation Server 2011). The authorisation system is implemented as a web service running in a servlet container (Apache Tomcat). The portions of the system that are implemented by the candidate, namely the PEP and Contract Validation Service, are described in the next sections. Here a brief description is presented for the configuration of the authorisation service.

5.2.1.1 Configuration of the authorisation service

The administrator of an organisation willing to use the authorisation service needs to download it from our open source repository (PERMIS Standalone Authorisation Server 2011). After installing the authorisation service the administrator needs to configure it. The administrator has full control over 1) the database configuration 2) the obligations service configuration and 3) the Master PDP configuration.

In the database configuration the administrator defines where to store the sticky policies. It can be a file system or a relational database. To use a file system backed storage the following two attributes have to be used: “directory” and “filePath”. The directory attribute should give an existing (writable) folder on the file system which is the **policy store** defined in 3.3.2.5. New sticky policies will be stored inside this folder with a file name that is a (SHA-1) hash of the sticky policy’s identifier. The filePath attribute should give a file that contains the mapping between resource identifiers and sticky policies, which is the **sticky store** defined in 3.3.2.5. When a relational database is used the database configuration simply points to a configuration file containing the necessary information to connect to the database.

The MasterPDPConfiguration element allows configuring the actual PDPs that will be used by the AIPEP. The administrator needs to specify the locations of the Legal and controller’s access control policies and the conflict resolution policy, along with the PDPs to use them (e.g. PERMIS PDP or XACML PDP). When the LegalPDP or ControllerPDP element is missing, this is interpreted as having a PDP that always returns NotApplicable.

The optional ObligationsServiceConfiguration element allows the administrator to specify their own obligations.

5.2.2 Implementation of the application dependent PEP

An application is implemented with PHP which provides a user interface at the front end and the PEP at the back end for calling the authorisation service using the right protocol on behalf of the user (or user’s application). This PHP application runs under an Apache web service and

calls the authorisation service using the SOAP protocol. The application performs the following functionalities –

i) When it is provided with an access request in SAML-XACML it passes the request via the PEP to the authorisation system which is running as a web service in another machine using the SOAP protocol and gets the response from that.

ii) When it receives a data item (along with the policy) to store, it assigns a locally unique RID (Resource Identifier) to identify the resource and passes an authorisation request to the AIPEP via the PEP with the policy. When it receives a Grant response it stores the data and the resource attributes (such as, resource type, validity time, identity attributes of the issuer and the data subject) in a MySQL table.

iii) When a request for accessing a data item identified by a specific RID is received it makes a MySQL query to retrieve the resource attributes attached to that resource. It formats each of the resource attributes to place them in the SAML-XACML request context. It then passes the request context to the authorisation system's AIPEP via the PEP with a SOAP call. After receiving the Grant (or Deny) response from the AIPEP, it either retrieves the Data (with the RID) and passes that to the requester or sends a Deny to the requester.

iv) When it receives an access request for any personal data item along with a signed digital contract it calls the ConVS to validate the contract. When it receives a response from the ConVS it formats the received data to form attributes of the SAML-XACML request context and passes the request context to the AIPEP via the PEP. The application/ the PEP also does necessary changes to the attributes' names to match with the attributes presented in the policy. For example, in the policy the Resource-Type attribute obtained from the contract is named as ContractResourceType to avoid confusion with the ResourceType obtained from the stored resource attributes of the requested resource. Therefore, the ResourceType attribute obtained from the contract is renamed as ContractResourceType. Similarly, the E-mail address of the first signer of contract obtained from the ConVS is renamed by as <ContractPartyOne'sE-mail> and the E-mail address of the second signer of contract is renamed as <ContractPartyTwo'sE-mail>.

v) Since the application is also intended to provide an interface to receive policies (in the form of preferences) from the user, based on a scenario of registering a person for a medical service at a medical service provider, a set of XACML policies is prepared. When a user registers at a medical service provider (e.g. Kent Health Centre) the person is provided with an electronic form, mainly with tick boxes, for giving his/her choices about who can access his/her medical record. Each of these tick boxes is converted into machine executable XACML policies as shown next.

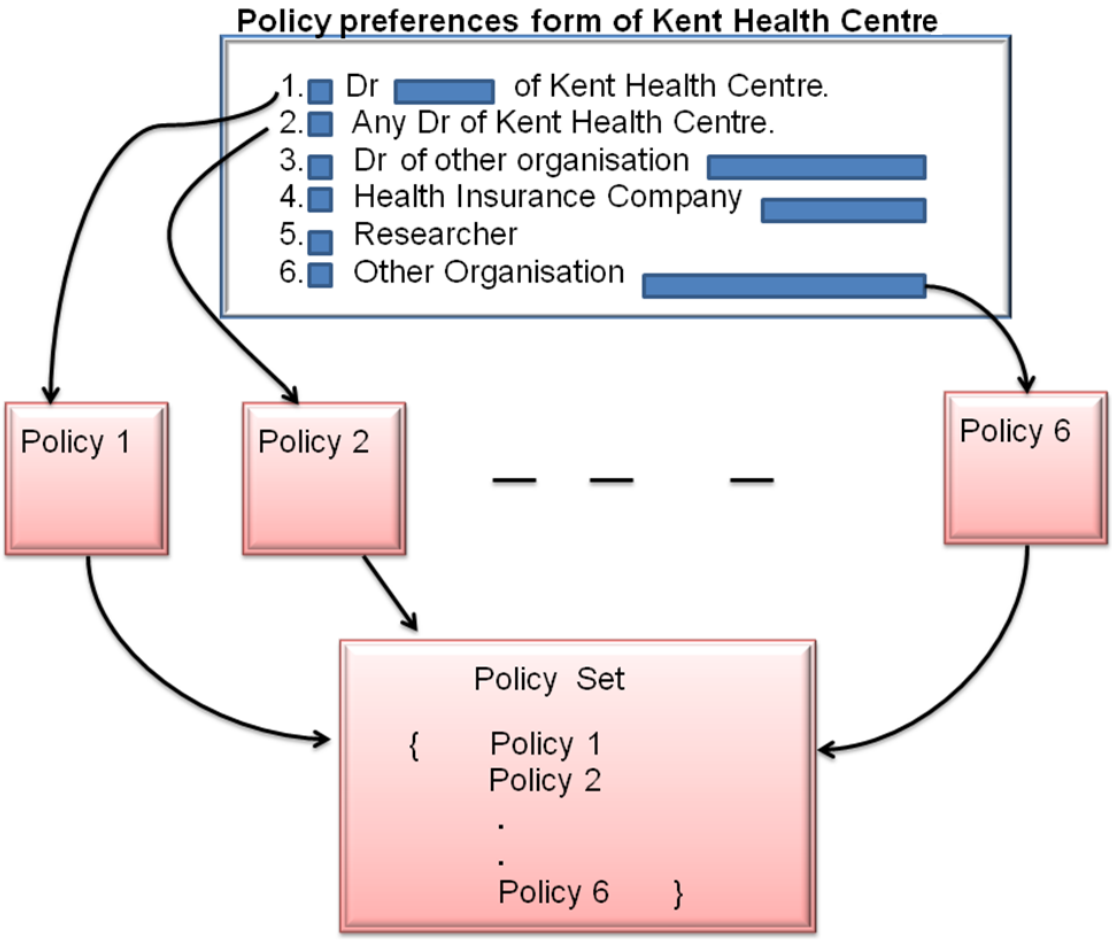

Figure 5-1. Conversion of choices (from tick boxes) into an XACML policy set

The user registers with the Kent Health Centre for getting services from it. The user first authenticates him/herself to the Kent Health Centre and then is given a registration form. During registration, the person is also given an electronic policy preference form (with tick boxes) which presents some options to choose about allowing others to access his/her medical data. These choices are converted into machine executable policies using the XACML policy language. In this example scenario the person is given 6 options (see Figure 5-1) to choose from to indicate with who he/she wants to share his/her medical data. The options are –

1. Dr. _ _ _ of Kent Health Centre
2. Any Dr. of Kent Health Centre
3. Dr. of other organisation _ _ _ .
4. Health Insurance Company _ _ _.
5. Researcher

6. Other Organisation _ _ _.

Some of these options contain spaces for providing extra information such as the name of a Dr., the name of another organisation and so on.

There is a pre-defined policy for each of these choices. Depending on the person's choices the policies are combined into a policy set and used in the data subject's PDP. For example if a person only chooses policies no 1 and 3, then only 1 and 3 will be put into the final policy set and be used in the data subject's PDP. If the person chooses option no 1 and writes the name of Dr. D in the space provided for that option then the XACML policy is formed as the policy presented in Table 5.3. This grants READ access to ResourceType =medical data to a subject with Name=Dr. D, working for Organisation Kent Health Centre with a Role of Medical Professional.

Table 5.3: Policy for option 1

```
<Policy PolicyId="PolicyNo1forMedicalData"
 RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Target/>
  <Rule RuleId="MedicalDataAccessByMedicalProfessional" Effect="Permit">
   <Description>Medical Professional of this organisation can read the medical data </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">Medical Professional</AttributeValue>
             <SubjectAttributeDesignator AttributeId=Role 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">Kent Health Centre</AttributeValue>
             <SubjectAttributeDesignator AttributeId="Organisation"
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">Dr. D</AttributeValue>
             <SubjectAttributeDesignator AttributeId="Name" 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">Medical Data</AttributeValue>
             <ResourceAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
AttributeId="ResourceType"/>
      </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 DataType="http://www.w3.org/2001/XMLSchema#string"
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
            </ActionMatch>
    </Action>
   </Target>
  </Rule>
</Policy>
```

When the user chooses option 5 to allow a researcher to access his/her medical data, an obligation is added in the policy to anonymise the data. The user can be informed about this in the form. Option 5 creates the policy presented in Table 5.4.

Table 5.4: Policy for option 5

```
<Policy PolicyId="PolicyNo5forMedicalData"
 RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Target/>
  <Rule RuleId="MedicalDataAccessByReseracher" Effect="Permit">
    <Description>Researcher can read the medical data </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">Reaseracher</AttributeValue>
               <SubjectAttributeDesignator AttributeId=Role 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">Medical Data</AttributeValue>
              <ResourceAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
AttributeId="ResourceType"/>
            </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 DataType="http://www.w3.org/2001/XMLSchema#string"
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
               </ActionMatch>
            </Action>
    </Target>
  </Rule>
  <Obligations>
    <Obligation ObligationId="Anonymise the requested data" FulfillOn="Permit"/>
  </Obligations>
</Policy>
```

When these 2 policies are put into one policy set it creates the full policy in XACML as presented in Table 5.5. The default policy combining rule is DenyOverrides. It can also be obtained from the data subject by providing him/her with a set of policy combining algorithms

(such as FirstApplicable, DenyOverrides, PermitOverride and so on) to choose from and the chosen combining algorithm can be used to combine the chosen policies.

Table 5.5: Combined policy from option 1 and 5

```
<PolicySet xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os /location-to-schema/access_control-xacml-2.0-policy-schema-os.xsd" PolicySetId="DataSubject's Policy" PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<Target xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os">
            </Target>
   <Policy PolicyId="PolicyNo1forMedicalData"
    RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
   ..........
  </Policy>
  <Policy PolicyId="PolicyNo5forMedicalData"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
   ..........
  </Policy>
</PolicySet>
```

User friendly interfaces and comprehensible texts can be provided with the options, so that the user can understand the options he/she is choosing without worrying about the complex policy languages underneath.

5.2.3 Implementation of the ConVS

For designing a ***signed contract*** that would allow some parties to access some data, we first have looked at the possible options of constructing the ***contract document*** and the way of accommodating signatures of the parties. The ***contract document*** that describes the terms and conditions of data access can be a plain document describing that. However to make the document easily interpretable by both machine and human a good option is to use the XML format. It allows to present data is a structured way and it is a W3C recommendation for describing data. The rationale of choosing various elements of the ***contract document*** is presented in Section 3.3.3.2. The schema of the ***contract document*** is presented in Table 5.6 and an example of that is presented in Table 5.7.

Table 5.6: Schema definition of the contract document

```
------ schema of the contract document that is referred by the URL of a contractID---
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="ContractDocument" type="ContractDocumentType"/>
  <xs:complexType name="ContractDocumentType">
        <xs:sequence minOccurs="1" maxOccurs="unbounded">
         <xs:element name="ValidityTime" type="ValidityTimeType"/>
         <xs:element name="ResourceType" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>

         <xs:element name="Description" type="xs:string"/>
         <xs:element name="SubjectOfContract" type="IdentifyingAttributesType" minOccurs="0"
maxOccurs="unbounded" />
         <xs:element name="AuthorisedRequester" type="IdentifyingAttributesType"
minOccurs="0"/>
         <xs:element name="PartyOfContract" type="PartyOfContractType" minOccurs="2"
maxOccurs="unbounded" />
         <xs:element name="ExtensionElement" type="AnyXMLType" minOccurs="0"/>
```

```
        </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ValidityTimeType">
        <xs:sequence>
         <xs:element name="StartDate" type="xs:date" />
         <xs:element name="EndDate" type="xs:date" />
   </xs:sequence>
  </xs:complexType>
  <xs:complexType name="IdentifyingAttributeType">
        <xs:choice minOccurs="0" maxOccurs="unbounded">
         <xs:element name="E-mail" type="xs:string"/>
         <xs:element name="NameAndAddress" type="NameAndAddressType"/>
         <xs:element name="NHSNumber" type="xs:string"/>
    <xs:element name="RoleAndOrganisation" type="RoleAndOrganisationType"/>
    <xs:element name="ExtensionElement" type="AnyXMLType" minOccurs="0"/>
        </xs:choice>
  </xs:complexType>
 <xs:complexType name="PartyOfContractType">
        <xs:sequence>
          <xs:element name="SignerOfContract" type="SignerOfContractType" minOccurs="1"/>
          <xs:element name="IdentifyingAttributesOfParty" type=" IdentifyingAttributesType "
minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
 </xs:complexType>

 <xs:complexType name="SignerOfContractType">
   <xs:sequence>
         <xs:element name="DN" type="xs:string" minOccurs="1" maxOccurs="1"/>
         <xs:element name="OtherIdentifyingAttributesOfSigner" type="IdentifyingAttributesType"
minOccurs="0" maxOccurs="unbounded" />
   </xs:sequence>
 </xs:complexType>

 <xs:complexType name="NameAndAddressType">
        <xs:sequence>
         <xs:element name="name" type="xs:string"/>
         <xs:element name="address" type="xs:string"/>
        </xs:sequence>
 </xs:complexType>

 <xs:complexType name="RoleAndOrganisationType">
        <xs:sequence>
          <xs:element name="role" type="xs:string"/>
          <xs:element name="organisation" type="xs:string"/>
        </xs:sequence>
 </xs:complexType>
        <xs:complexType name="AnyXMLType" mixed="true">
     <xs:sequence>
      <xs:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="lax">
        <xs:annotation>
          <xs:documentation>
          Any xml content is allowed in this element.
          </xs:documentation>
        </xs:annotation>
      </xs:any>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Table 5.7: Example of contract document

```
--- Example of Contract ---
<?xml version="1.0" encoding="UTF-8"?>
<ContractDocument xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation ="contractDocumentXMLSchema.xsd">
  <ValidityTime>
                <StartDate>2009-09-24</StartDate>
                <EndDate>2012-10-23</EndDate>
  </ValidityTime>
  <ResourceType>MedicalData</ResourceType>
  <Description>
                This is a contract between CN=kaniz,O=UKC,C=GB and O=HIC,C=GB for a service.
  </Description>
  <SubjectOfContract>
                <E-mail>kf66@kent.ac.uk</E-mail>
  </SubjectOfContract>
  <AuthorisedRequester>
                <RoleAndOrganisation>
        <role>Employee</role>
        <organisation>HIC</organisation>
    </RoleAndOrganisation>
  </AuthorisedRequester>
  <PartyOfContract>
        <SignerOfContract>CN=A,O=HIC,C=GB</SignerOfContract>
        <OtherIdentifyingAttributesOfSigner>
      <RoleAndOrganisation>
        <role>Authority</role>
        <organisation>HIC</organisation>
      </RoleAndOrganisation>
    </OtherIdentifyingAttributesOfSigner>
    <IdentifyingAttributesOfParty>
         <name>HIC</name>
         <address>Kent,UK</address>
    </IdentifyingAttributesOfParty>
 </PartyOfContract>
 <PartyOfContract>
         <SignerOfContract>
      <DN>CN=Kaniz,O=UKC,C=GB</DN>
           <OtherIdentifyingAttributesOfSigner>
        <E-mail>kf66@kent.ac.uk</E-mail>
      </OtherIdentifyingAttributesOfSigner>
    </SignerOfContract>
 </PartyOfContract>
</ContractDocument>
```

We then have looked into various options for signing an XML document with a digital signature which are presented next.

The syntax and semantics of the XML signature can be found in the W3C standard document (W3CXMLSignature 2008). A brief description of the formats of the XML signature is given here to justify the reason for choosing this format.

i) **Enveloped signature**- in which the resultant signed content contains the signature as an element.

ii) Enveloping signature- in which the signed content is embedded within the XML signature in the resultant signed document i.e. the signature is the parent element.

iii) **Detached signature-** in this format the XML content is external to the Signature

element, and can be identified via a URL. Consequently, the signature is "detached" from the content it signs.

Since the implementation of a digital contract requires that the same content (the XML formatted ***contract document***) be signed by a number of parties the detached signature seems to be more suitable than the other two formats. Hence, the XML formatted ***contract document*** is presented via a URL and all the parties sign over the contents of the ***contract document*** with their own keys. The URL and the digest of the ***contract document*** become a part of the digital signatures and are used while validating the contract. The URL of the ***contract document*** and the signatures by all the parties over the contract document together form the ***signed contract*** and is used as a unique identifier of the contract. The schema of the ***signed contract*** is given in Table 5.8. All the parties have access to the signed contract which they can use for validation while accessing data based on that contract.

The element ***Signature*** of the signed contract schema is defined by the W3C. A brief description of the signature element is provided to aid understanding of the validation process.
The ***SignedInfo*** element includes the canonicalisation[6] algorithm, a signature algorithm, and one or more references. The ***Reference*** element (not shown in Table 5.8, can be seen in (W3CXMLSignature 2008)) contains the URL of the resource, the list of processing steps performed on the reference's content before it was digested (in ***Transforms*** element of ***Reference*** element), the digest method and the value of the digest of the referenced resource.

The ***SignatureValue*** element contains the actual value of the digital signature (i.e. the encrypted digest) in base64 encoded form.

The optional ***KeyInfo*** element contains keys, names, certificates and other public key management information, such as in-band key distribution or key agreement data. The optional element ***Object*** may contain MIME type, ID, encoding attributes which attribute allows an element to be referenced by other objects.

The XML formatted digital contract resides inside an Apache web server. The URI of the contract is used for signing by the parties of the contract. For generating a digest of the document SHA1 is used and for the signature the RSA algorithm is used. The signatures of each party of the contract are passed to the ConVS by the application or the PEP (which is implemented in PHP). If the signatures are valid and the contract is valid (i.e., if the contract passes the four validity steps mentioned in Section 3.3.3), a string containing the information from the detached signatures and the contract document in a comma separated value (CSV) format is passed to the calling element. After receiving the information the application / PEP parses that, constructs and adds the desired XACML elements to the request context to pass them to the authorisation service.

The ConVS is implemented as a Web Service (although it could also be used as an API if desired) using both Java SE and J2EE. It relies on the Axis2 (AXIS2 2004) framework to provide

[6] It is the process of making something canonical, which means in conformance with some specification.

the web services functionality. The ConVS runs inside a servlet container (Apache Tomcat). Figure 5.2 presents the Snapshot of the ConVS web service. The web service has only one public class which is called by the application or the PEP and is passed the detached signatures of the contract. This public class calls all the other private classes for providing the four functionalities mentioned in Section 3.3.3.

Table 5.8: Schema definition of the signed contract

```
-- signed contract schema –
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="DigSignatureSchema.xsd"/>

<xs:element name="SignedContract" type="SignedContractType" />
  <xs:complexType name="SignedContractType">
   <xs:sequence>
     <xs:element name="ContractIdentifier" type="xs:httpURL" />
     <xs:element name="Signature" type="ds:SignatureType" minOccurs="2"
maxOccurs="unbounded">
     </xs:element>
   </xs:sequence>
  </xs:complexType>
</xs:schema>
--digital signature schema definition from http://www.w3.org/TR/xmldsig-core/--
        <complexType name="SignatureType">
          <sequence>
                 <element ref="ds:SignedInfo"/>
                 <element ref="ds:SignatureValue"/>
                 <element ref="ds:KeyInfo" minOccurs="0"/>
                 <element ref="ds:Object" minOccurs="0" maxOccurs="unbounded"/>
          </sequence>
          <attribute name="Id" type="ID" use="optional"/>
   </complexType>
   <element name="SignedInfo" type="ds:SignedInfoType"/>
   <complexType name="SignedInfoType">
           <sequence>
                 <element ref="ds:CanonicalizationMethod"/>
         <element ref="ds:SignatureMethod"/>
         <element ref="ds:Reference" maxOccurs="unbounded"/>
          </sequence>
          <attribute name="Id" type="ID" use="optional"/>
  </complexType>
```

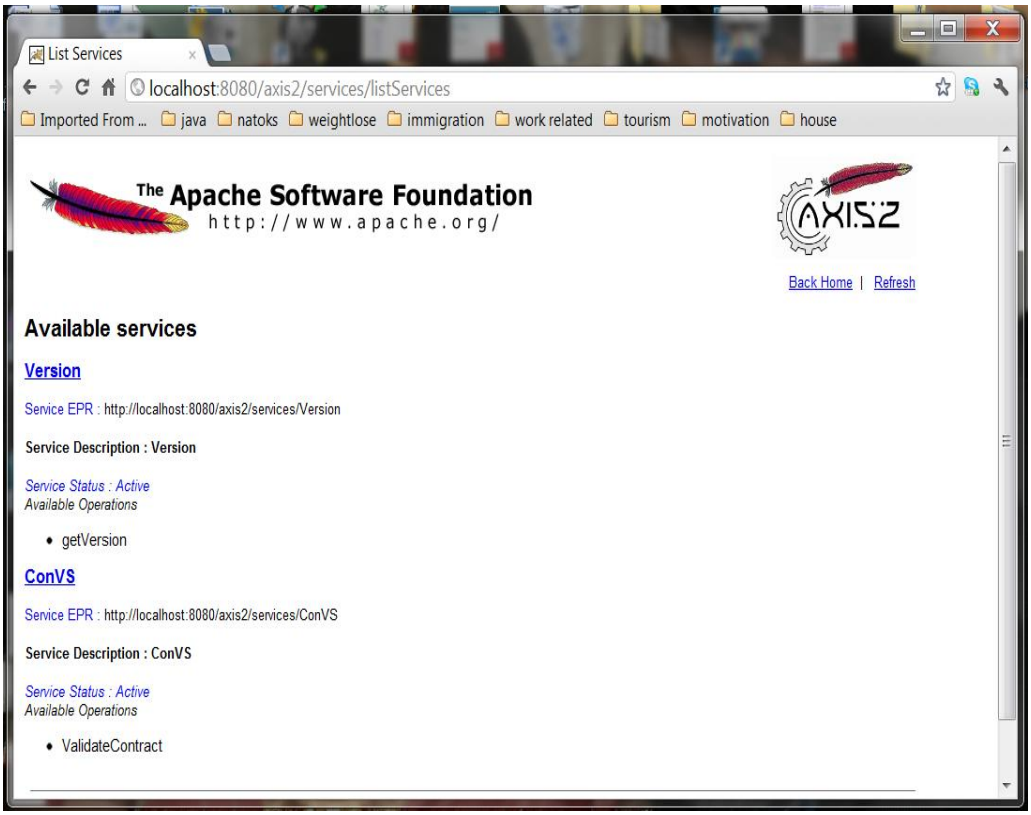

Figure 5-2. Snapshot of the ConVS web service

5.3 Conversion of CNL Rules to PDP Rules

The Legal access control and conflict resolution policies are implemented using both the XACMLv2 and PERMIS policy languages. We chose XACML as a language for implementing Legal policies as it is an OASIS standard and is regarded as the most popular policy language. Since our system is capable of dealing with more than one language we wanted to demonstrate such capability by encoding the Legal rules in another policy language as well and enforcing that in our system. Therefore, we chose PERMIS for this purpose due to the availability of the code (being open source) and expertise within the project. Here a brief description of the procedure is provided. Both the XACML and PERMIS rules are obtained manually following the procedure. A program is written to automatically convert the rules in XACML and compare with the corresponding manually obtained XACML rules to demonstrate that it is possible to have automatic extraction of machine executable rules from the CNL rules.

5.3.1 Conversion of CNL rules to XACMLv2 rules

The basic structure XACMLv2 policy elements and the evaluation process of that are presented in Section 2.2.9. Each of the <PolicySet>, <Policy> or <Rule> element contains a <Target> element that specifies the set of subject, resource, action and environment attributes to which to apply. A <Target> value evaluates to "Match" if the all the <Subjects>, <Resources>, <Actions> and <Environments> elements specified in the <Target> match the values in the request context. If any of the <Subjects>, <Resources>, <Actions> and <Environments> elements specified in the <Target> evaluates to "Indeterminate", then the <Target> and

eventually the effects of the <Rule>, <Policy> and <PolicySet> containing the target evaluates to "Indeterminate". The absence of an attribute in the request context that is found in the <Policy> is evaluated to "Indeterminate".

Let us consider the implementation of a Legal rule saying that the data subject can submit policy for his/her personal data, where the data subject can be identified by either an {e-mailAddress}, or {{Name} and {Address}}, or a {NHS Number}. The requester is expected to provide any one of these sets of attributes to identify him or herself. The data subject's identity attributes are passed by the PEP as resource attributes in the request context. The attributes passed by the requester should be the same as a subset of these subject's identity attributes in order for the requester to be granted access. If these identity attributes are written inside the target element of the rule then they all are needed to be present in the request context for the requester to be granted access. If any of the attributes are missing from the request context then an "Indeterminate" decision is returned. However, the requester only needs to be identified by one of these sets of attributes at a time and the other sets of attributes need not be present in the request context. Therefore a single target element is not the correct place to do the matching. Instead either three separate policies or rules with one set of identity attributes in each target element should be used, or one policy or rule with an empty target element and a condition element need to be used.

The <Condition> element represents a Boolean expression. When the <Target> element of an XACML policy evaluates to "Match" and the condition element evaluates to "True", the rule evaluates to "Effect" and when the <Target> element of an XACML policy evaluates to "Match" and the condition element evaluates to "False" the rule evaluates to "NotApplicable" (XACMLv2 2005). XACML has a collection of ***functions*** that provide a powerful way to compare attribute values. The XACML function *urn:oasis:names:tc:xacml:1.0:function:type-at-least-one-member-of* takes two arguments that are both a ***bag*** of '*type*' values. It returns a Boolean value. The function evaluates to "True" if and only if at least one element of the first argument is contained in the second argument as determined by "urn:oasis:names:tc:xacml:x.x:function:type-is-in". In this case the <Condition> element containing the function evaluates to "True" and eventually the rule evaluates to the "effect". Otherwise the function returns "False" and the <Condition> evaluates to "False" and consequently the rule evaluates to "NotApplicable". Hence this strategy is used to match the subject attributes and the above rule is implemented as mentioned in Table 5.9.

Table 5.9: Implementation of a Legal rule in XACML v2

```
<Rule RuleId="3" Effect="Permit">
<Description> ACR 3: If the Subject:Email:string is equal to the resource:DataSubject'sE-mail:string OR the Subject:NHSNumber:string is equal to the Resource:DataSubject'sNHSNumber:string then Grant the SubmitPolicy for PersonalData. </Description>
<Target>
</Target>
 <Condition>
  <Apply FunctionId="X:and">
    <Apply FunctionId="X:or">
            <Apply FunctionId="X:string-at-least-one-member-of">
              <SubjectAttributeDesignator AttributeId="E-mail" DataType="Y#string" />
              <ResourceAttributeDesignator AttributeId="DataSubject'sE-mail" DataType="Y#string" />
```

```
            </Apply>
            <Apply FunctionId="X:string-at-least-one-member-of">
              <SubjectAttributeDesignator AttributeId="NHSNumber" DataType="Y#string" />
              <ResourceAttributeDesignator AttributeId="DataSubject'sNHSNumber" DataType="X#string" />
            </Apply>
            <Apply FunctionId="X:any-of">
        <Function FunctionId="X:string-equal"/>
        <ActionAttributeDesignator DataType="Y#string"
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
        <AttributeValue DataType="Y#string">SubmitPolicy</AttributeValue>
    </Apply>
        <Apply FunctionId="X:any-of">
        <Function FunctionId="X:string-equal"/>
        <ActionAttributeDesignator DataType="Y#string"
AttributeId="ResourceType"/>
        <AttributeValue DataType="Y#string">PersonalData</AttributeValue>
    </Apply>
  </Apply>
 </Condition>
</Rule>
```

Here X= urn:oasis:names:tc:xacml:1.0:function and Y= http://www.w3.org/2001/XMLSchema

The CNL rules written according to the ABNF grammar (presented in Section 4.4) are converted into XACML using the following steps-

1. The initial string of a rule "ACR" indicates that the rule is an access control rule and the initial string "CRR" indicates that the rule is a CRR.
2. The ABNF grammar element <rule-id> becomes the <PolicyId> and <Rule-Id> of the XACML policy.
3. The ABNF grammar element <GrantOrDeny> becomes the <effect> of XACML rule.
4. Depending on the value of <GrantOrDeny> the rule combining algorithm is chosen for XACML. If the value of <GrantOrDeny> is Grant then the rule combining algorithm is *permit-overrides* and if the value is Deny or BreakTheGlass then rule combining algorithm is *deny-overrides.* Note: the policy combining algorithm for Legal PDP is *first-applicable*.
5. The whole rule is copied to the <Description> element of the XACML rule.
6. All the <relationaloperator> of the ABNF grammar are translated into corresponding XACML functions.
7. Each <attribute> of a <condition> element of the ABNF grammar consists of <category>, <name> and <type>. The value of <category> becomes the prefix (i.e. Subject /Resource /Action /Environment) of the AttributeDesignator of XACML, the value of <name> becomes the AttributeId and the <type> becomes the <DataType> of the AttributeDesignator of XACML.
8. The attribute <value> of ABNF becomes the <AttributeValue> of XACML.
9. The ABNF <operator> "AND" / "OR" becomes the XACML FunctionId = "and" / "or" [7] .
   [Note: the XACML FunctionId= "not" comes from the ABNF <relationaloperator> "is not" and "is not equal to".]
10. The <action> element of the ABNF grammar becomes the <AttributeValue> in

XACML against which the <ActionAttributeDesignator> with AttributeId= action-id[8] is matched.

11. The <obligations> of the ABNF element becomes the XACML <Obligations>.

5.3.2 Conversion of CNL rules to PERMIS rules

The basic construction of PERMIS rules is explained in Section 2.2.10. Here we consider the implementation of the same example of the rule mentioned in the Section 5.3.1 in PERMIS The data subject can submit a policy for his/her personal data, where the data subject can be identified by either an {e-mailAddress}, or a {NHS Number}. In PERMIS policies there is no way to define arbitrary subject's or resource's attributes. They are tested to be either present or not. So the requester's identity attributes are passed as environment attributes, as are the resource's subject identity attributes as presented in Table 5.10.

Table 5.10: Implementation of a Legal rule in PERMIS

```
<TargetAccess>
 <RoleList>
 </RoleList>
 <TargetList>
  <Target>
   <TargetDomain ID="PersonalData"/>
   <AllowedAction ID="SubmitPolicy"/>
  </Target>
 </TargetList>
 <IF>
  <OR>
        <EQ>
    <Environment Parameter="E-mail" Type="String" />
          <Environment Parameter="DataSubject'sE-mail" Type="String" />
        </EQ>
        <EQ>
          <Environment Parameter="NHSNumber" Type="String" />
          <Environment Parameter="DataSubject'sNHSNumber" Type="String" />
        </EQ>
        </OR>
 </IF>
</TargetAccess>
```

Similar to XACML the CNL rules written according to the ABNF grammar (presented in Section 4.4) are converted into PERMIS using the following steps-

a) The initial of a rule "ACR" indicates that the rule is an access control rule and the initial "CRR" indicates that the rule is a CRR.
b) The ABNF grammar element <rule-id> becomes the *OID* of the PERMIS policy.
c) If the value of <GrantOrDeny> is Grant then the value of *DenyBased* attribute of PERMIS policy is "false", otherwise it is "true".
d) All the <relationaloperator> of the ABNF grammar are translated into the corresponding PERMIS functions <EQ> (equal to), <GT> (greater than), <LT> (less than), <NOT> (not).
e) Each <attribute> of a <condition> element of the ABNF grammar consists of <category>, <name> and <type>. Since no arbitrary subject/ resource attributes can be

[7] The prefix "urn:oasis:names:tc:xacml:1.0:function:" is omitted for readability.
[8] The prefix "urn:oasis:names:tc:xacml:1.0:function:" is omitted for readability.

defined in PERMIS all the attributes are presented as Environment attributes. The value of <name> becomes the value of *Parameter* and the <type> becomes the *Type* of the Environment attribute of PERMIS.

f) The attribute <value> of the ABNF becomes the Constant Value against which the environment attributes are compared according to the PERMIS function.
g) The ABNF <operator> of "AND" / "OR" becomes the <AND> / <OR> function of PERMIS.
h) The value of the <action> element of the ABNF grammar becomes the *Action Name* and *ID* of the PERMIS policy.
i) The <obligations> of the ABNF element becomes the PERMIS <Obligations>.

5.3.3 Automated conversion of CNL rules to XACML rules

A tool named XACMLConverter was created in Java to convert the CNL rules into XACML. The conversion process is performed in two stages as shown in Figure 5-3.

Stage1: Parsing the *input.txt* (containing the CNL rules) according to the ABNF grammar (passed by *grammar.txt*) produces an *intermediate.xml* file. *Intermediate.xml* is XML whose structure follows the ABNF grammar rules.

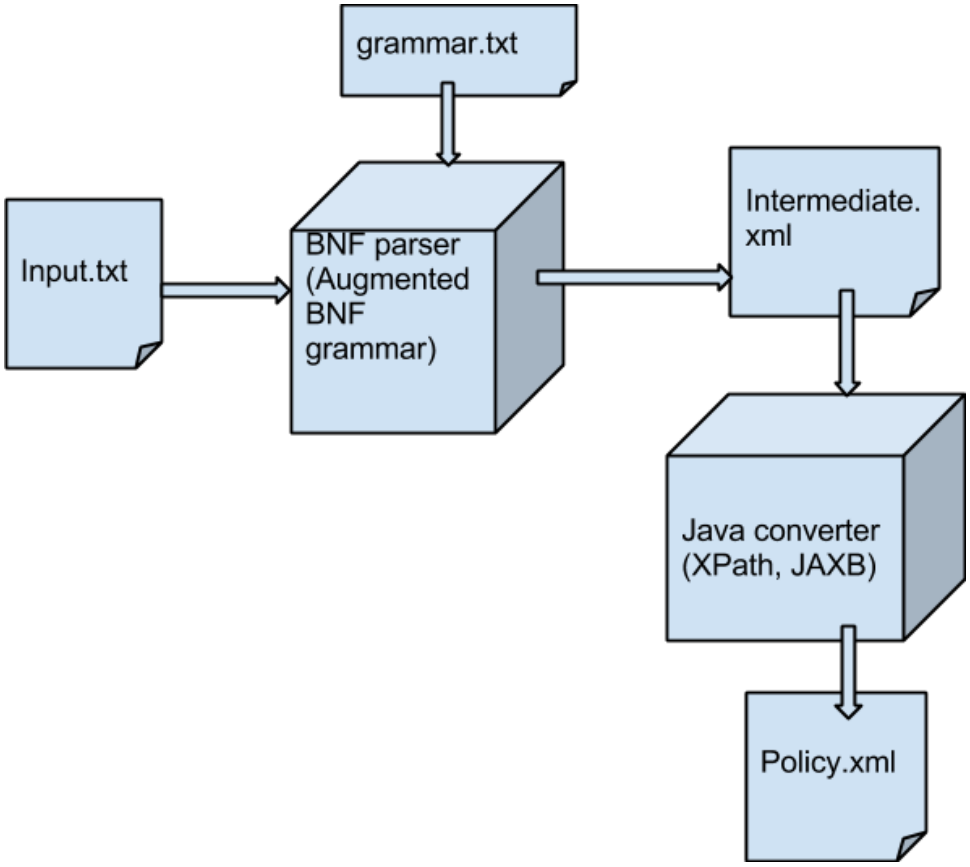

Figure 5-3. Automated conversion process from CNL rules to XACML rules.

Table 5.11 is showing the intermediate.txt[9] produced from this input.txt containing "*ACR 3: If the Environment:RequestTime:date is less than Resource:ValidityTime:date then Deny the Access to the PersonalData.*"

Table 5.11: Example intermediate.txt produced from the input.txt

<rule-definition>**ACR**<wp> </wp>
<rule-id><STRING>3</STRING></rule-id>
<wp></wp>**:**<wp> </wp>
<rule-statement>**If**<wp> </wp>
<conditions><condition><article><wp></wp>**the**<wp> </wp></article>
<attributes><attribute><category>**Environment**</category>**:**<name><STRING>**RequestTime**</STRING>
</name>**:**<type>**date**</type>
</attribute></attributes><relationalOperator>**is less than**</relationalOperator>
<attributes><attribute><category>**Resource**</category>:<name><STRING>**ValidityTime**</STRING>
</name>:<type>**date**</type></attribute></attributes></condition></conditions>
<wp> </wp>**then**<wp> </wp>
<GrantOrDeny>**Deny**</GrantOrDeny><article>**the**</article>
<actions><action><word>**Access**</word></action></actions>
< <prep>**to**</prep><article><wp></wp>**the**<wp> </wp></article>
<ResourceType><word>**PersonalData**</word></ResourceType</rule-statement>**.**</rule-definition>

For implementing stage 1 a free Java tool aParse (http://www.parse2.com) is used to convert the CNL rules into XACML.

Stage 2: Converting the intermediate.xml into policy.xml using XPath and JAVA Architecture for XML binding.

A Java object representation of XACML is created first. Java Architecture for XML binding (JABX) is used to convert the Java classes into XML. For example, to represent <Rule RuleId="3" Effect="Grant"> the following Java classes are created:

```
public class Rule {
protected String ruleId;
protected String effect;
//getters and setters methods will follow.
}
```

To tell the class, and how to marshall it into the desired XML format, Java annotations are used as specified in JAXB. Since we want both the ruleId and effect to be attributes of the <Rule> element, we simply 'annotated' them as @XMLAttribute

```
public class Rule {
@XmlAttribute(name = "RuleId")
protected String ruleId;
@XmlAttribute(name = "Effect")
protected String effect;
//getters and setters methods will follow}
```

After creating the *Rule* object the values for the attributes need to be provided. Intermediate

[9] Some <whitespace> (<wp>) elements have been removed from the produced output for readability.

XML contains all the needed values. We have used 'XPath' to fetch the interested values. For example, From intermediate.xml we read the *ruleId* by the following code: *String ruleId = inputHelper.getNodeValue("/rule-definition/rule-id/STRING/text()");*

"/rule-definition/rule-id/STRING/" defines the path in intermediate.xml where ruleId is specified. Table 5.12 is showing some example source code of stage 2 of the XACMLConverter.

Table 5.12: Example source code of stage2 of XACMLConverter

```
// Create a policy object, which will be converted to xml using JAXB
Policy policy = factory.createPolicy();
// get ruleId using XPath
String ruleId = inputHelper.getNodeValue("//rule-definition/rule-id/STRING/text()");
// Set ruleId to policy
Rule rule = factory.createRule();
rule.setRuleId(ruleId);
policy.setRule(rule);
// prepare marshaling to xml
JAXBContext jaxbContext = JAXBContext.newInstance(Policy.class);
Marshaller jaxbMarshaller = jaxbContext.createMarshaller();
// write to file
File file = new File(convertedFile);
jaxbMarshaller.marshal(policy, file);
```

5.3.4 Implementation of the conflict resolution rules

All the ACRs are also converted into CRRs. While converting the ACR into CRR the <effect> of XACML is always "permit" and the <Obligation> of XACML becomes "permit-overrides" or "deny-overrides" depending on the value of <GrantOrDeny> of the ABNF (see Section 4.5.6).

The conversion of a CRR presented according to the ABNF grammar of Section 4.4 to an XACML / PERMIS policy follows the steps similar to the conversion of an ACR. However, the new element DCR becomes an <obligation> in XACML.

In XACMLv2 the smallest element on which the <Obligation> can be applied is <Policy>. Hence in XACML the Conflict Resolution Rules are implemented as separate <Policy> elements inside a <PolicySet> element. The <Obligation> returns the DCA to use (as described in Chapter 3 and 4). In PERMIS each CRR is written as a separate Target Access Rules (TAR) with an obligation to use a DCA. As an example of conversion we use the same Legal rule mentioned in Section 5.3.1 (i.e. the data subject can submit a policy for his/her personal data, where the data subject can be identified by either an {E-mailAddress}, or a {NHS Number}) as a CRR to convert into XACML v2. This is shown in Table 5.13.

Table 5.13: Implementation of a CRR in XACML v2 and PERMIS

```
----------Implementation of a CRR in XACML v2----------

<Policy PolicyId="3" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Target></Target>
  <Rule RuleId="3" Effect="Permit">
```

```
    <Description> ACR 4: If the Subject:Email:string is equal to the resource:DataSubject'sE-mail:string OR the
Subject:NHSNumber:string is equal to the Resource:DataSubject'sNHSNumber:string then Grant the SubmitPolicy for
PersonalData. </Description>
     <Target/>
     <Condition>
      <Apply FunctionId="X:and">
              <Apply FunctionId="X:or">
          <Apply FunctionId="X:string-at-least-one-member-of">
                <SubjectAttributeDesignator AttributeId="E-mail" DataType="Y#string" />
                <ResourceAttributeDesignator AttributeId="DataSubject'sE-mail" DataType="Y#string" />
               </Apply>
               <Apply FunctionId="X:string-at-least-one-member-of">
                 <SubjectAttributeDesignator AttributeId="NHSNumber" DataType="Y#string" />
                 <ResourceAttributeDesignator AttributeId="DataSubject'sNHSNumber" DataType="X#string" />
              </Apply>
       </Apply>
       <Apply FunctionId="X:any-of">
         <Function FunctionId="X:string-equal"/>
           <ActionAttributeDesignator DataType="Y#string" AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
          <AttributeValue DataType="Y#string">SubmitPolicy</AttributeValue>
        </Apply>
        <Apply FunctionId="X:any-of">
          <Function FunctionId="X:string-equal"/>
            <ActionAttributeDesignator DataType="Y#string" AttributeId="ResourceType"/>
           <AttributeValue DataType="Y#string">PersonalData</AttributeValue>
          </Apply>
        </Apply>
     </Condition>
    </Rule>
    <Obligations>
             <Obligation ObligationId="http://sec.cs.kent.ac.uk/masterpdp/conflictresolution/permit-overrides"
FulfillOn="Permit"/>
    </Obligations>
</Policy>
Here X= urn:oasis:names:tc:xacml:1.0:function and Y= http://www.w3.org/2001/XMLSchema
----------Implementation of a CRR in PERMIS----------

<?xml version="1.0" encoding="UTF-8"?>
<X.509_PMI_RBAC_Policy OID="LawPolicyv1">
 <SubjectPolicy>
  <SubjectDomainSpec ID="everywhere">
   <Include LDAPDN=""/>
  </SubjectDomainSpec>
 </SubjectPolicy>
 <RoleHierarchyPolicy>
  <RoleSpec Type="permisRole" OID="1.2.826.0.1.3344810.1.1.14">
  </RoleSpec>
 </RoleHierarchyPolicy>
 <SOAPolicy>
  <SOASpec ID="anyone" LDAPDN=""/>
 </SOAPolicy>
 <RoleAssignmentPolicy>
  <RoleAssignment>
   <SubjectDomain ID="everywhere"/>
   <RoleList>
   </RoleList>
   <Delegate Depth="0"/>
   <SOA ID="anyone"/>
   <Validity/>
```

```
  </RoleAssignment>
 </RoleAssignmentPolicy>
 <TargetPolicy>
  <TargetDomainSpec ID="PersonalData">
   <Include URL=""/>
   <ObjectClass Name="PersonalData"/>
  </TargetDomainSpec>
 </TargetPolicy>
 <ActionPolicy>
  <Action Name="SubmitPolicy" ID="SubmitPolicy">
   <TargetDomain ID="PersonalData"/>
  </Action>
 </ActionPolicy>
 <TargetAccessPolicy>
    <!-- DataSubject can SubmitPolicy for his/her personal data.
   -->
  <TargetAccess>
   <RoleList>
   </RoleList>
   <TargetList>
    <Target>
     <TargetDomain ID="PersonalData"/>
     <AllowedAction ID="SubmitPolicy"/>
    </Target>
   </TargetList>
   <IF>
           <OR>
        <EQ>
                <Environment Parameter="E-mail" Type="String" />
                      <Environment Parameter="DataSubject'sE-mail" Type="String" />
              </EQ>
              <EQ>
                       <Environment Parameter="NHSNumber" Type="String" />
                       <Environment Parameter="DataSubject'sNHSNumber" Type="String" />
               </EQ>
             </OR>
           </IF>
           <Obligations>
                      <Obligation ObligationId="http://sec.cs.kent.ac.uk/masterpdp/conflictresolution/permit-overrides"
FulfillOn="Permit"/>
           </Obligations>
  </TargetAccess>
 </TargetAccessPolicy>
</X.509_PMI_RBAC_Policy>
```

5.4 Validation of the Legal Rules

The aim of the validation tests of the Legal policy rules is to evaluate whether the rules give the desired responses or not. Martin et al. (Martin, et al. 2008) proposed an approach for determining a set of properties with which policies can be verified. It creates mutant policies (e.g. by altering the effects of policies) and tests against the properties of original policies. If the mutant policy still holds the properties of original then the quality of the properties is not sufficient. The number of properties of the original policy that is not held by the mutant policy determines the quality of the properties for verifying a policy. Xiao et al. (Xiao, et al. 2012) used a validation process which evaluates the extracted access requests from NL functional

requirements against extracted policies from software documents which is helpful for finding name inconsistency (i.e. the same user is defined with different names in the policy). However this process is not rigorous enough to use for our validation purpose. Fisler et al. (Fisler, et al. 2005) used a decision-diagram based presentation of an access control policy for analysing it in software called Margrave[10] (Nelson 2010). This software provides functions to answer queries about properties of a policy. Although Margrave can handle the core XACML its current implementation suffers from some limitations, such as, it cannot support XACML functions other than string equality, and it cannot handle obligations. Hence this tool cannot be used for validation testing of our extracted rules. However our test cases used a similar approach of Fisler for analysing the rules. Our approach is to methodologically test each rule by forming a request context with the various combinations of the rule's attributes.

We constructed a set of test cases based on evaluating each rule one by one. Each Legal rule is a combination of conditions and each condition consists of either an attribute-attribute pair or attribute-value pair and their relationship. Each condition can evaluate to *true* or *false*. A condition is related to other conditions by a binary operator (AND, OR). Exhaustive test cases have been generated based on each condition where each condition in each rule had two test cases created for it, one where the condition was known to be true and one where the condition was known to be false. For generating the test cases a MTBDD (Multi-terminal binary decision diagram) can be generated where each condition becomes a node in the binary tree. Figure 5-4 shows the MTBDD of the test cases shown in Table 5.14.

[10] http://www.margrave-tool.org/

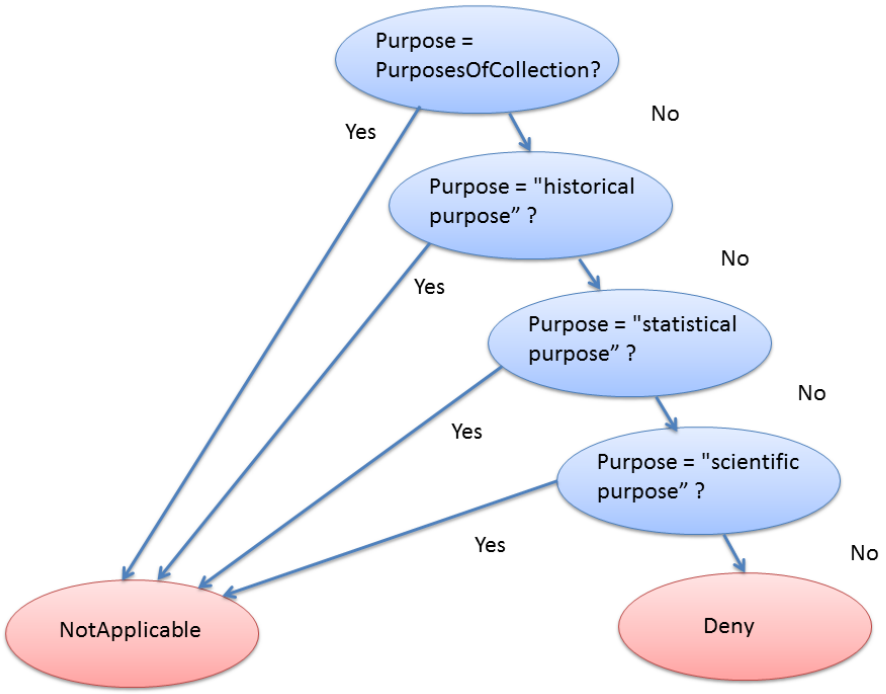

Figure 5-4. Presentation of rule in the form of MTBDD where the results form terminal nodes and each condition becomes a decision node.

A request context has been prepared for each text case. For each request context the responses of the Legal access control and conflict resolution PDPs are compared with the desired outcomes. For performing the validation tests the system is first configured with only the Legal Conflict Resolution PDP and the decisions are obtained like any normal PDP decisions. The DCA is obtained as a part of the returned obligation. Then the system is configured with only the Legal Access Control PDP to see the response. The same set of tests is conducted for it as well. Table 5.14 shows the responses of both the Legal Access Control PDP and the Legal Conflict Resolution PDP for one Legal rule. Similar test cases have been generated for all the Legal rules. More than 100 test cases were generated in total and they are presented in Appendix 1. The PHP PEP and the SoapUI[11] both are used for the validation tests.

Table 5.14: Validation tests of a Legal access control and conflict resolution policy

Test case No.	Request Context	Expected result	Obtained result	Comments
1.	Purpose of collection= [X] purpose (current purpose)= [X]	DCA = N/A Decision = N/A	DCA = N/A Decision = N/A	These tests are verifying the NL Legal rule "If the requested purpose of processing does not match with the original purpose of
2.	Purpose of collection = [X] purpose (current	DCA=DenyOverrides	DCA=DenyOverrides	

[11] http://www.soapui.org/

	purpose)= [Y, where Y≠ X and Y≠"historical purpose" / Y≠ "statistical purpose" / Y≠"scientific purpose"]	Decision= Deny	Decision= Deny	collection or is not for a historical purpose/statistical purpose / scientific purpose then Deny the request." Which formed the CNL "If the Action:Purpose:string is not the Resource:PurposesOfCollection:string OR the Action:Purpose:string is not a "historical purpose" / "statistical purpose" / "scientific purpose" then Deny the Access to the PersonalData." It is notable that this rule does not Grant access if the purpose of collection matches the purpose of processing or the purpose of processing is a historical purpose/statistical purpose / scientific purpose so that the decision is left for the other PDPs
3.	Purpose of collection =[X] purpose (current purpose)= [Y, where Y≠ X, Y=scientific purpose , Y≠"historical purpose" / Y≠ "statistical purpose"]	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
4.	The purpose of collection = [X] purpose (current purpose) = [Y, where Y≠ X and Y= "statistical purpose", Y≠ "historical purpose" , Y≠ "scientific purpose"]	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
5.	Purpose of collection =[X] purpose (current purpose)= [Y, where Y≠ X and Y=historical purpose, Y≠ "statistical purpose", Y≠ "scientific purpose"]	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	

5.5 Validation of the System

Once we have validated that the Legal PDP gives the correct decisions, we next tested the complete system using a set of PDPs (Legal, data subject's, controller's PDP). We developed a set of use case scenarios for the validation tests based on the following objectives:

i) Show that the Legal PDP decisions get priority over that of data subject and controller.
ii) Show that the privileged access right provided to certain stakeholders by the Legal PDP (e.g. Medical Professional, data subject) is enforced.
iii) Show that the data subject's PDP decisions get preferences over that of the controller.
iv) Show how contract based access to personal data is enforced.
v) Show access scenario to both very sensitive personal data (e.g. medical data) and less sensitive personal data (e.g. CV).
vi) Demonstrate that the important functional properties of the system are correctly integrated together, namely:
    (1) Sticky policy enforcement
    (2) Support for multiple languages

(3) Support for obligation combination
(4) Distributed policy enforcement

5.5.1 Use case scenario1 (access to medical data)

This use case shows access control scenario of a health care centre which deals with very sensitive personal data such as medical data. The validation tests based on this scenario verify the access rights of the data subject and others such as Medical Professionals, researcher or other organisational roles such as administrative officers or financial officers are being executed as expected by our system.

A patient Mr. M registers with the Kent Health Centre and fills out a registration form. He is also given a form where he mentions his policy about who can access his medical record. This is a form mainly of tick boxes so that the patient does not need to know the complex policy languages. The patient's preference is translated into the machine executable policies using a policy language.

The CRRs of the law, issuer, data subject, controller and default one are placed in order. The Legal CRRs are mentioned in Table A 1.3 of Appendix 1.
Let the **Issuer's CRRs** be

1. If the ResourceType is medical data and the requester is a Medical Professional DCA=GrantOverrides.
2. If the ResourceType is prescription, DCA=GrantOverrides.
3. If the ResourceType is medical data and the requester is not a Medical Professional DCA=DenyOverrides.

The **Data subject's CRR** is

- If the request is for my personal data DCA=DenyOverrides.

The **Controller's** is

- If the request is for any data DCA=GrantOverrides.

The **Default CRR** is

- DCA=DenyOverrides

The policies in the system are the Legal Policy, Issuer's Policy, Data subject's Policy, Controller's Policy. The Legal Policy is presented in table A1.3 of Appendix1.
The **Issuer's policy** for this case is-

- Medical Professional can change the value of MedicalObjection attribute of the medical data or of each medical record in the medical data separately for the purpose of saving the vital interest of the data subject.

The **DataSubject's Policy** rules are –

- Dr. D of Kent Health Centre can READ and WRITE my medical data.
- Researchers are allowed to READ the medical data if the data can be anonymised.

The **Controller's Policy** rules are

- Administrative Officers can READ and WRITE administrative data (such as the contact information of patients, and which doctor is treating which patient etc.) but cannot access the medical data.
- Financial Officers can READ the billing and payment information but cannot READ the medical data or administrative data.

- Medical Professionals cannot access the billing and payment information or the patient's financial information.

All the policies and Request Context (RC) used for the tests are presented in Appendix 5. Mr. M gets some problem with his lungs and his treating doctor Dr. D tries to Read the medical data of Mr. M at the "Kent Health Centre", the request of Dr. D is granted by the system. (How to determine whether a requester is a treating doctor is discussed in Chapter 4.)

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Controller PDP Decision	Final Decision
1.	Dr. D of Kent Health Canter wants to READ the medical data of Mr. M.	GrantOverrides by Legal CRR	Grant	N/A	Grant	N/A	Grant

Mr. M then has an X-ray. Dr. D enters the preliminary results into Mr. M's record as the Write request is granted for Dr. D.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Controller PDP Decision	Final Decision
2.	Dr. D of Kent Health Canter wants to WRITE the medical data of Mr. M.	GrantOverrides by Legal CRR	Grant	N/A	Grant	N/A	Grant

Dr. D suggests Mr. M to consult a specialised doctor. Mr. M goes to a Medical Consultant Dr. S at London Hospital and Dr. S of London Hospital tries to access the data at the "Kent Health Centre" and gets a 'BTG' response.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Controller PDP Decision	Final Decision
3.	Dr. S of London Hospital wants to READ the medical data of Mr. M (at Kent Health Centre).	GrantOverrides by Legal CRR	BTG	N/A	N/A	N/A	BTG

Dr. S at London Hospital suggests the patient changes his policy to allow him access or he would need to break the glass every time. The patient contacts the Kent Health Centre and tries to change his policy to allow Dr. S of London Hospital to READ/WRITE on his data which is granted by Legal policy.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
4.	Mr. M wants to UpdatePolicy of the medical data of Mr. M (at Kent Health Centre).	GrantOverrides by Legal CRR	Grant	N/A	N/A	N/A	Grant

So the Data Subject's PDP now has policy rules –

- Dr. D of Kent Health Centre can READ and WRITE my medical data.
- Researchers are allowed to READ the medical data if the data can be anonymised.
- Dr. S of London Hospital can READ and WRITE my medical data.

Now Dr. S of London Hospital tries to read Mr. M's medical record at Kent Health Organisation.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
5.	Dr. S of London Hospital wants to READ the medical data of Mr. M.	GrantOverrides by Legal CRR	BTG	N/A	Grant	N/A	Grant

Now Dr. S can read the medical data of Mr. M at Kent Health Centre. Dr. S wants to write about the treatment he is offering in that record which is now granted by the data subject's PDP.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
6.	Dr. S of London Hospital wants to WRITE the medical data of Mr. M.	GrantOverrides by Legal CRR	BTG	N/A	Grant	N/A	Grant

Mr. M is suspected to have a severe problem in his lungs and at the moment knowing this would be harmful for his mental condition since he is a patient with chronic depression as well. Dr. S at the London Medical Centre tries to change the value of the MedicalObjection attribute of Mr. M.'s medical record (in the Mr. M's medical data) containing this report to "true".

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
7.	Dr. S of London Hospital wants to WRITE the MedicalObjection attribute of one of Mr. M's Medical Record (at Kent Health Centre).	GrantOverrides by LegalCRR	N/A	Grant	N/A	N/A	Grant

Now Mr. M requests to view his medical data. The PEP first requests the PDP to see if the requester is allowed to READ the medical data at all. When the request is granted then the PEP calls the PDPs with the requests for each medical record separately. Mr. M's medical record containing record of lungs problem will have the value of the resource attribute MedicalObjection to be "true" and this resource attribute will be expanded with the request context automatically by the PEP.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decisio n	Final Decision
8.	Mr. M wants to READ the MedicalRecord X of Mr. M (at Kent Health Centre) and MedicalObjection=true .	DenyOver rides by Legal CRR	Deny	N/A	N/A	N/A	Deny

For all the medical record other than the medical record with MedicalObjection = true he is granted access.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
9.	Mr. M wants to READ the medical record Y of Mr. M (at Kent Health Centre).	GrantOve rrides by Legal CRR	Grant	N/A	N/A	N/A	Grant

So Mr. M does not know that the severe lungs problem record exists. When the Deny decision is obtained the PEP simply skips this record and proceeds to the next one.

Dr. D gently tells him about the lung condition. The report directly on screen might scare the patient and could cause harm to his mental state. But he needs to be treated. So Dr. D gently informs the patient about the condition and explains to him the situation. When the doctor finds that Mr. M is mentally strong enough and can handle the situation he requests to change the resource attribute of this medical data record to have MedicalObjection=false. For the medical record issued by Dr. S the issuer's policy and CRR will be provided by Dr. S. Let us assume that Dr. S did not specify any CRR or policy.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
10.	Dr. D wants to WRITE on the MedicalObjection attribute of one of Mr. M's medical record.	GrantOve rrides by Legal CRR	Grant	N/A	N/A	N/A	Grant

Mr. M now tries to access the medical data and none of the medical records have

MedicalObjection = true. Hence like case 9 Mr. M gets access to all the records.

When the administrator at the Kent Health Centre wants to access the medical data and is denied by the controller PDP.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
11.	Administrator of Kent Health Centre wants to READ the medical data of Mr. M.	DenyOver rides by the Legal CRR	N/A	N/A	N/A	Deny	Deny

A Medical Professional wants to access the billing data of Mr. M and is denied by the controller PDP.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
12.	Dr. L of Kent Health Centre wants to READ the Billing Data of Mr. M.	GrantOverr ides by Controller CRR	N/A	N/A	N/A	Deny	Deny

A friend of Mr. M wants to access the medical data of Mr. M and gets a NotApplicable decision.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
13.	A friend of Mr. M wants to READ the medical data of Mr. M.	DenyOverri des by Legal CRR	N/A	N/A	N/A	N/A	N/A

The NotApplicable decision is turned into a Deny by the application PEP which denies access to all requests that are not explicitly granted.

Mr. M suddenly faces an accident while he is on a holiday. He is taken to the nearest hospital H. The doctor of that hospital needs to treat him and they need to know his medical history to treat him properly. From his accompanying friend the doctor came to know that the Kent Health centre is his regular medical service provider. Dr. T of Hospital H tries to access the medical data of Mr. M at the Kent Health Centre and gets a BTG response.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decisio n	Final Decision
14.	Dr. T of Hospital H wants to READ the medical data of Mr. M (at Kent Health Centre).	GrantOve rrides by Legal CRR	BTG	N/A	N/A	N/A	BTG

Since it is an emergency situation Dr. T decides to BTG at the Kent Health Centre.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Controller PDP Decision	Final Decision
15.	Dr. T of Hospital H wants to BTG the medical data of Mr. M (at Kent Health Centre).	GrantOverrides by Legal CRR	Grant	N/A	N/A	N/A	Grant

As the glass has been broken all the activities on Mr. M's record are monitored and an email is sent to the Kent Health Centre's authority about the BTG. Now Dr. T wants to READ the medical data of Mr. M at the Kent Health Centre which is now granted.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Controller PDP Decision	Final Decision
16.	Dr. T of Hospital H wants to READ the medical data of Mr. M (at the Kent Health Centre) and BTG=true.	GrantOverrides by Legal CRR	Grant	N/A	N/A	N/A	Grant

Dr. T of Hospital H wants to write on the medical data of Mr. M at the Kent Health Centre which is now granted.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Controller PDP Decision	Final Decision
17.	Dr. T of Hospital H wants to WRITE on the medical data of Mr. M (at Kent Health Centre) and BTG=true.	GrantOverrides by Legal CRR	Grant	N/A	N/A	N/A	Grant

A friend of Mr. M tries to READ the medical data of Mr. M at the Kent Health Centre when the glass is broken. The friend is still denied access by the application PEP because the glass was only broken for Dr. T to have access.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Controller PDP Decision	Final Decision
18.	A friend of Mr. M wants to READ the medical data of Mr. M and BTG=true.	DenyOverrides by the Legal CRR	N/A	N/A	N/A	N/A	N/A

A Researcher wants to access the medical data and gets response of Grant if the data can be anonymised.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
19.	Researcher of University of Kent wants to READ the medical data of Mr. M (at Kent Health Centre).	DenyOver rides by the Legal CRR	N/A	N/A	Grant with Obligatio n to anonymis e the data	N/A	Grant if the data can be anonymis ed otherwise Deny

Mr. M relocates to Germany and registers with G Health Centre. G Health Centre tries to transfer the medical record of Mr. M from the Kent Health Centre and gets a NotApplicable response.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
20.	G Health Centre wants to Transfer the medical data of Mr. M to Germany.	DenyOverri des by the Legal CRR	N/A	N/A	N/A	N/A	N/A

The Kent Health Centre contacts Mr. M for his consent about the transfer and Mr. M provides the consent to transfer to = G Health Centre. Now G Health Centre tries to transfer the medical record of Mr. M from the Kent Health Centre which is now granted.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
21.	G Health Centre wants to Transfer the medical data of Mr. M to Germany and SubjectConsentsToTra nsferTo=G Health Centre.	GrantOverri des by the Legal CRR	Grant	N/A	N/A	N/A	Grant with obligation to TRANSFE R sticky policies

[Alternatively this transfer could be allowed by Mr. M by changing his policy at the Kent Health Centre.]

5.5.2 Use case scenario2 (contract based access to personal data)

Here three different use case scenarios of accessing personal data based on contracts are presented.

5.5.2.1 Health centre and pharmacy contract

There is a contract between the Kent Health Centre and each of the registered pharmacies to access the prescriptions of patients for providing medicines. The <ResourceType> element contains the value "prescription" to indicate what resource is being allowed to access by this contract. An authority of the Kent Health Centre and an authority of a registered pharmacy B sign the contract with their Distinguished Names (DNs). These are mentioned in the <SignerOfContract> elements of the <PartyOfContract> elements. Each <PartyOfContract> element also has an element called <OtherIdentifyingAttributesOfSigner> which contains the other identifying attributes of the signers. The name and address of Kent Health Centre and that of pharmacy B are mentioned in the <IdentifyingAttributesOfParty> elements of the <PartyOfContract> elements. The <AuthorisedRequester> element has the <role>= "Employee" and the <organisation>="pharmacy B" elements. The <role> element contains the role the requester must hold and the <organisation> element contains the name of the organisation to which the role holder must belong. This means that anyone holding the "Employee" role of "pharmacy B" is allowed to access the "prescriptions" of the "Kent Health Centre". The Kent Health Centre stores the contract information in a contract repository after validating the contract. The stored information comprises contract's identifier, validity time, resource type and the identifying information of the authorised requester and parties of the contract.

Mr. M is prescribed some medicines which he needs to buy from a registered pharmacy. Mr. M goes to a registered pharmacy B and asks for the medicine saying that his prescription belongs to the Kent Health Centre. The employee of the pharmacy B asks at the Kent Health Centre to transfer Mr. M's prescription based on the contract identifier. The application PEP of the health centre looks up the contract with the provided contract identifier from the repository and passes the validity time, resource type and the identifying attributes of the authorised requester and parties of the contract to the request context. The application PEP also maintains a table of resource attributes where each column represents a resource attribute and each row represents the resource attributes of a resource (distinguished by the RID). When a request to access a resource arrives it passes all the resource attributes (such as identifying attributes of the data subject of the resource, resource type and so on) of the requested resource to the request context. The legal PDP then matches the resource type added to the request context from the resource attributes (named ResourceType) with that obtained from the contract (named ContractResourceType) and the identifying attributes of the subject (as presented by the requester) with the identifying attributes of the authorised requester (as obtained from the contract) and the request time (passed by the PEP as an environment attribute) with the validity time (obtained from the contract) to grant the transfer. Legal PDP also checks whether the controller is a party of the contract as mentioned in Section 3.3.3. Let's assume that the CRRs and the policies are the same as mentioned in the previous section.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
22.	An employee of pharmacy B wants to TRASNFER Mr. M's prescription from the Kent Health Centre to the pharmacy B (in the UK).	GrantOverrides by Legal CRR	Grant with obligation to TRANSFER sticky policies	N/A	N/A	N/A	Grant with obligation to TRANSFER sticky policies

When the data are transferred to the Pharmacy B's system the policies of the issuer and the data subject are also passed and PDPs are started at the pharmacy's site.

A researcher wants to access Mr. M's data at the pharmacy B's site.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
23.	Researcher wants to READ the prescription of Mr. M (at the Pharmacy D's site).	GrantOverrides by the Issuer CRR	N/A	N/A	Grant with Obligation to anonymise the data	N/A	Grant if the data can be anonymised otherwise Deny

After the validity time of the contract is over the value of the <end time> element becomes less than the request time and if a transfer is requested again by the Pharmacy B (in the UK) the Legal PDP replies N/A for that. (This is because the request for transfer is to a country in the EU. For a transfer request to a non EU country a Deny would be returned for such a case).

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
24.	An employee of pharmacy B wants to TRANSFER Mr. M's prescription from the Kent Health Centre to the pharmacy B (in the UK) and the request time is NOT within the [validity time].	DenyOverrides by the default CRR	N/A	N/A	N/A	N/A	N/A

5.5.2.2 Personal contract with a health insurance company

Mr. M signs a contract with a health insurance company named HIC for covering the cost of treatment at the dental clinic D. In the contract the mentioned resource types are

SummaryOfGivenTreatment and BillingInformation. The <SubjectOfContract> element of the contract document has the identity attributes of Mr. M and the <AuthorisedRequester> element has the <role>= "Employee" and <organisation>= "HIC". Both the officer of HIC and Mr. M sign the contract and their DNs are mentioned as the <SignerOfContract> elements of the <PartyOfContract> elements. The <OtherIdentifyingAttributesOfSigner> elements have the other identifying attributes of Mr. M and that of the signer who signed the contract of behalf of HIC. The name and address of HIC is mentioned in the <IdentifyingAttributesOfParty> element.

Let us assume that the CRRs and the policies for the medical data at the dental clinic D are the same as the Kent Health Centre. One employee of HIC asks at the dental clinic D to access the billingInformation of Mr. M and presents the contract of Mr. M and HIC. The dental clinic D validates the contract by the ConVS and the ConVS returns the validity time of the contract, resource type and identifying attributes of the subject of contract, authorised requester and the parties of the contract which are added to the request context. The PEP passes the resource attributes such as the identifying attributes of the data subject, resource type and so on to the request context. The legal PDP matches the attributes of the request context obtained from the resource attributes of the resource with that obtained from the contract as mentioned earlier in Section 5.5.2.1.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decisio n	Final Decision
22.	Employee of HIC wants to READ the BillingInformation of Mr. M at the dental clinic D.	GrantO verrides by legalCR R	Grant with obligation to TRANSFER sticky policies	N/A	N/A	N/A	Grant with obligation to TRANSFE R sticky policies

5.5.2.3 Outsourcing contract

In order to provide another example for contract based access to personal data let us consider a scenario where a university U outsources the employee details to a bank B for paying the salaries to the employees. There is a contract between the university and the bank where both of them are parties of the contract. An administrator of the university and an administrator of the bank B actually sign the contract and the DNs of the signers are mentioned in the contract as <SignerOfContract> of the <PartyOfcontract> elements and the <OtherIdentifyingAttributesOfSigner> elements contain the identifying attributes of the signer other than the DNs. The <IdentifyingAttributesOfParty> elements contain the name and address of the university U and bank B to indicate that these organisations are the parties of the contract. The <ResourceType>s mentioned in the contract are the "bank details" and "salary amount" of the employees. The <role>= "Employee" and the <organisation>="University U" are mentioned in the <SubjectOfContract> element. The <role>="SeniorOfficer" and the <organisation>="Bank B" are mentioned in the <AuthorisedRequester> element, which means that only the <role>= "SeniorOfficer" of the <organisation>="Bank B" can access the <ResourceType>= "bank details" and "salary amount" of the <role>="Employee" of the <organisation> ="University U". A

<role>="SeniorOfficer" of the <organisation>="Bank B" asks for the information (bank details and salary amount) of an employee (or a number of employees one by one) at the university's site and mentions the unique identifier of the contract. As the university is also a party of the contract the university should have a copy of the contract stored in the contract repository (after validating it). The PEP of the university gets the contract with the unique contract identifier from the repository and passes the validity time, resource type and the identifying attributes of the subject of contract, authorised requester, and parties of the contract to the request context. The PEP passes all the resource attributes of the requested resource such as the identifying attributes (including the <role> = "Employee" of the <organisation> = "University U") of the data subject of the requested resource and so on to the request context. The legal PDP matches the attributes of the request context obtained from the resource attributes of the resource with that obtained from the contract as mentioned earlier in Section 5.5.2.1.

Let us assume that there is no policy by the issuer/ data subject or the controller (but if there were they could not override the Legal PDP's decision) and the default CRR is DenyOverrides.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
26.	A Senior Officer of Bank B requests to TRANSFER Mr. M's bank details from the university U to the bank B (in UK).	GrantOverrides by Legal CRR	Grant with obligation to TRANSFER sticky policies	N/A	N/A	N/A	Grant with obligation to TRANSFER sticky policies

When the data are transferred the policies and resource attributes (such as the identifying attributes of the data subject, resource type etc.) are transferred with that. The data, resource attributes and policies are stored and PDPs are started at the receiving site with the received policies. The data subject Mr. M wants to READ his bank details at the Bank B. The legal PDP returns Grant as the data subject is requesting to read his data and the identifying attributes of the subject (requester) matches the data subject's identifying attributes.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
27.	Mr. M wants to READ his bank details at the bank B.	GrantOverrides by Legal CRR	Grant	N/A	N/A	N/A	Grant

A <role>="Junior Officer" of the <organisation>= "Bank B" asks for the information (bank details) of an employee at the university's site and mentions the unique identifier of the contract. As the <role>= "Junior Officer" is not mentioned as an authorised requester, the request cannot be granted based on the contract.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
28.	A Junior Officer of Bank B requests to TRANSFER Mr. M's bank details from the university U to the Bank B (in UK).	DenyOver rides by the default CRR	N/A	N/A	N/A	N/A	N/A

The obtained NotApplicable decision is turned into a Deny by the PEP.

5.5.3 Use case scenario 3 (access to CV and degree certificate)

This use case shows access control scenario of an employment agency which deals with personal data like CV.

Mr. M is looking for a job and wants to get a service from an employment agency jobs.com. Jobs.com only allows members of the agency to apply for jobs or upload their CVs. Mr. M joins as a member. Mr. M, the Data Subject, is the Issuer of his CV. Mr. M is presented with a list of potential employers and he can choose from the list who he wants to share his CV with. From his choice an Issuer policy is created. Suppose that Mr. M chooses to share his CV with Companies C and D.

Legal CRR- as before.

Issuer's CRR/ DataSubject's CRR- DCA= DenyOverrides

Controller's CRR- DCA=GrantOverrides

Default CRR- DCA=DenyOverrides

Legal Policy- as before.

Issuer's policies/ DataSubject's policies

1. Only Company C and D can read and transfer my CV.
2. Only Data subject can update / write/read his data.

Controller's policies

1. Member can upload CV
2. Member can apply for job
3. Any staff of jobs.com can read any CV

After Mr. M registers with jobs.com one of the staff of jobs.com asks to read the CV of Mr. M which is granted.

NO	Request Context	DCA	Legal PDP Decision	Issuer /DataSubject PDP Decision	Contro-ller PDP Decision	Final Decision
29.	Staff of jobs.com wants to READ the CV of Mr. M.	DenyOverrides by Issuer/DataSubject CRR	N/A	N/A	Grant	Grant

Now the staff member gets to know about the educational information of Mr. M from his CV and knows that Mr. M has a degree from University U. The staff member of jobs.com (located in the UK) asks the University U for a copy of the degree certificate. At the site of the university the issuer (university) has a CRR saying if the request is for a degree certificate DCA=DenyOverrides and has a policy saying "Only the issuer can write or update degree certificate". The data subject Mr. M has no CRR but a policy saying "if the request is to transfer my degree certificate then deny the request with an obligation to ask for consent." Let us assume that as a controller the university does not have any policy or CRR and the default CRR is DenyOverrides.

Hence at the university U's site

Legal CRR- as before.

Issuer's CRR- if the request is for a degree certificate DCA=DenyOverrides

Default CRR- DCA=DenyOverrides

Legal Policy- as before.

Issuer's policy – Only the issuer can write or update degree certificate. Issuer and data subject can read degree certificate.

DataSubject's policy- if the request is to transfer my degree certificate then deny the request with an obligation to ask for consent.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Final Decision
30.	Staff of jobs.com (located in the UK) wants to TRANSFER the degree certificate of Mr. M.	DenyOverrides by Issuer CRR	N/A	N/A	Deny with an obligation to ask for consent	Deny with an obligation to ask for consent

After getting the decision the PEP of University U requests for consent of Mr. M to transfer the degree certificate to jobs.com. When the data subject Mr. M consents to the transfer the PEP of University U adds that consent as a resource attribute SubjectConsentsToTransfer = jobs.com. Jobs.com is informed either by the University U or the data subject that the consent of data subject has been updated. The staff of jobs.com requests to transfer the degree certificate of Mr. M again. This time the PEP of University U appends the resource attribute SubjectConsentsToTransfer with the request context and the request is granted.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
31.	Staff of jobs.com wants to TRANSFER the degree certificate of Mr. M SubjectConsentsToTransfer = jobs.com.	GrantOverrides by Legal CRR	Grant with obligation to TRANSFER sticky policies	N/A	Deny with an obligation to ask for consent	N/A	Grant with obligation to TRANSFER sticky policies

Along with the degree certificates the policies of the data subject and the policies of the issuer are sent as sticky policies and PDPs are started at the jobs.com's site with those policies.

Mr. M requests to read his degree certificate at jobs.com.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Contro-ller PDP Decision	Final Decision
32.	Mr. M wants to READ his degree certificate.	GrantOverrides by Legal CRR	Grant	Grant	N/A	N/A	Grant

Mr. M request to update his degree certificate at the jobs.com. Although the policy of Mr. M allows updating his data by himself that decision is overridden by the Issuer PDP decision.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	Data Subject PDP Decision	Control-ler PDP Decision	Final Decision
33.	Mr. M wants to Update his degree certificate.	DenyOverrides by Issuer CRR	N/A	Deny	Grant	N/A	Deny

A request comes from the Company B to view the CV of Mr. M

NO	Request Context	DCA	Legal PDP Decision	Issuer /DataSubject PDP Decision	Controller PDP Decision	Final Decision
34.	Company B wants to READ the CV of Mr. M.	DenyOverrides by Issuer/DataSubject CRR	N/A	Deny	N/A	Deny

A staff of jobs.com tries to write the CV of Mr. M which is denied by the Issuer PDP.

NO	Request Context	DCA	Legal PDP Decision	Issuer /DataSubject PDP Decision	Controller PDP Decision	Final Decision
35.	Staff of jobs.com wants to WRITE the CV of Mr. M.	DenyOverrides by Issuer/ DataSubject CRR	N/A	Deny	N/A	Deny

A staff of jobs.com tries to Update the CV of Mr. M.

NO	Request Context	DCA	Legal PDP Decision	Issuer /DataSubject PDP Decision	Controller PDP Decision	Final Decision
36.	Staff of jobs.com wants to Update the CV of Mr. M.	DenyOverrides by Issuer/ DataSubject CRR	N/A	Deny	N/A	Deny

A request comes from Company C to read the CV of Mr. M which is granted.

NO	Request Context	DCA	Legal PDP Decision	Issuer /DataSubject PDP Decision	Controller PDP Decision	Final Decision
37.	Company C wants to READ the CV of Mr. M.	DenyOverrides by Issuer/ DataSubject CRR	N/A	Grant	N/A	Grant

Company C is interested in Mr. M. So it asks to transfer the CV of Mr. M to its system.

NO	Request Context	DCA	Legal PDP Decision	Issuer /DataSubject PDP Decision	Controller PDP Decision	Final Decision
38.	Company C wants to TRANSFER the CV of Mr. M to Company C located in Germany.	DenyOverrides by Issuer /DataSubject CRR	N/A	Grant with obligation to TRANSFER sticky policies	N/A	Grant with obligation to TRANSFE R sticky policies

When the CV of Mr. M is transferred to Company C the policy of Mr. M also goes with it and a PDP is started with those policies.

Company C wants to work in collaboration with Company X. Company C wants to share the CV of Mr. M with company X and company X sends a request to the system of Company C to READ the CV of Mr. M's CV. Let us assume that company C has a policy to let the company X read a CV. In this case the Controller's Grant decision is overridden by the Data subject's Deny decision.

NO	Request Context	DCA	Legal PDP Decision	Issuer /DataSubject PDP Decision	Contro-ller PDP Decision	Final Decision
39.	Company X wants to READ the CV of Mr. M.	DenyOverrides by Issuer/ DataSubject CRR	N/A	Deny	Grant	Deny

Mr. M wants to READ his CV at the Company C's site which is granted by the Legal PDP.

NO	Request Context	DCA	Legal PDP Decision	Issuer /DataSubject PDP Decision	Controller PDP Decision	Final Decision
40.	Mr. M wants to READ his CV.	GrantOverrides by Legal CRR	Grant	N/A	N/A	Grant

Mr. M tries to update his CV which is granted by Issuer's PDP

NO	Request Context	DCA	Legal PDP Decision	Issuer /DataSubject PDP Decision	Controller PDP Decision	Final Decision
41.	Mr. M wants to Update his CV.	GrantOverrides by Issuer CRR	N/A	Grant	N/A	Grant

Company C tries to update the CV of Mr. M which is denied by the Issuer's PDP.

NO	Request Context	DCA	Legal PDP Decision	Issuer /DataSubject PDP Decision	Controller PDP Decision	Final Decision
42.	Company C wants to Update CV of Mr. M.	DenyOverrides by issuerCRR	N/A	Deny	N/A	Deny

Company C requests to transfer the degree certificate of Mr. M from jobs.com which is denied.

NO	Request Context	DCA	Legal PDP Decision	Issuer PDP Decision	DataSubject PDP Decision	Controller PDP Decision	Final Decision
43.	Company C wants to TRANSFER degree certificate of Mr. M.	DenyOverrides by Issuer CRR	N/A	N/A	Deny with an obligation to ask for consent	N/A	Deny with an obligation to ask for consent

After getting the decision the PEP of jobs.com asks for consent from Mr. M. He refuses to give consent and so the resource attribute SubjectConsentsToTransfer is not updated and the decision is not changed if the company C requests again.

5.5.4 Use case scenario 4 (testing system properties)

Here a simple use case is used where a controller allows its members to submit data (e.g. class notes) and share with his/her friends. The objective of this use case is to demonstrate that various functional properties of the system are integrated correctly together, namely:

(1) Sticky policy enforcement
(2) Support for multiple languages
(3) Support for obligation combination
(4) Distributed policy enforcement

The aim of this use case is to provide some simple tests so that it is easily possible for anyone to repeat them to examine P-PAAS system properties.

i) For this use case the XACML Legal PDP remains as it is and the system is configured with the controller policy saying "1. Member can SUBMIT policy 2. Member can TRANSFER sticky policy" (as presented in Appendix 4.6.1).

ii) A request is sent by MyFriend role to Read a personal data having RID value "rid-1". The request context is presented in Appendix 4.6.2. A "NotApplicable" response in obtained. The Read request is sent before the data subject submits a sticky policy and hence gets this response.

iii) A member sends a SUBMIT request with an XACML policy saying that "MyFriend Role can Read my data with an obligation to LogTheRequest" for RID= "rid-1" and PolicyID="sticky-policy-1" as presented in Appendix 4.6.3. This will mean to instruct the system that this PolicyID is stuck with this RID and the policy identified by this PolicyID will be evaluated for the access requests for the resource identified by this RID. [This demonstrates the submission of sticky policy with the authorisation request]

iv) The request context of Appendix 4.6.2, representing a request by MyFriend role to Read a personal data having RID value "rid-1", is sent again and this time it replies Grant with obligation to LogTheRequest. This is demonstrating that the policy coming as a XACML sticky policy is being stored and evaluated for access request for the resource (identified by RID). [This demonstrates the enforcement of the sticky policy which was submitted in step (iii) above]

v) A SUBMIT request is sent by member with a PERMIS policy saying "MyFriend can Write my data with an obligation to Send e-mail" for the same RID ("rid-1") with PolicyID="sticky-policy-2"as presented in Appendix 4.6.4. [This demonstrates the submission of sticky policy with multiple policy languages, in this case a PERMIS policy is submitted and in step (iii) an XACML policy was submitted.]

vi) A Write request by MyFriend is submitted (as presented in Appendix 4.6.5) which evaluates the PERMIS policy and gets a Grant response. [This demonstrates that the system evaluates policies of multiple policy languages, XACML and PERMIS.]

vii) A TRANSFER request is sent by the member for "rid-1" as provided in Appendix 4.6.6. All the policies that were submitted for "rid-1" are transferred as sticky policies. Appendix 4.6.7 is showing the response obtained after sending the request presented in Appendix 4.6.6 which contains the previously submitted sticky policies. [Since the system can receive sticky policies coming with the request context and can also send sticky policies with the response, this ensures that the system can be used *for distributed enforcement of* sticky policies.]

viii) The member sends another SUBMIT request for "rid-1" with an XACML policy same as the policy presented in Appendix 4.6.3 with a different obligation "SendE-mail". Note that the PolicyID will have the value "sticky-policy-3" in this case to make it distinguished from the previous policy.

ix) The request context presented in Appendix 4.6.2, representing a request by MyFriend role to Read a personal data having RID value "rid-1", is sent again and this time the response is obtained with two obligations, "LogTheRequest" and "SendE-mail" as presented in 4.6.8. [This demonstrates that the system can combine obligations from two different policies.]

5.6 Performance Tests of the Authorisation Service

In this section we present performance tests of the prototype implementation of the P-PAAS authorisation system.

5.6.1 Performance tests in a single machine

The authorisation service was installed in a single machine whose configuration was: dual core processors each with cpu speed = 2.99 GHz; cache=2 MB; the total memory size was 2 GB and the machine was running Ubuntu 10.04. The configuration of the client machine: Dual core processor with 2.53 GHz CPU speed and memory size was 2.98 GB and was running Windows XP. As client software the SOAPUI[12] was used. The client was operating across a local area network. Two series of tests were performed. The first set tested how long it took for the authorisation system to store and retrieve a user's sticky policy and make an authorisation decision. The second set tested the reduction in performance for an increasing number of embedded PDPs running inside the authorisation system. All the embedded PDPs in this case were XACML PDPs, which we know from previous research (Chadwick, Su and Laborde 2008) is not the fastest of PDPs. Hence, these figures can be improved upon by using faster PDPs (such as PERMIS).

5.6.1.1 Making authorisation decisions by the authorisation server

For the first set of tests, the legal PDP had 1 rule in it and the data controller's PDP had 1 rule in it. The user's sticky policy also had 1 rule in it and this policy was added in test 2 and transferred in test 4.

Test 1 was a request to read personal information when only the Legal and controller's PDP were running. Test 2 was to store personal information along with a sticky policy. Two policies were evaluated (Legal and controller's) and the request was granted and the user's policy was then stored by the authorisation system. Test 3 was a request by a third party to read the user's personal information. Three policies (user's, Legal and controller's) were interrogated and the decision was granted. Test 4 was a request to move the user's personal data to another system. Three policies were interrogated, the request was granted, and the user's sticky policy was returned with the decision.

Each test was run sequentially 500 times and then the mean time and standard deviation were calculated. Any results that varied over 3 times the standard deviation from the mean time

were removed as outliers. This resulted in 2.25% of the results being discarded on average. The results are presented in Table 5.15.

Table 5.15: Time (in ms) to make an authorisation decision and/or store/retrieve a sticky policy

Test	Mean	Std Dev	% Discarded
1. Authz decision with 2 PDPs	6.34	0.74	2.47
2. Request to store personal data and sticky policy	15.02	2.04	3.4
3. Authz decision with 3 PDPs	14.82	1.47	2
4. Request to transfer data (and retrieve sticky policy)	28.81	2.12	2.6

From the results one can see that:
- the time taken to store personal data and a sticky policy was approximately 2.5 times the time taken to make the authorisation decision
- the time taken to retrieve a sticky policy for transfer is approximately 14 ms twice the time taken to store it, and that approximately half of this time is on decision making using 3 PDPs.

5.6.1.2 Increasing the number of PDPs embedded in the authorisation server

In this second series of tests the authorisation server was configured with an increasing number of policies/PDPs, each containing 1 rule. In the first test the authorisation server only had 1 policy configured into it (the legal PDP with 1 rule). In the second test the authorisation server was configured with the Legal policy and the data controller's policy (with 1 rule). In the third test the authorisation server had 3 policies: the user's sticky policy (with 1 rule), and the legal and controller's configured policies. In the subsequent tests an additional sticky policy was added. The number of rules in each additional PDP is kept the same for this set of tests so that the results are not affected by different numbers of rules in different PDPs.
In each case we measured the time taken for an authorisation decision to be made when a third party asked to read a record, and a Grant decision was obtained. This necessitated all configured policies being interrogated and the Master PDP determining the overall decision, using a GrantOverrides combining rule. The results are shown in Table 5.16.

Table 5.16: Time (in ms) to make an authorisation decision for different number of PDPs

Test	Mean	Std Dev	% Discarded	Mean PDPi – MeanPDPi-1
1 PDP	5.27	0.51	4.07	
2 PDPs	6.34	0.74	2.47	1.07
3 PDPs	14.82	1.47	2	8.48
4 PDPs	22.64	1.53	2	7.82
5 PDPs	30.37	1.9	1.4	7.73
6 PDPs	38.30	1.99	1.6	7.93

[12] http://www.soapui.org/

7 PDPs	46.47	2.32	1.2	8.17
8 PDPs	54.26	2.28	2.4	7.79
9 PDPs	62.51	2.59	0.8	8.25
10 PDPs	69.61	2.55	1.2	7.1

From the results one can observe that the time taken to make an authorisation decision increases linearly with the number of sticky policy PDPs, and for this configuration, each PDP adds approximately 8ms. The reason the 2nd PDP added only a small amount of time (approx. 1 ms) is that it is a built in PDP and not a sticky policy PDP. Figure 5-5 shows the response time vs. the number of PDPs.

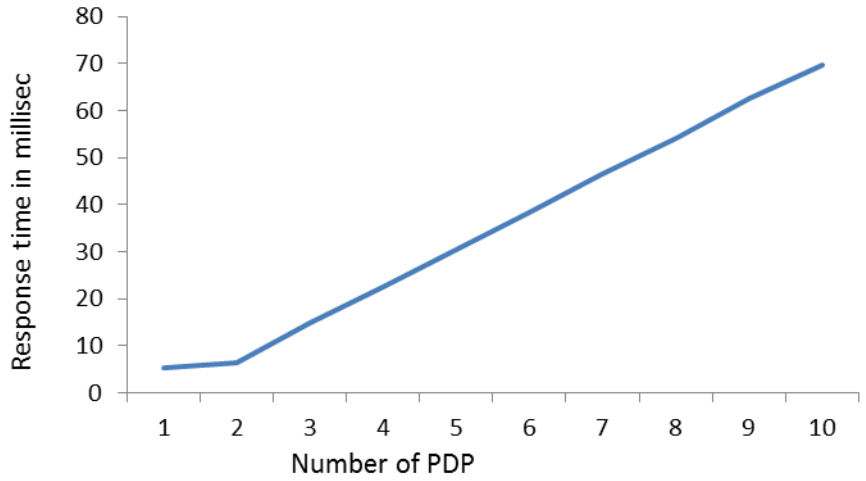

Figure 5-5. Response time as number of PDPs is increased incrementally.

5.6.2 Performance tests in a cloud

The P-PAAS can be used by an IaaS provider as a part of the infrastructure offering to platform and application developers. The latter may then use this to further develop privacy preserving applications. Our authorisation infrastructure does not obviate the need for trust, but rather is built on the assumption that cloud providers can be trusted to the extent that they wish to provide an automated infrastructure that can easily enforce each other's policies reliably and automatically. If they cannot support an incoming sticky policy they should inform the sender of the fact. Our system provides IaaS providers with an application independent authorisation infrastructure that makes it easy for them to enforce users' privacy policies without having to write a significant amount of new code themselves. Furthermore the user has the potential for more complete control over his/her privacy than now, in that the infrastructure allows the user to specify a complete privacy policy along with obligations.

A cloud provider can use the P-PAAS infrastructure as a service for its users to allow them to set their own privacy policies, thereby guaranteeing no unauthorised access to their data. The infrastructure ensures that these privacy policies are stuck to the users' data and access to the data is controlled by the relevant policies even if the data is transferred between cloud

providers or services. The authorisation system has been installed in a private cloud server to test its applicability and performance in the cloud.

The authorisation infrastructure was up and running on a small cloud server, whose configuration was: 2 CPUs each with 4 core and each core with hyper-threading support (equivalent to 16 core) with each core's speed being 2.53 GHz; cache size for each core is 12 MB; memory is 25 GB in total. The configuration for each Virtual Machine inside the cloud is: cpu speed= 2.53 GHz; cache=4 MB; memory=~256MB. We ran the authorisation infrastructure as one VM.
The configuration of the client machine, which performed the role of the cloud medical application running in a different cloud was: cpu speed = 2.99 GHz; cache=2 MB; memory= 2 GB. Note that there was no underlying medical application, as all the client machine did was make authorisation decisions requests across a LAN and receive authorisation decisions. Because the client was operating across a local area network we realise that the results will be slower than if the client was running inside another VM of the same cloud service, but they will be faster than if the cloud application was being run by a different cloud provider accessing the storage service over the Internet.
Two series of tests were performed. The first set tested how long it took for the authorisation system to store and retrieve a user's sticky policy and make an authorisation decision. The second set tested the reduction in performance for an increasing number of PDPs running inside the authorisation system.

5.6.2.1 Making authorisation decisions by the authorisation server

The same sets of policies were used as mentioned in Section 5.6.1.1.
Each test was run sequentially 500 times and then the mean time and standard deviation were calculated. Any results that varied over 3 times the standard deviation from the mean time were removed as outliners. This resulted in 2.27% of the results being discarded on average. The results are presented in Table 5.17.

Table 5.17: Time (in ms) to make an authorisation decision and/or store/retrieve a sticky policy

Test	Mean	Std Dev	% Discarded
1. Authz decision with 2 PDPs	5.40	0.73	3.8
2. Request to store personal data and sticky policy	13.84	2.0	1.9
3. Authz decision with 3 PDPs	7.41	0.82	1.2
4. Request to transfer data (and retrieve sticky policy)	11.38	0.9	3.0

From the results one can see that:
- the time to retrieve a sticky policy for transfer is approximately the same as the time to store it along with personal data.
- the cloud service performs faster than the single machine due to its greater CPU power and memory. In particular, adding a third PDP does not slow the cloud service down significantly.

5.6.2.2 Increasing the number of PDPs embedded in the authorisation server

The setting of PDP and policies are same a mentioned in Section 5.6.1.2.
The results are shown in Table 5.18.

Table 5.18: Time (in ms) to make an authorisation decision for different number of PDPs

Test	Mean	Std Dev	% Discarded	Mean PDPi – MeanPDPi-1
1 PDP	4.11	0.31	5.4	
2 PDPs	5.40	0.73	3.8	1.29
3 PDPs	7.19	1.35	5.8	1.79
4 PDPs	10.17	0.49	5.6	2.98
5 PDPs	13.06	0.95	0.8	2.89
6 PDPs	15.63	1.11	1.0	2.57
7 PDPs	18.57	1.08	2.12	2.94
8 PDPs	21.34	1.2	1.9	2.77
9 PDPs	23.93	1.4	0.79	2.59
10 PDPs	25.54	1.5	1.0	1.61

From the results one can observe that there is a linear increase in time as the number of PDPs increases, and for this particular configuration, each PDP adds an additional 2.6 ms.

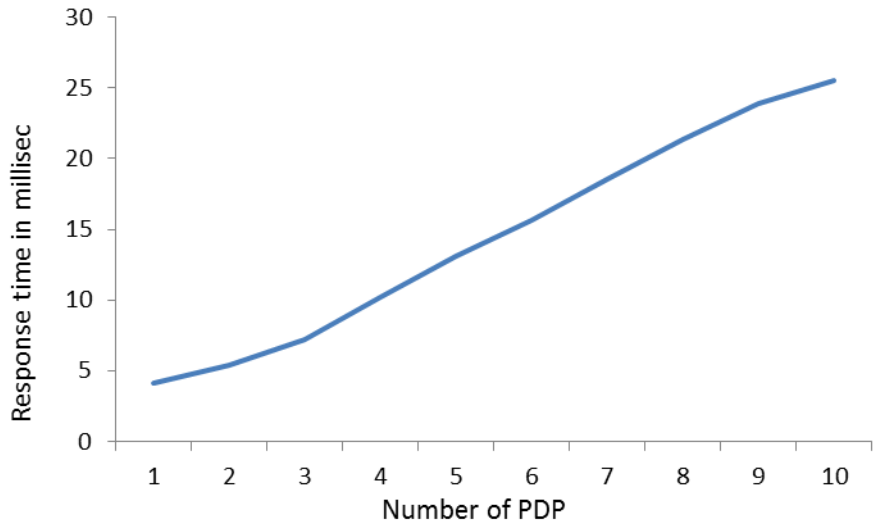

Figure 5-6. Response time as number of PDPs is increased incrementally.

5.6.3 Performance tests for increasing number of rules

A series of tests are conducted to see when the number of rules in a policy is increased whether the overhead of increasing the number of PDPs becomes insignificant or not. The performance of the authorisation service is tested using only one embedded PDP in a single

machine set up as discussed in Section 5.6.1. The PDP was set at the configuration time so that it does not add any overhead for starting a PDP with a sticky policy. The number of rules is increased at each test and the time to get a response for a request is measured. The results are shown in Table 5.19.

Table 5.19: Time (in ms) to make an authorisation decision for different number of rules

Test	Mean	Std Dev	% Discarded	Time per rule
1 Rule	5.9	0.7	5.2	5.9
10 Rules	15.58	0.9	5.2	1.56
100 Rules	69.26	1.59	4.2	0.69
1000 Rules	581.19	8.12	0.98	0.58

We can see that as the number of rules increases, the time to reach a decision increases. There is an overhead in the operation of the authorisation service and to a first approximation we may assume this is fixed. Hence, as the number of rules increases, the overhead per rule decreases until eventually it will become negligible per rule. If one assumes the time for 1 rule approximately equals the overhead, and subtract this time from the other measurements, we can still see that the time per rule for 100 rules (without overhead) (0.634 ms) is still more than the time per rule for 1000 rules even when the latter includes the overhead (0.581 ms). It can be observed from Figure 5-7 that time taken to evaluate a single rule decreases as the number of rules increases and for the existing configuration we reach nearly a steady state of approximately 0.576 ms per rule for 999 rules. So there is still something else which is causing the time taken per rule to decrease as the number of rules increase. This could be the Java machine optimisation code that is continually refining the optimisation.

It can further be observed from Table 5.16 and 5.19 that the mean time to get a decision from 1 PDP with 100 rules (69.26 ms) is equivalent to getting a decision from 10 PDPs each having 1 rule (69.61 ms) in the tested configuration.

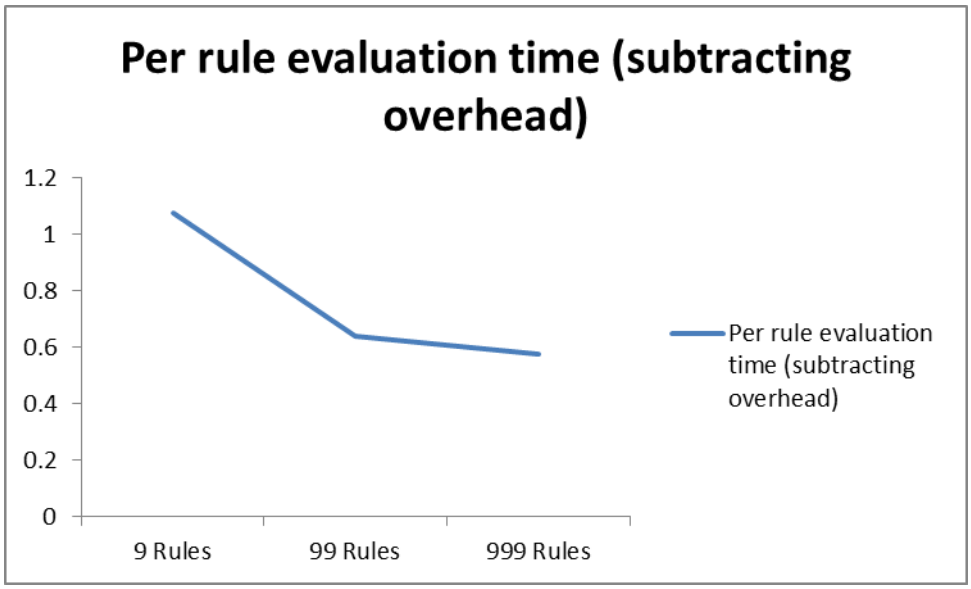

Figure 5-7. Per rule evaluation time with increasing number of rules.

5.7 Conclusion

In this chapter the implementations of the authorisation service and the components (such as, ConVS and PEP) and the implementation of the Legal access control and conflict resolution policies are discussed. Above 100 test cases have been performed for the validation of the Legal rules. Examples of the validation tests of one of the Legal policy rules are presented here. The rest of the test cases are presented in Appendix 1. The validation tests of the overall system are performed based on a number of use case scenarios. For the validation tests the authorisation decisions returned by the system are matched against the expected decisions. The request context for each of the test case is presented in Appendix 5. The results of the performance tests of the system in a single computer and in a cloud computer are also presented here. The performance results show that in both cases, adding an additional sticky policy PDP increases the processing time of the authorisation server linearly, so that the system scales well. They also show that increasing the number of rules in a single PDP increases the processing time, but the time per rule decreases logarithmically until a steady state is reached, so this also scales well.

Chapter 6

6 Conclusions

This chapter concludes the thesis. The contributions and limitations of the research are discussed here. The research avenues that have been stimulated by our research are revealed in this chapter as well.

6.1 Research Contributions

The privacy of personal data has been endangered to a great extent due to the widespread increase and use of modern computers and high speed internet connections. The goal of this research was to design an authorisation system that provides privacy of personal data based on the privacy and access control policies from multiple authorities, such as the law, data subject as well as the issuer and controller of the data. The net result is the design, testing and validation of the Privacy-Protecting Advanced Authorisation System (P-PAAS). The contributions of this research are as follows:

- The P-PAAS system accommodates policies written in multiple policy languages, as demonstrated by building a system comprising both PERMIS PDPs and XACML PDPs;
- The P-PAAS system supports independent policies written by multiple autonomous authorities, specifically the law, data subject, data controller and data issuer;
- The P-PAAS system has a strategy for dynamically resolving conflicts among the decisions returned by multiple policies obtained from various independent authorities, by determining which conflict resolution rule to use based on the current request context;
- The P-PAAS system provides a mechanism for the distributed enforcement of a data subject's privacy policy by using sticky policies and obligations to ensure that a remote system can either enforce the subject's policy or will refuse to accept the subject's personal data;
- Overall the P-PAAS system has more functional features than any other authorisation system. Besides the functionalities described above, the system also comprises some other components such as the CVS for validating credentials, the application independent obligations service for evaluating "before" type obligations, and the ConVS for validating contracts to allow access to and transfer of personal data based on signed contracts.
- Finally, we have developed a methodology for the semi-automatic extraction of access control rules from legislation, have demonstrated this by extracting access control rules from the EU DPD in both the XACML and PERMIS policy languages, and have written a program for the automatic conversion of controlled natural language (CNL) rules to XACML rules.

6.2 Comparison of P-PAAS With Other Systems

Table 6.1 provides a summary and comparison of the features (mentioned in Section 2.3.1) of the reviewed privacy protecting authorisation systems (presented in Sections 2.3.2, 2.2.9 and 2.2.10) with that of our proposed system P-PAAS. The first row of the table presents the names of the models and the corresponding section numbers containing the description of that. In this table 'y' represents 'yes', 'n' represents 'no' and 'l' represents 'limited'. It can be observed from the table that none of the tools except the P-PAAS addressed the requirements of privileged access by certain parties to personal data, the conditions to be met for transferring data, contract based access to personal data, multiple policy language enforcement and inclusion of Legal policies with the highest priority. Among the eighteen requirements of a privacy protecting authorisation system, as mentioned in Section 2.3.1, fourteen requirements were satisfied by our system which was not done by any other system. None of the systems' design was based on the thorough examination of the EU DPD, hence they are poor in meeting the requirements.

Table 6. 1: Summary and comparison of features of privacy protecting systems

	EPAL (2.3.2.1)	P-RBAC (2.3.2.2)	Privacy research of HP (2.3.2.3)	Privacy research of EnCoRe (2.3.2.4)	P3p (2.3.2.5)	PPL (2.3.2.6)	PuRBAC (2.3.2.7)	Model of Al-Harabi and Osborn (2.3.2.8)	Model of Byun and Li (2.3.2.9)	Model of Smari et al. (2.3.2.10)	Model of Xu et al. (2.3.2.11)	PPAS (2.3.2.12)	Model of Goyal et al. (2.3.2.13)	PERMIS (2.2.10)	XACML (2.2.9)	P-PAAS
1) Purpose specification	y	y	y	y	y	y	y	y	y	y	n	y	y	y	y	y
2) Consent specification	y	y	y	y	y	y	l	y	y	n	y	y	l	l	l	y
3) Limited Collection	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
4) Limited Use and Limited Disclosure	y	y	y	y	n	y	l	l	l	y	y	y	y	y	y	y
5) Limited Retention	l	l	l	y	n	l	y	n	n	y	n	l	l	l	l	l
6) Accuracy	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
7) Safety	n	n	n	l	n	l	n	n	n	n	n	n	y	n	n	n
8) Openness	l	l	l	l	n	n	n	n	n	y	n	n	n	n	n	y
9) Compliance	l	n	n	l	l	n	n	n	n	n	n	n	n	n	n	n
10) User's control	y	y	y	y	n	y	l	l	l	n	y	y	y	y	y	y
11) Enforcing privacy obligation	y	y	y	y	n	y	y	n	l	l	n	y	y	y	y	y
12) Privileged access	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	y
13) Transferring data	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	y
14) Contract based access	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	y
15) Simple user interaction	y	l	l	y	y	y	n	n	n	n	n	n	n	n	n	y
16) Multiple policy language support	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	y
17) Distributed enforcement	y	n	y	y	n	y	n	n	n	y	n	n	n	n	n	y
18) Inclusion of Legal policy with the highest priority	n	n	n	l	n	n	n	n	n	n	n	n	n	n	n	y

6.3 Research Limitations

This section discusses the limitations of our current research.

1. 100% extraction of the authorisation rules from the EU DPD was not possible due to the fuzzy nature of the legislative language, some rules depend upon or can be overridden by national legislations and some rules require human judgements. Full realisation of automated access control rules from the EU DPD may not be possible until fully deterministic explanation of certain legal rules can be obtained.
2. Among the eighteen requirements of privacy protecting authorisation systems, as mentioned in Section 2.3.1, fourteen requirements were satisfied by our system and the four other requirements (limited collection, accuracy, safety and compliance) were not possible to satisfy by our system. Further research is needed to ensure the requirements to collect only the necessary data and not more than that, or data are stored accurately and safely, or to verify the compliance of the privacy protection offered by the data controller with the rules of law.
3. The sticky policy(/ies) has only one signature element. It is assumed that the controller of the sending system would pack the data and policies together and sign the contents before sending that over to the receiving system.
4. The current implementation of the authorisation system is not capable of including the conflict resolution policies dynamically from sticky policies and instead of having a single conflict resolution PDP it has separate conflict resolution PDPs for the law and the controller, which are executed sequentially. However, it is possible to implement conflict resolution PDPs from sticky conflict resolution policies by using the similar approach used for sticky access control policies and call them dynamically based on the request. Furthermore, it does not evaluate the user's sticky policy when a data item is first received. However, it can store the sticky policies and the subsequent requests to access the data item are evaluated using the stored sticky policies. The current implementation only supports the FirstApplicable, DenyOverrides and GrantOverrides DCA. The current implementation can combine obligations from the configured PDPs or from the sticky policy PDPs, but not from all the PDPs together.
5. The current design also assumes that other systems would be using the P-PAAS compliant architecture and would operate in a trustworthy manner by not violating the protocols. Although the model ensures the distributed enforcement when the protocols are maintained properly but it does not detect any violation of protocols.

6.4 Recommendations and Future Work

This work has stimulated further research in the following areas-

1. The method that we used to extract authorisation rules from the EU DPD can be used on other legislation (for example, the country specific implementation of the EU DPD, such as, Data Protection Act 1998 of United Kingdom) to verify whether the methodology works or not.
2. Although some other previous works, which attempted to get requirements from legal texts, are highly dependent on the representation of the legislative language and may not be suitable for the texts of other legislation (Breaux and Anton 2008), an

experiment can be performed by applying those methodologies on the EU DPD to show how fit those methodologies are in extracting the access control rules from the EU DPD.

3. With the P-PAAS it is possible to accommodate authorisation policies from legislation of two different jurisdictions by having two Legal PDPs. More research on international laws needs to be done to make sure which Legal PDP policy should be applicable in what situation.
4. More research can be done on the 9 rules out of the 53 rules that are discarded as no deterministic rules are possible to make out of them. There might be a possibility of creating an alert or obligation to require human intervention while an access case is encountered for which these rules might have an issue.
5. Further research needs to be done on the monitoring and auditing of privacy protection provided by an authorisation system, for example, whether the obligations are being enforced or the privacy policies are being evaluated with appropriate priorities and so on.
6. More research needs to be done on the usability of a privacy preserving system. A system may become useless if the user cannot use the system comfortably.
7. Contracts are deemed to be confidential only among the parties. More research can be done on protecting confidentiality based on anonymity. The digital contract that is presented in this thesis contains the identity information of the parties and the data subject which are all personal data. Although it is assumed that the parties would make the contract available only between themselves it would be much better with respect to confidentiality and privacy protection if an anonymity or pseudo-anonymity based protection can be given to these data.
8. Further research is needed on obligation combination. XACML has possibility of ignoring some obligations which is solved by our system, as it combines all the obligations of all the PDPs having similar effects. Our design was for the situation when a policy says (e.g. controller) to permit with an obligation to write on the log and another policy says (e.g. data subject) to permit with an obligation to e-mail, then both will be enforced. However, this may create a problem for situations when the police officer is allowed to access the data of a person and the person's policy also says allow the access with an obligation to e-mail him the details. The enforcement of such an obligation by the data subject may not be allowed by the law. Different kinds of obligation enforcement might be needed for such situations to determine any conflict with the Legal obligations and perhaps autonomous priority needs to be given to Legal obligations. Further research is needed in conflict resolution among obligations and obligation co-ordination.

Bibliography

Agrafiotis, I., S. Creese, M. Goldsmith, N. Papanikolaou, M. C. Mont, and S. Pearson. "Defining consent and revocation policies." *IFIP/PrimeLife Summer School 2010.* Helsingborg, Sweden: Springer, 2010.

Al-Harbi, A. L., and S. L. Osborn. "Mixing privacy with role-based access control." *Proceedings of The Fourth International Conference on Computer Science and Software Engineering.* ACM, 2011. 1--7.

Allmer, T. "A critical contribution to theoretical foundations of privacy studies." *Journal of Information, Communication & Ethics in Society* 9, no. 2 (2011): 83--101.

Ananthanarayanan, R., M. Mohania, and A. Gupta. "Management of conflicting obligations is self-protecting policy-based systems." *2nd International Conference on Autonomic Computing, 2005 (ICAC 2005).* IEEE, 2005. 274--285.

Andress, J. *The basics of information security: understanding the fundamentals of InfoSec in theory and practice.* Syngress, 2011.

ANSI. "Information technology - role based access control." *American National Standards Institute, International Committee for Information Technology Standards (ANSI/INCITS)* (American National Standards Institute, International Committee for Information Technology Standards (ANSI/INCITS)), 2004.

Ardagna, C. A., S. De Capitani di Vimercati, S. Paraboschi, E. Pedrini, and P. Samarati. "An XACML-based privacy-centered access control system." *First ACM workshop on Information security governance.* ACM, 2009. 49--58.

Assembly, UN General. "Universal declaration of human rights." *Resolution adopted by the General Assembly,* 1948.

"Authorization C API Developer Reference." *IBM Tivoli Access Manager for e-business.* November 2003. https://publib.boulder.ibm.com/tividd/td/ITAME/SC32-1355-00/en_US/PDF/am51_authC_devref.pdf.

AXIS2. *Apache AXIS2.* 2004. http://axis.apache.org/axis2/java/core/ (accessed August 1, 2013).

Barth, A., J.C. Mitchell, and J. Rosenstein. "Conflict and combination in privacy policy languages." *ACM Workshop on Privacy in the Electronic Society.* ACM, 2004. 45--46.

BBC news. 27 April 2011. http://www.bbc.co.uk/news/world-13214078 (accessed July 30, 2013).

BBC news. 24 August 2010. http://www.bbc.co.uk/news/business-11070217 (accessed July 30, 2013).

BBC news. 22 July 2009. http://news.bbc.co.uk/1/hi/business/8162787.stm (accessed July 30, 2013).

BBC news. 01 July 2014. http://www.bbc.com/news/technology-28107277 (accessed July 07, 2014).

Bekara, K., and M. Laurent. "A semantic information model based on the privacy legislation." *IEEE Conference on Network and Information Systems Security (SAR-SSI).* IEEE, 2011. 1--6.

Bell, D. E., and L. J. La Padula. *Secure computer system: unified exposition and Multics interpretation.* Technical report, DTIC Document, 1976.

Bettini, C. "Obligation monitoring in policy management." *3rd International Workshop on Policies for Distributed Systems and Networks.* IEEE, 2002. 2--12.

Biba, K. J. *Integrity considerations for secure computer systems.* Technical report, DTIC, 1977.

Blaze, M., J. Feigenbaum, and J. Ioannidis. *The KeyNote trust-management system, version 2.* Technical report, IETF RFC, 1999.

Borking, J. J. "Why adopting Privacy Enhancing Technologies (PETs) takes so much time." In *Computers, Privacy and Data Protection: an Element of Choice*, 309--341. Springer, 2011.

Breaux, T. D., and A. I. Anton. "Mining rule semantics to understand legislative compliance." *ACM Workshop on Privacy in the Electronic Society.* ACM, 2005. 51--54. A

Breaux, T. D., and A.I. Anton. "A systematic method for acquiring regulatory requirements: a frame-based approach." *6th International Workshop on Requirements for High Assurance Systems.* 2007.

Breaux, T. D., and A.I. Anton. "Analyzing regulatory rules for privacy and security requirements." *IEEE Transactions on Software Engineering*, 2008: 5--20.

—. "Deriving semantic models from privacy policies." *6th IEEE International Workshop on Policies for Distributed Systems and Networks.* IEEE, 2005. 67--76. B

Breaux, T. D., M.W. Vail, and A.I. Anton. "Towards regulatory compliance: extracting rights and obligations to align requirements with regulations." *14th IEEE International Conference on Requirements Engineering.* IEEE, 2006. 49--58.

Byun, J., and N. Li. "Purpose based access control for privacy protection in relational database systems." *The VLDB Journal, Springer*, 2008: 603--619.

Casellas, N., et al. "Ontological semantics for data privacy compliance: the NEURONA project." *AAAI Spring Symposium: Intelligent Information Privacy Management.* Palo Alto, California, 2010.

Cavoukian, A. *Privacy by design.* Ontario: Information and Privacy Commissioner of Ontario, Canada, 2009.

Chadwick, D. W., and K. Fatema. "An advanced policy based authorisation infrastructure." *5th ACM Workshop on Digital Identity Management (DIM'09).* ACM, 2009. 81--84.

Chadwick, D. W., and S. F. Lievens. "Enforcing "Sticky" security policies throughout a distributed application." *ACM Workshop on Middleware Security.* ACM, 2008. 1--6.

Chadwick, D. W., and L. Su. "Use of WS-TRUST and SAML to access a Credential Validation Service." 13 November 2009.

Chadwick, D. W., L. Su, and R. Laborde. "Coordinating access control in grid services." *Concurrency and Computation: Practice and Experience*, 2008: 1071--1094.

Chadwick, D. W., S. Otenko, and T. A. Nguyen. "Adding support to XACML for multi-domain user to user dynamic delegation of authority." *International Journal of Information Security*, 2009: 137--152.

Chadwick, D. W., W. Xu, S. Otenko, R. Laborde, and B. Nasser. “Multi-session separation of duties (MSoD) for RBAC.” *First International Workshop on Security Technologies for Next Generation Collaborative Business Applications (SECOBAP'07).* Istanbul, Turkey, 2007.

Chadwick, D.W, G. Zhao, S. Otenko, R. Laborde, L. Su, and T.A. Nguyen. “PERMIS: a modular authorization infrastructure.” *Concurrency and Computation: Practice and Experience*, 2008: 1341--1357.

Chanda, R. “A Cautionary note about policy conflict resolution.” *Military Communications Conference.* Washington DC: IEEE, 2006.

Charalambides, M., et al. “Dynamic policy analysis and conflict resolution for DiffServ quality of service management.” *Network Operations and Management Symposium, 2006 (NOMS 2006).* IEEE, 2006. 294--304.

Charmaz, K. “Grounded theory.” *Strategies of Qualitative Inquiry*, 2003.

Choi, H., S. Lee, and H. Lee. “Design and implementation of a policy-based privacy authorization system.” In *Intelligence and Security Informatics*, 129--140. Springer, 2006.

CIFAS report. 15 Oct 2010. http://www.cifas.org.uk/press_release_tenp (accessed July 30, 2013).

Clarke, R. “What's privacy.” *Roger Clarke's web-site.* August 2006. http://www.rogerclarke.com/DV/Privacy.html (accessed July 30, 2013).

Crampton, J., and C. Morisset. “Towards a generic formal framework for access control systems.” *CoRR* abs/1204.2342 (2012).

Crampton, J., and M. Huth. “A framework for the modular specification and orchestration of authorization policies.” In *NordSec*, 155-170. Springer, 2010. A

Crampton, J., and M. Huth. “An authorization framework resilient to policy evaluation.” In *Computer Security--ESORICS 2010*, by D Gritzalis, B Preneel and Marianthi T., 472--487. Springer, 2010. B

“Data protection act .” 1998. http://www.ico.gov.uk/upload/documents/library/data_protection/detailed_specialist_guides/160408_v1.0_determining_what_is_personal_data_-_quick_reference_guide.pdf (accessed July 30, 2013).

Demchenko, Y., O. Koeroo, C. de Laat, and H. Sagehaug. “Extending XACML authorization model to support policy obligations handling in distributed application.” *6th International Workshop on Middleware for Grid Computing.* ACM, 2008.

di Vimercati, S. D. C., P. Samarati, and S. Jajodia. “Policies, models, and languages for access control.” In *Databases in Networked Information Systems*, 225--237. Springer, 2005.

Directive 95/46/EC . Directive, European Parliament, 1995.

Dunlop, N., J. Indulska, and K. Raymond. “Methods for conflict resolution in policy-based management systems.” *Enterprise Distributed Object Computing Conference.* IEEE, 2003. 98--109.

EC. “Convention for the protection of individuals with regard to automatic processing of personal data.” 1981. http://conventions.coe.int/Treaty/en/Treaties/Html/108.htm (accessed August 1, 2013).

EnCoRe. *EnCoRe ensuring consent and revocation.* 2008. http://www.encore-

project.info/ (accessed August 1, 2013).

Fatema, K., D. W. Chadwick, and B. Van Alsenoy. "Extracting access control and conflict resolution policies from European data protection law." *IFIP/PrimeLife International Summer School.* Trento, Italy: Springer, September 5-9, 2011. 59-72.

Fatema, K., D. W. Chadwick, and S. Lievens. "A multi-privacy policy enforcement system." In *Privacy and Identity Management for Life*, 297--310. Springer, 2011.

Ferraiolo, D. F., and D. R. Kuhn. "Role based access control." *15th National Computer Security Conference.* 1992. 554--563.

Ferreira, A., et al. "How to securely break into RBAC: the BTG-RBAC model." *Computer Security Applications Conference, 2009 (ACSAC'09).* IEEE, 2009. 23--31.

Fischer-Hubner, S. *IT-security and privacy: design and use of privacy-enhancing security mechanisms.* Springer-Verlag, 2001.

Fisler, K., S. Krishnamurthi, L. A. Meyerovich, and M. C. Tschantz. "Verification and change-impact analysis of access-control policies." *27th International Conference on Software engineering.* ACM, 2005. 196--205.

Gama, P., and P. Ferreira. "Obligation policies: an enforcement platform." *Sixth IEEE International Workshop on Policies for Distributed Systems and Networks.* IEEE, 2005. 203--212.

Gellman, Robert. *WPF report: privacy in the clouds: risks to privacy and confidentiality from cloud computing.* World Privacy Forum, 2009.

Godik, S., A. Anderson, B. Parducci, P. Humenn, and S. Vajjhala. *OASIS eXtensible access control 2 markup language (XACML).* Technical report, OASIS, 2002.

Gong, C., J. Liu, Q. Zhang, H. Chen, and Z. Gong. "The characteristics of cloud computing." *39th International Conference on Parallel Processing Workshops (ICPPW).* IEEE, 2010. 275--279.

Gopalan, R., A. Anton, and J. Doyle. "UCONlegal: a usage control model for HIPAA." *2nd ACM SIGHIT International Health Informatics Symposium.* ACM, 2012. 227--236.

Goyal, H., A. Deodia, and A. Gupta. "A system for privacy policy enforcement & access control for web applications." *International Conference on Information and communication Technology in Electrical Science (ICTES 2007).* Chennai, Tamil Nadu, India: IEEE, 2007.

IBTimes report. 10 January 2011. http://www.ibtimes.com/proposed-online-id-system-raises-privacy-concerns-253357 (accessed July 30, 2013).

Irwin, K., T. Yu, and W. H. Winsborough. "On the modeling and analysis of obligations." *13th ACM Conference on Computer and Communications Security.* ACM, 2006. 134--143.

ISO. "Security frameworks for open systems: access control framework." *ISO/IEC 10181-3:*, 1996.

Itani, W., A. Kayssi, and A. Chehab. "Privacy as a service: privacy-aware data storage and processing in cloud computing architectures." *8th IEEE International Conference on Dependable, Autonomic and Secure Computing, 2009 (DASC'09).* IEEE, 2009. 711--716.

Jones, A.J.I., and M. Sergot. "Deontic logic in the representation of law: towards a methodology." *Springer Journal of Artificial Intelligence and Law*, 1992: 45--

64.

Jones, A.J.I., and M. Sergot. “Formal specification of security requirements using the theory of normative positions.” In *Computer Security ESORICS 92*, 103--121. Springer, 1992.

Karjoth, G., and M. Schunter. “A privacy policy model for enterprises.” *15th IEEE Computer Foundations Workshop.* IEEE, 2002.

Karjoth, G., M. Schunter, and M. Waidner. “Platform for enterprise privacy practices: privacy-enabled management of customer data.” *Privacy Enhancing Technologies.* San Francisco, CA: Springer, 2003. 69--84.

Karjoth, G., M Schunter, and M Waidner. “Privacy-enabled services for enterprises.” *13th International Workshop on Database and Expert Systems Applications.* IEEE, 2002. 483- 487.

Karjoth, G., M. Schunter, and E.V. Herreweghen. “Translating privacy practices into privacy promises—how to promise what you can keep.” *Policies for Distributed Systems and Networks (POLICY 2003).* IEEE, 2003. 135-146.

Katt, B., X. Zhang, R. Breu, M. Hafner, and J. Seifert. “A general obligation model and continuity: enhanced policy enforcement engine for usage control.” *13th ACM Symposium on Access Control Models and Technologies.* ACM, 2008. 123--132.

Kiyavitskaya, N., et al. “Automating the extraction of rights and obligations for regulatory compliance.” In *Conceptual Modeling-ER 2008*, 154--168. Springer, 2008.

Kounga, G., M.C. Mont, and P. Bramhall. “Extending XACML access control architecture for allowing preference-based authorisation.” *7th International Conference on Trust, Privacy and Security in Digital Business (TrustBus'10).* Springer, 2010. 153--164.

Lengwiler, M. *Privacy, justice and equality: the history of privacy legislation and its significance for civil society.* Discussion paper, Berlin: University of Zurich, 2004.

Li, N., and J.C. Mitchell. “RT: a role-based trust-management framework.” *The Third DARPA Information Survivability Conference and Exposition (DISCEX III).* Washington DC: IEEE, 2003.

Li, N., et al. “Access control policy combining: theory meets practice.” *Proceedings of the 14th ACM Symposium on Access Control Models and Technologies.* ACM, 2009. 135-144.

Li, N., H. Chen, and E. Bertino. “On practical specification and enforcement of obligations.” *Second ACM Conference on Data and Application Security and Privacy.* ACM, 2012. 71--82.

Li, N., J.C. Mitchell, and W.H Winsborough. “Design of a role-based trust-management framework.” *In Proceedings of 2002 IEEE Symposium on Security and Privacy.* California: IEEE, 2002. 114-130.

Li, N., W.H. Winsborough, and J.C. Mitchell. “Distributed credential chain discovery in trust management.” *8th ACM Conference on Computer and Communications Security (CCS-8).* ACM, 2001. 156–165.

Lin, D., and A. Squicciarini. “Data protection models for service provisioning in the cloud.” *Proceedings of the 15th ACM Symposium on Access Control Models and Technologies.* ACM, 2010. 183--192.

Linington, P., Z. Milosevic, and K. Raymond. “Policies in communities: extending the

ODP enterprise viewpoint." *Enterprise Distributed Object Computing Workshop, 1998 (EDOC'98).* IEEE, 1998. 14--24.

Lupu, E. C., and M. Sloman. "Conflicts in policy-based distributed systems management." *IEEE Transactions on Software Engineering*, 1999: 852--869.

Ma, C., G. Lu, and J. Qiu. "Conflict detection and resolution for authorization policies in workflow systems." *Journal of Zhejiang University SCIENCE A*, 2009: 1082--1092.

Malhotra, N.K., S.K. Sung, and J. Agarwal. "Internet users' information privacy concerns (IUIPC): the construct, the scale, and a causal model." *Information Systems Research* 15, no. 4 (2004): 336--355.

Martin, E., J.H. Hwang, T. Xie, and V. Hu. "Assessing quality of policy properties." *Computer Security Applications Conference, 2008 (ACSAC 2008).* IEEE, 2008.

Masoumzadeh, A., and J.B.D. Joshi. "PuRBAC: purpose-aware role-based access control." In *On the Move to Meaningful Internet Systems: OTM 2008*, 1104--1121. Springer, 2008.

Massacci, F., M. Prest, and N. Zannone. "Using a security requirements engineering methodology in practice: the compliance with the Italian data protection legislation." *Elsevier Computer Standards & Interfaces*, Elsevier: 445--455.

Mazzoleni, P., B. Crispo, S. Sivasubramanian, and E. Bertino. "XACML policy integration algorithms." *ACM Transactions on Information and System Security (TISSEC)*, 2008.

Mohan, A. *Design and implementation of an attribute based authorization management system.* PhD thesis, Georgia Institute of Technology, 2011.

Mohan, A., and D. M. Blough. "An attribute-based authorization policy framework with dynamic conflict resolution." *Proceedings of the 9th Symposium on Identity and Trust on the Internet.* Gaithersburg, MD: ACM, 2010. 37--50.

Mont, M. C., and F. Beato. "On parametric obligation policies: enabling privacy-aware information life cycle management in enterprises." *Eighth IEEE International Workshop on Policies for Distributed Systems and Networks, 2007. POLICY'07.* IEEE, 2007. 51--55.

Mont, M. C., S. Pearson, and P. Bramhall. "Towards accountable management of identity and privacy: sticky policies and enforceable tracing services." *14th International Workshop on Database and Expert Systems Applications.* IEEE, 2003. 377--382. A

Mont, M. C., S. Pearson, and P. Bramhall. "Towards accountable management of privacy and identity information." In *Computer Security ESORICS 2003*, 146--161. Springer, 2003. B

Mont, M. C., S. Pearson, G. Kounga, Y. Shen, and P. Bramhall. *On the management of consent and revocation in enterprises: setting the context.* Technical Report, Bristol: HPL-2009-49, HP Labs, 2009.

Mont, M.C. "Dealing with privacy obligations: important aspects and technical approaches." *International Conference on Trust and Privacy in Digital Business.* Zaragoza: Springer, 2004. 120-131.

Mont, M.C., and R. Thyne. *Privacy policy enforcement in enterprises with identity management solution.* Technical report, Bristol: Trusted Systems Laboratory, HP Laboratories Bristol, HPL-2006-72, 2006.

Mont, M.C., S. Pearson, S. Creese, M. Goldsmith, and N. Papanikolaou. "EnCoRe: towards a conceptual model for privacy policies." *PrimeLife/IFIP Summer*

School 2010: Privacy and Identity Management for Life. Helsingborg, Sweden: Springer, 2010.

Mowbray, M., and S. Pearson. "A client-based privacy manager for cloud computing." *4th International ICST Conference on Communication System Software and Middleware.* ACM, 2009.

NBC news. 16 Januray 2008. http://www.msnbc.msn.com/id/22685515/ (accessed July 30, 2013).

Nelson, R., M. Schunter, M.R. McCullough, and J.S. Bliss. "Trust on demand - enabling privacy, security, transparency, and accountability in distributed systems." *33rd Research Conference on Communication, Information and Internet Policy (TPRC).* Arlington VA, 2005.

Nelson, T. *Margrave: An improved analyzer for access-control and configuration policies.* Ph.D. Thesis, Worcester Polytechnic Institute, 2010.

Ni, Q., A. Trombetta, E. Bertino, and J. Lobo. "Privacy aware role based access control." *12th ACM Symposium on Access Control Models and Technologies (SACMAT '07).* New York, USA: ACM, 2007. 41-50.

Ni, Q., E. Bertino, and J. Lobo. "An obligation model bridging access control policies and privacy policies." *13th ACM Symposium on Access Control Models and Technologies.* ACM, 2008. 133-142.

OASIS. "WS-Trust 1.3." *OASIS Standard*, 2007.

OAuth. "The OAuth 2.0 Authorization Protocol." *The OAuth 2.0 Authorization Protocol.* 9 September 2012. (accessed August 1, 2013).

OECD. "OECD guidelines on the protection of privacy and transborder flows of personal data." Technical report, 1980.

Ortalo, R. "Using Deontic logic for security policy specification." *LAAS CNRS*, 1996.

Otenko, S., D. Chadwick, and E. Thornton. *PERMIS Java API Cookbook.* 23 December 2002. http://sec.cs.kent.ac.uk/permis/documents/PERMISAPICookbook.html.

Papanikolaou, N., S. Creese, M. Goldsmith, M. C. Mont, and S. Pearson. "ENCORE: towards a holistic approach to privacy." *2010 International Conference on Security and Cryptography (SECRYPT).* Athens: IEEE, 2010. 1--6.

Papanikolaou, N., S. Pearson, and M.C. Mont. "Towards natural-language understanding and automated enforcement of privacy rules and regulations in the cloud: survey and bibliography." In *Secure and Trust Computing, Data Management, and Applications*, 166--173. Springer, 2011.

Park, J., and R. Sandhu. "The UCON ABC usage control model ." *ACM Transactions on Information and System Security (TISSEC)*, 2004: 128-174.

Pearson, S. "Taking account of privacy when designing cloud computing services." *ICSE Workshop on Software Engineering Challenges of Cloud Computing (CLOUD'09).* IEEE, 2009. 44-52.

Pearson, S., and A. Charlesworth. "Accountability as a way forward for privacy protection in the cloud." In *Cloud Computing*, 131--144. Springer, 2009.

Pearson, S., Y. Shen, and M. Mowbray. "A privacy manager for cloud computing." In *Cloud Computing*, 90--106. Springer, 2009.

Penycate, J. *BBC news.* 18 June 2001. http://news.bbc.co.uk/1/hi/uk/1395109.stm. (accessed July 30, 2013).

PERMIS Standalone Authorisation Server. 2011. http://sec.cs.kent.ac.uk/permis/downloads/Level3/standalone.shtml (accessed

August 1, 2013).
Preibusch, S. “Guide to measuring privacy concern: Review of survey and observational instruments.” *International Journal of Human-Computer Studies*, 2013: 1133-1143.
“Public law 93-579.” 31 Dec 1974. http://www.llsdc.org/attachments/wysiwyg/544/PL093-579.pdf (accessed August 1, 2013).
Randic, M., M. Kunstic, and B. Blaskovic. “Object by value transfer mechanisms for obligation policy enforcement object loading.” *12th IEEE Mediterranean Electrotechnical Conference, 2004 (MELECON 2004).* IEEE, 2004. 727--730.
Robinson, N., H. GRAUX, M. BOTTERMAN, and L. VALERI. “Review of EU data protection directive: summary.” May 2009. http://www.ico.org.uk/upload/documents/library/data_protection/detailed_specialist_guides/review_of_eu_dp_directive_summary.pdf (accessed August 1, 2013).
Russello, G., C. Dong, and N. Dulay. “Authorisation and conflict resolution for hierarchical domains.” *Policies for Distributed Systems and Networks, 2007(POLICY'07).* IEEE, 2007. 201--210.
Sabahi, F. “Cloud computing security threats and responses.” *3rd International Conference on Communication Software and Networks(ICCSN)* . IEEE, 2011. 245--249.
SAMLv2.0. “Assertions and protocols for the OASIS security assertion markup language (SAML) V2.0.” 10 August 2010. http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-cs-01-en.pdf.
SAML v 2.0; “Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0.” 19 August 2014. http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-en.html.
Sandhu, R. S., and P. Samarati. “Access control: principle and practice.” *Communications Magazine, IEEE*, 1994: 40--48.
Sandhu, R. S., D. Ferraiolo, and R. Kuhn. “The NIST model for role-based access control: towards a unified standard.” *ACM Workshop on Role-Based Access Control.* ACM, 2000.
Sandhu, R. S., E. J. Coyne, H. L. Feinstein, and C. Youman. “Role-based access control models.” *Computer, IEEE*, 1996: 38--47.
Schunter, M., and C. V. Berghe. “Privacy injector - automated privacy enforcement through aspects.” *6th Workshop on Privacy Enhancing Technologies.* Cambridge, United Kingdom: Springer, 2006. 99--117.
Smari, W. W., J. Zhu, and P. Clemente. “Trust and privacy in attribute based access control for collaboration environments.” *11th International Conference on Information Integration and Web-based Applications & Services.* ACM, 2009. 49--55.
Sun's XACML Implementation. 21 June 2006. http://sunxacml.sourceforge.net/ (accessed September 24, 2014).
Syukur, E., S. W. Loke, and P. Stanski. “Methods for policy conflict detection and resolution in pervasive computing environments.” *Policy Management for Web Workshop.* Chiba, Japan, 2005.
Tavani, H. T. “Philosophical theories of privacy: implications for an adequate online privacy policy.” *Metaphilosophy* (Wiley Online Library) 38, no. 1 (2007): 1--

22.

Times higher education news. 23 May 2014. http://www.timeshighereducation.co.uk/news/nottingham-apologises-after-personal-data-leak/2013528.article (accessed July 07, 2014).

Trabelsi, S., A. Njeh, L. Bussard, and G. Neven. "The Ppl engine: a symmetric architecture for privacy policy handling." *W3C Workshop on Privacy and Data Usage Control.* W3C, 2010.

Trabelsi, S., J. Sendor, and S. Reinicke. "Ppl: Primelife privacy policy engine." *IEEE International Symposium on Policies for Distributed Systems and Networks (POLICY), 2011.* IEEE, 2011. 184--185.

Um-e-Ghazia, R. Masood, M.A. Shibli, and M. Bilal. "Usage control model specification in XACML policy language - XACML policy engine of UCON." *CISIM.* Venice, Italy, 2012. 68-79.

Voice of America news. 01 November 2009. http://www1.voanews.com/english/news/science-technology/a-13-2008-04-29-voa44.html (accessed July 30, 2013).

"W3C XML signature syntax and processing 2nd edition." 10 June 2008. http://www.w3.org/TR/xmldsig-core/ (accessed August 1, 2013).

W3C: the platform for privacy preferences 1.0 (P3P 1.0). Technical Report, W3C, 2002.

Wang, J., Y. Zhao, S. Jiang, and J. Le. "Providing privacy preserving in cloud computing." *International Conference on Test and Measurement, 2009 (ICTM'09).* IEEE, 2009. 213--216.

Wang, L., D. Wijesekera, and S. Jajodia. "A logic based framework for attribute based access control." *ACM Workshop on Formal Methods in Security Engineering (FMSE '04).* Washington DC: ACM, 2004.

Waterman, K. K. "Pre-processing legal text: policy parsing and isomorphic intermediate representation." In *AAAI Spring Symposium: Intelligent Information Privacy Management*. 2010.

Wu, R., G. J. Ahn, and H. Hu. "Towards HIPAA-compliant healthcare systems." *2nd ACM SIGHIT International Health Informatics Symposium.* ACM, 2012. 593--602.

XACMLv2. "OASIS eXtensible Access Control Markup Language Version 2.0." 2005. http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf (accessed August 1, 2013).

XACMLv3. "OASIS eXtensible Access Control Markup(XACML) Version 3.0." 2013. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf (accessed August 1, 2013).

Xiao, X., A. Paradkar, S. Thummalapenta, and T. Xie. "Automated extraction of security policies from natural-language software documents." *20th ACM International Symposium on the Foundations of Software Engineering (SIGSOFT).* ACM, 2012.

Xu, F., J. He, X. Wu, and J. Xu. "A privacy-enhanced access control model." *International Conference on Wireless Communications and Trusted Computing (NSWCTC'09).* IEEE, 2009. 703--706.

—. "A user-centric privacy access control model." *2nd International Symposium on Information Engineering and Electronic Commerce (IEEC).* IEEE, 2010. 1--4.

Yuan, E., and J. Tong. "Attribute Based Access Control (ABAC) for web services."

International Conference on Web Services (ICWS). IEEE, 2005.

Zhu, J., and W. W. Smari, "Attribute based access control and security for collaboration environments." *Aerospace and Electronics Conference (NAECON 2008).* IEEE, 2008. 31--35.

Appendix 1: Conversion of Legal Policies

Table A 1. 1: Extracting Legal authorisation rules

No	Legal rules	Access Control rule / Comments
1.	Article 6.1(a): personal data must be processed fairly and lawfully;	This rule is related to authorisation as it mentions an action (processing) on personal data but it is not capable of saying, based on which specific condition, the action is allowed and so it is discarded.
2.	Article 6.1(b): personal data must be collected for specified, explicit and legitimate purposes and not further processed in a way incompatible with those purposes. Further processing of data for historical, statistical or scientific purposes shall not be considered as incompatible provided that Member States provide appropriate safeguards.	The system structure of P-PAAS helps to ensure that the personal data are collected for specified, explicit purposes and those purposes of collection have become parts of a resource attribute of the personal data that are being collected. To ensure the enforcement of this Legal rule the following authorisation rule is formed, "If the requested purpose of processing does not match with any of the original purposes of collection or is not for a historical purpose/ statistical purpose/ scientific purpose then deny the request ". It should be mentioned that it does not grant access only if the purposes match rather it denies access if the purposes don't match. The data subject and the other authority's PDPs may have detailed policies about by whom and for what purposes the processing is allowed and those policies will allow the access.
3.	Article 6.1(c): personal data must be adequate, relevant and not excessive in relation to the purposes for which they are collected and/or further processed;	It is not feasible to ensure the adequacy or relevance or excessiveness of data in relation to a purpose by an access control rule. So this rule can't form an access control rule. Our system has a RID for each personal data item. Request to access a data item is made by referencing the RID and the decision is returned only for that requested RID. RIDs can be specified for any granularity of the data. Furthermore within the same RID more granularities can be provided based on some resource attributes such as resource type. Hence, the system provides a mechanism to control access only to adequate, relevant data items.
4.	Article 6.1(d): personal data must be accurate and, where necessary, kept up to date; every reasonable step must be taken to ensure that data which are inaccurate or	It is not practical to ensure the accuracy of data by an authorisation rule. The authorisation rule, "A data subject can send a data update request with

	incomplete, having regard to the purposes for which they were collected or for which they are further processed, are erased or rectified.	an obligation to log the request" will ensure that the data subject can send an update request if s/he finds that the data are not accurate or not up-to-date. This request will notify the controller about the condition of the data and the controller can take reasonable steps by justifying the request to erase or rectify the data.
5.	Article 6.1(e): personal data must be kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the data were collected or for which they are further processed. Member States shall lay down appropriate safeguards for personal data stored for longer periods for historical, statistical or scientific use.	This forms an authorisation rule, "If the validity time of data is earlier than the requested time i.e. the request is made after the validity time of the data is over then deny the request." The second part of the Legal rule allows longer validity time for certain purposes but that extended validity time will depend on the national implementation of the rule.
6.	Article 7.(a): personal data may be processed if the data subject has unambiguously given his consent.	Since our authorisation system is capable of having detailed policies from the data subject specifying who for what purposes are allowed to access his/her personal data, those policies from the data subject work as consents of the data subject. Hence, the authorisation policy saying, "A data subject can submit a policy / update a policy" ensures that by submitting and updating a policy the data subject can give, update or revoke consents.
7.	Article 7.(b): personal data may be processed if processing is necessary for the performance of a contract to which the data subject is party or in order to take steps at the request of the data subject prior to entering into a contract; or	The first part of this Legal rule forms an authorisation rule, "When the purpose is performance of a contract and a party of the contract is the data subject and the requester is mentioned as an authorised requester of the contract then grant access to the resource mentioned in the contract". To be able to access a personal data item the requester must be mentioned as an authorised requester of the contract. This constraint was added to the rule to make sure that not anyone can have access to a personal data item just by showing a contract where the data subject is a party. Furthermore, the contract should permit access to the resource for which the contract was signed. How the access based on a contract would be authorised is mentioned in details in Chapter 3 and Chapter 5. The second part of the Legal rule makes an authorisation rule "When the purpose is entering into a contract and the data subject requested to process the resource then grant the access".
8.	Article 7.(c): personal data may be processed if	This forms an authorisation rule, "Personal data

	processing is necessary for compliance with a legal obligation to which the controller is subject	can be accessed when there is a data access mandate."
9.	Article 7.(d) personal data may be processed if processing is necessary in order to protect the vital interests of the data subject;	Saving the life of a data subject in the case of an emergency is an example of protecting the vital interests of the data subject. An authorisation rule is formed saying, "Medical Professionals can Break the Glass to Read or Write medical data." Break the glass is the ability to override access controls in the case of emergency in order to gain access to the data which is normally denied to the requester. This rule is an example of accessing personal data to save the vital interest of the data subject. There can be more such examples which will need to be added by the controller depending on the data it handles and the national implementation of the EU DPD.
10.	Article 7.(e) personal data may be processed if processing is necessary for the performance of a task carried out in the public interest or in the exercise of official authority vested in the controller or in a third party to whom the data are disclosed;	An authorisation rule is formed, "Entities with a specific role (e.g. social security authority) can access a specific resource type (e.g., personal data related to pensions) and if the purpose is the performance of a task of public interest (e.g., social security administration) or an exercise of the official authority." This rule is an example of allowing access to personal data to satisfy public interest. The controller can add more such rules depending on the data it handles and the national implementation of the EU DPD.
11.	Article 7.(f) personal data may be processed if processing is necessary for the purposes of the legitimate interests pursued by the controller or by the third party or parties to whom the data are disclosed, except where such interests are overridden by the interests for fundamental rights and freedoms of the data subject which require protection under Article 1 (1).	The balance of interest is not feasible to present in one single policy, so it does not form an access control rule.
12.	Article 8.1: Member States shall prohibit the processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, trade-union membership, and the processing of data concerning health or sex life.	As the rule provides restrictions on processing sensitive personal data this forms a conflict resolution rule with a DenyOverrides DCA but no access control rule (i.e. the decision of the Legal access control PDP will be NotApplicable) when a request comes for accessing sensitive personal data (as discussed in 4.4.2). The rule "If the request is for a sensitive personal data then DCA=DenyOverrides" makes sure that a Deny decision will get priority over any other decisions.
13.	Article 8.2.(a): Paragraph 1 (Article 8.1) shall not apply where: the data subject has given his explicit consent to	The 1st part of this Legal rule does not form any access control rule or conflict resolution rule. In

	the processing of those data, except where the laws of the Member State provide that the prohibition referred to in paragraph 1 may not be lifted by the data subject's giving his consent;	this case the assumption is that the data subject will provide conflict resolution rules and thus the data subject's choices will override the controller's choices. Consequently the access to personal data will be protected by the data subject's consent when there is no decision or DCA from the legal PDP. The 2nd part of the rule could form an access control rule such as - if the resource type is X then deny access where X is configurable. However, the value of X depends on the national implementations of the law which may also have more conditions imposed on it and those would also need to be added. Since it requires examination of national implementations of the EU DPD it is not possible to form a complete access control rule from it at the moment. When there is no national law in place to prohibit access to personal data when there is a consent from the data subject, then the access decision will be denied or granted by the data subject's policy maintained in his/her PDP
14.	Article 8.2.(b): Paragraph 1 (Article 8.1) shall not apply where processing is necessary for the purposes of carrying out the obligations and specific rights of the controller in the field of employment law in so far as it is authorised by national law providing for adequate safeguards;	There is not enough information in this paragraph to form an authorisation rule feasibly as it requires examining national laws.
15.	Article 8.2.(c): Paragraph 1 (Article 8.1) shall not apply where processing is necessary to protect the vital interests of the data subject or of another person where the data subject is physically or legally incapable of giving his consent	Similar to the rule obtained from article 7.(d)an authorisation rule is formed saying, "Medical Professionals can Break the Glass to Read or Write on medical data."
16.	Article 8.2.(d): Paragraph 1 (Article 8.1) shall not apply where processing is carried out in the course of its legitimate activities with appropriate guarantees by a foundation, association or any other non-profit-seeking body with a political, philosophical, religious or trade-union aim and on condition that the processing relates solely to the members of the body or to persons who have regular contact with it in connection with its purposes and that the data are not disclosed to a third party without the consent of the data subjects;	This forms the rule "Organisation X (e.g. Temple/Church) can access the personal data of type Y (where Y is related to X) (e.g. Cast information) of the subjects who are the members of X." This rule needs to be configured by the controller depending on the national implementations of the law. Another example can be- "The Labour party can access the information of trade union membership of its members" and so on.
17.	Article 8.2.(e): Paragraph 1 (Article 8.1) shall not apply where the processing relates to data which are manifestly made public by the data subject or is necessary for the establishment, exercise or defence of legal claims.	It forms an authorisation rule, "Personal data can be accessed or processed when there is a data access mandate." In order to establish, exercise and defend a legal claim a data access mandate can be provided following the appropriate legal

		procedures. Note that the procedure of obtaining a data access mandate requires human interactions.
18.	Article 8.3: Paragraph 1 (Article 8.1) shall not apply where processing of the data is required for the purposes of preventive medicine, medical diagnosis, the provision of care or treatment or the management of health-care services, and where those data are processed by a health professional subject under national law or rules established by national competent bodies to the obligation of professional secrecy or by another person also subject to an equivalent obligation of secrecy.	This forms an authorisation rule, "Treating Medical professionals can process or access personal data for the purposes of preventive medicine, medical diagnosis, the provision of care or treatment or the management of health-care services."
19.	Article 8.4: Subject to the provision of suitable safeguards, Member States may, for reasons of substantial public interest, lay down exemptions in addition to those laid down in paragraph 2 either by national law or by decision of the supervisory authority.	This forms an authorisation rule, "Personal data can be processed when there is a data access mandate". In order to satisfy substantial public interest a data access mandate can be provided following the appropriate legal procedures which requires human interactions.
20.	Article 8.5: Processing of data relating to offences, criminal convictions or security measures may be carried out only under the control of official authority, or if suitable specific safeguards are provided under national law, subject to derogations which may be granted by the Member State under national provisions providing suitable specific safeguards. However, a complete register of criminal convictions may be kept only under the control of official authority.	This forms an authorisation rule, "Role X (e.g. Police Officer) can read the register of criminal convictions." The controller will need to put the exact value of X depending on the application and the national implementation of the EU DPD.
21.	Article 8.6: Derogations from Paragraph 1 (Article 8.1) provided for in paragraphs 4(Article 8.4) and 5 (Article 8.5) shall be notified to the Commission.	This can't form an access control / authorisation rule. This is a guideline.
22.	Article 8.7: Member States shall determine the conditions under which a national identification number or any other identifier of general application may be processed.	This is a guideline and does not specifically say for what conditions those personal data can be processed.
23.	Article 9: Member States shall provide for exemptions or derogations from the provisions of this Chapter, Chapter IV and Chapter VI for the processing of personal data carried out solely for journalistic purposes or the purpose of artistic or literary expression only if they are necessary to reconcile the right to privacy with the rules governing freedom of expression.	This rule requires examination of national implementations of the EU DPD and the balance of interests is too critical to form an authorisation rule.
24.	Article 10: Information in cases of collection of data from the data subject. Member States shall provide that the controller or his representative must provide a data	The rules in this article can't form separate access control/ authorisation rules as they don't say based on what conditions personal data can be accessed

	subject from whom data relating to himself are collected with at least the following information, except where he already has it: (a) the identity of the controller and of his representative, if any; (b) the purposes of the processing for which the data are intended; (c) any further information such as - the recipients or categories of recipients of the data, - whether replies to the questions are obligatory or voluntary, as well as the possible consequences of failure to reply, - the existence of the right of access to and the right to rectify the data concerning him in so far as such further information is necessary, having regard to the specific circumstances in which the data are collected, to guarantee fair processing in respect of the data subject.	or processed. These rules specify what information to be given to the data subject when the data have been obtained from the data subject. A solution to the enforcement of this legal requirement is to have an obligation policy inside an Obligation PDP to notify the data subject when the personal data are stored or accessed. This Obligation PDP will be called by the Master PDP so that the obligation is always executed.
25.	Article 11 Information where the data have not been obtained from the data subject 1. Where the data have not been obtained from the data subject, Member States shall provide that the controller or his representative must at the time of undertaking the recording of personal data or if a disclosure to a third party is envisaged, no later than the time when the data are first disclosed provide the data subject with at least the following information, except where he already has it: (a) the identity of the controller and of his representative, if any; (b) the purposes of the processing; (c) any further information such as - the categories of data concerned, - the recipients or categories of recipients, - the existence of the right of access to and the right to rectify the data concerning him in so far as such further information is necessary, having regard to the specific circumstances in which the data are processed, to guarantee fair processing in respect of the data subject.	The rules in this article can't form separate access control/ authorisation rules as they don't say based on what condition personal data can be accessed or processed. These rules specify what information is to be given to the data subject when the data have not been obtained from the data subject. A solution to this is to use an Obligation PDP which will be called by the Master PDP after it gets a grant decision from the Access Control PDPs and the Obligation PDP will add the obligations to notify the data subject.
26.	Article 11.2: Paragraph 1(Article 11.1) shall not apply where, in particular for processing for statistical purposes or for the purposes of historical or scientific	The 1st part of the rule can add conditions to the rules obtained from the article 11.1 to specify when the purpose of processing is not a statistical

	research, the provision of such information proves impossible or would involve a disproportionate effort or if recording or disclosure is expressly laid down by law. In these cases Member States shall provide appropriate safeguards.	purpose or historical or scientific research purpose the obligation to notify the data subject applies. However, the other conditions mentioned in this rule require human judgement and so those can't be converted into deterministic conditions.
27.	Article 12: Right of access Member States shall guarantee every data subject the right to obtain from the controller: (a) without constraint at reasonable intervals and without excessive delay or expense: - confirmation as to whether or not data relating to him are being processed and information at least as to the purposes of the processing, the categories of data concerned, and the recipients or categories of recipients to whom the data are disclosed, - communication to him in an intelligible form of the data undergoing processing and of any available information as to their source, - knowledge of the logic involved in any automatic processing of data concerning him at least in the case of the automated decisions referred to in Article 15 (1)	The authorisation rule, "the data subject can Read his/her personal data", would ensure the access right of the data subject. For the other conditions the solution is to have an obligation PDP called by the Master PDP. The Obligation PDP provides in its policy the obligation to notify the data subject when the data are processed. This obligation will ensure that if his/her personal data are processed the data subject gets notification. However, it requires human judgement to determine an intelligible form to inform the data subject or to determine the knowledge of any logic involved in any automatic processing. Hence, it is not possible to have an access control rule in the legal PDP for ensuring those.
28.	Article 12. (b) as appropriate the rectification, erasure or blocking of data the processing of which does not comply with the provisions of this Directive, in particular because of the incomplete or inaccurate nature of the data	It requires human judgement to determine whether the processing does not comply with the provisions of the directive, or whether the data are incomplete or irregular. The authorisation policy "A data subject can object to processing with an obligation to log the request" will ensure that if a data subject finds that his/her personal data are not processed lawfully can object to that. When a data subject objects to a processing a log is created with that information and the controller will determine how to handle the objection depending on the cases.
29.	Article 12.(c) notification to third parties to whom the data have been disclosed of any rectification, erasure or blocking carried out in compliance with (b), unless this proves impossible or involves a disproportionate effort.	This can't form an access control rule rather an update mechanism would be appropriate to satisfy the rule.
30.	Article 13 Exemptions and restrictions 1. Member States may adopt legislative measures to restrict the scope of the obligations and rights provided for in Articles 6 (1), 10, 11 (1), 12 and 21 when such a restriction constitutes a necessary measures to safeguard: (a) national security;	This article mentions when the exemption to data access right of the data subject is applied and when the notification should not be provided to data subject for accessing his/her personal data. Hence this article leads to the construction of the following authorisation rules, "Data subject is denied to access personal data if there is a national security issue. Data subject is denied to access personal data if there is a legal objection.

	(b) defence; (c) public security; (d) the prevention, investigation, detection and prosecution of criminal offences, or of breaches of ethics for regulated professions; (e) an important economic or financial interest of a Member State or of the European Union, including monetary, budgetary and taxation matters; (f) a monitoring, inspection or regulatory function connected, even occasionally, with the exercise of official authority in cases referred to in (c), (d) and (e); (g) the protection of the data subject or of the rights and freedoms of others.	Data subject is denied to access personal data if there is an important economic or financial issue. The data subject is denied access to his/her medical record if there is a medical objection provided by a Medical Professional in cases s/he thinks that seeing the record might cause serious harm to the physical or mental health condition of the data subject." Furthermore, based on the national laws there needs to be rules in the Legal PDP to say who are authorised to set a national security issue, a legal objection, an important economic or financial issue or a medical objection. Hence a rule is formed saying, "Authority X (e.g. Medical Professional) can issue Y (e.g. Medical Objection) to Z type of data (e.g. Medical Data) for the purpose P (e.g. protecting harm to the data subject)."
31.	Article 13. 2.Subject to adequate legal safeguards, in particular that the data are not used for taking measures or decisions regarding any particular individual, Member States may, where there is clearly no risk of breaching the privacy of the data subject, restrict by a legislative measure the rights provided for in Article 12 when data are processed solely for purposes of scientific research or are kept in personal form for a period which does not exceed the period necessary for the sole purpose of creating statistics.	This rule is limiting a data subject's access right. However, it requires examination of national laws and also requires human judgement to determine whether the data are being used for taking measure or decisions regarding any particular individual or not. Hence, it does not form any authorisation rule.
32.	Article 14 : The data subject's right to object Member States shall grant the data subject the right: (a) at least in the cases referred to in Article 7 (e) and (f), to object at any time on compelling legitimate grounds relating to his particular situation to the processing of data relating to him, save where otherwise provided by national legislation. Where there is a justified objection, the processing instigated by the controller may no longer involve those data; (b) to object, on request and free of charge, to the processing of personal data relating to him which the controller anticipates being processed for the purposes of direct marketing, or to be informed before personal data are disclosed for the first time to third parties or used on their behalf for the purposes of direct marketing, and to be expressly offered the right to object free of charge to such disclosures or uses.	This forms an authorisation rule, "A data subject can send Object to Processing with an obligation to log the request". However, when the "Object To Processing" is sent by a data subject to the system the objection is placed in a log. The controller checks the log and determines (by human judgment) if the objection will be granted or not.

33.	Article 15 :Automated individual decisions 1. Member States shall grant the right to every person not to be subject to a decision which produces legal effects concerning him or significantly affects him and which is based solely on automated processing of data intended to evaluate certain personal aspects relating to him, such as his performance at work, creditworthiness, reliability, conduct, etc.	This rule is talking about the right of a data subject not to be a subject of a decision but does not talk about any authorisation, i.e. it does not say based on what condition an action on a personal data item can be performed. Hence this Legal rule does not form an access control rule.
34.	Article 15.2: Subject to the other Articles of this Directive, Member States shall provide that a person may be subjected to a decision of the kind referred to in paragraph 1 if that decision: (a) is taken in the course of the entering into or performance of a contract, provided the request for the entering into or the performance of the contract, lodged by the data subject, has been satisfied or that there are suitable measures to safeguard his legitimate interests, such as arrangements allowing him to put his point of view; or	Similar to the previous rule this rule is talking about the right of a data subject not to be a subject of a decision but does not talk about any authorisation, i.e. it does not say based on what condition an action on a personal data item can be performed. Hence this Legal rule does not form an access control rule.
	(b) is authorised by a law which also lays down measures to safeguard the data subject's legitimate interests.	This requires to examine other laws and can't form an access control rule
35.	Article 17. 1. Member States shall provide that the controller must implement appropriate technical and organisational measures to protect personal data against accidental or unlawful destruction or accidental loss, alteration, unauthorised disclosure or access, in particular where the processing involves the transmission of data over a network, and against all other unlawful forms of processing. Having regard to the state of the art and the cost of their implementation, such measures shall ensure a level of security appropriate to the risks represented by the processing and the nature of the data to be protected.	This rule is related to access control but it is only a guideline / instruction and can't form an access control rule.

36.	Article 17.2: The Member States shall provide that the controller must, where processing is carried out on his behalf, choose a processor providing sufficient guarantees in respect of the technical security measures and organisational measures governing the processing to be carried out, and must ensure compliance with those measures.	This rule is related to access control but it is only a guideline / instruction and can't form an access control rule.
37.	Article 17.3: The carrying out of processing by way of a processor must be governed by a contract or legal act binding the processor to the controller and stipulating in particular that: - the processor shall act only on instructions from the controller, - the obligations set out in paragraph 1, as defined by the law of the Member State in which the processor is established, shall also be incumbent on the processor.	This rule is related to access control but it is only a guideline / instruction and can't form an access control rule.
38.	Article 25.1: The Member States shall provide that the transfer to a third country of personal data which are undergoing processing or are intended for processing after transfer may take place only if, without prejudice to compliance with the national provisions adopted pursuant to the other provisions of this Directive, the third country in question ensures an adequate level of protection.	This can't form an access control rule to allow the transfer of personal data only because of the request is coming from a country that has an adequate level of protection. Many other factors may be involved in forming a decision on whether the transfer should be allowed or not. So this does not form an authorisation rule.

39.	Article 25.2: The adequacy of the level of protection afforded by a third country shall be assessed in the light of all the circumstances surrounding a data transfer operation or set of data transfer operations; particular consideration shall be given to the nature of the data, the purpose and duration of the proposed processing operation or operations, the country of origin and country of final destination, the rules of law, both general and sectoral, in force in the third country in question and the professional rules and security measures which are complied with in that country.	This is an instruction for determining whether there is an adequate level of protection. This does not form an authorisation rule.
40.	Article 25.3: The Member States and the Commission shall inform each other of cases where they consider that a third country does not ensure an adequate level of protection within the meaning of paragraph 2.	This is an instruction only and does not form an access control rule.
41.	Article 25.4: Where the Commission finds, under the procedure provided for in Article 31 (2), that a third country does not ensure an adequate level of protection within the meaning of paragraph 2 of this Article, Member States shall take the measures necessary to prevent any transfer of data of the same type to the third country in question.	This forms an authorisation rule, "Deny a transfer of personal data when the transfer is requested to one of the countries that does not have an adequate level of protection[13]."
42.	Article 25.5: At the appropriate time, the Commission shall enter into negotiations with a view to remedying the situation resulting from the finding made pursuant to paragraph 4.	This is an instruction only and does not form an access control rule.

[13] The third countries that have an adequate level of protection can be found in http://ec.europa.eu/justice/policies/privacy/thridcountries/index_en.htm

43.	Article 25.6: The Commission may find, in accordance with the procedure referred to in Article 31 (2), that a third country ensures an adequate level of protection within the meaning of paragraph 2 of this Article, by reason of its domestic law or of the international commitments it has entered into, particularly upon conclusion of the negotiations referred to in paragraph 5, for the protection of the private lives and basic freedoms and rights of individuals. Member States shall take the measures necessary to comply with the Commission's decision.	This is also an instruction to find out whether the 3rd country has an adequate level of protection.
44.	Article 26.1: By way of derogation from Article 25 and save where otherwise provided by domestic law governing particular cases, Member States shall provide that a transfer or a set of transfers of personal data to a third country which does not ensure an adequate level of protection within the meaning of Article 25 (2) may take place on condition that: (a) the data subject has given his consent unambiguously to the proposed transfer; or	This forms an authorisation rule, "Personal data can be transferred to any country if the data subject has given consent to the transfer. "
45.	Article 26.1.(b) the transfer is necessary for the performance of a contract between the data subject and the controller or the implementation of pre contractual measures taken in response to the data subject's request; or	The first part forms an authorisation rule, "When the purpose is performance of a contract and the data subject and the controller are the parties of the contract and the requester is mentioned as an authorised requester then grant the transfer of the personal data mentioned in the contract." The second part of the Legal rule makes an authorisation rule "When the purpose is implementation of the pre contractual measures and the data subject requested to process the resource then grant the access".
46.	Article 26.1.(c) the transfer is necessary for the conclusion or performance of a contract concluded in the interest of the data subject between the controller and a third party; or	This forms an authorisation rule, "When the purpose is performance of a contract / conclusion of a contract and a party of the contract is the controller and the subject of the contract is the data subject and the requester is mentioned as an

		authorised requester of the contract then grant the transfer of the personal data mentioned in the contract" (An assumption is made that the controller is a trusted party and so when the controller signs a contract with a third party it puts the identifying attributes of the data subject as a subject of the contract after making sure that the data subject will be benefitted from the contract.)
47.	Article 26.1.(d) the transfer is necessary or legally required on important public interest grounds, or for the establishment, exercise or defence of legal claims; or	This forms an authorisation rule, "When there is a data transfer mandate the transfer of the personal data is allowed." The controller can add rules depending on the data it holds and depending on the national rule to allow the transfer of personal data to satisfy an important public interest.
48.	Article 26.1.(e) the transfer is necessary in order to protect the vital interests of the data subject; or	This forms an authorisation rule, "Medical Professionals can BTG to transfer medical data."
49.	Article 26.1.(f) the transfer is made from a register which according to laws or regulations is intended to provide information to the public and which is open to consultation either by the public in general or by any person who can demonstrate legitimate interest, to the extent that the conditions laid down in law for consultation are fulfilled in the particular case.	The conditions in the rule are not feasible to be presented in an access control rule. It depends on national laws as well. Furthermore, public information is already available and hence there is no need to control access to that. Hence it does not form an access control rule.
50.	Article 26. 2: Without prejudice to paragraph 1, a Member State may authorise a transfer or a set of transfers of personal data to a third country which does not ensure an adequate level of protection within the meaning of Article 25 (2), where the controller adduces adequate safeguards with respect to the protection of the privacy and fundamental rights and freedoms of individuals and as regards the exercise of the corresponding rights; such safeguards may in particular result from appropriate contractual clauses.	This forms an authorisation rule, "Personal data can be transferred to a country not having an adequate level of protection when there is a contract between the controllers (data sender and receiver controllers) to ensure an adequate safeguard."
51.	Article 26. 3: The Member State shall inform the Commission and the other Member States of the authorisations it grants pursuant to paragraph 2. If a Member State or the Commission objects on justified grounds involving the protection of the privacy and fundamental rights and freedoms of individuals, the Commission shall take appropriate measures in accordance with the procedure laid down in Article 31 (2). Member States shall take the necessary measures to comply with the Commission's decision.	This rule is a guideline for the member states and can't form an authorisation rule.

52.	Article 28.3: Each authority shall in particular be endowed with: - investigative powers, such as powers of access to data forming the subject-matter of processing operations and powers to collect all the information necessary for the performance of its supervisory duties.	This forms the authorisation rules, "The Supervisory Authority can access and collect personal data for the performance of supervisory duties."
	- effective powers of intervention, such as, for example, that of delivering opinions before processing operations are carried out, in accordance with Article 20, and ensuring appropriate publication of such opinions, of ordering the blocking, erasure or destruction of data, of imposing a temporary or definitive ban on processing, of warning or admonishing the controller, or that of referring the matter to national parliaments or other political institutions,	This forms an authorisation rule "The Supervisory Authority can order the blocking/ erasing /destruction of data, or impose a temporary ban on the processing or impose a definitive ban on processing."
	- the power to engage in legal proceedings where the national provisions adopted pursuant to this Directive have been violated or to bring these violations to the attention of the judicial authorities. Decisions by the supervisory authority which give rise to complaints may be appealed against through the courts.	This rule does not mention any action to be performed on personal data on a particular condition and so does not form any authorisation rule.
53.	Article 28.4: Each supervisory authority shall hear claims lodged by any person, or by an association representing that person, concerning the protection of his rights and freedoms in regard to the processing of personal data. The person concerned shall be informed of the outcome of the claim. Each supervisory authority shall, in particular, hear claims for checks on the lawfulness of data processing lodged by any person when the national provisions adopted pursuant to Article 13 of this Directive apply. The person shall at any rate be informed that a check has taken place.	This rule does not mention any action to be performed on personal data on a particular condition and so does not form any authorisation rule.

Table A 1. 2: Authorisation rules in natural language

Policy No.	Articles	Legal Natural Language Policies
1.	Article 6.1 (b)	If the requested purpose of processing does not match with any of the original purposes of collection or is not for a historical purpose/statistical purpose / scientific purpose deny the request.
2.	Article 6.1 (d).	The data subject can send a data update request with an obligation to log the request.
3.	Article 6.1 (e).	If the validity time of the data is earlier than the request time i.e. the request is made after the validity time of the data has expired, then deny the request.
4.	Articles 7 (a) and 8. 2 (a)	A data subject can submit a policy / update a policy.
5.	Article 7 (b).	When the purpose is performance of a contract or entering into a contract and a party of the contract is the data subject and the requester is an authorised requester of the contract then grant access to the resource mentioned in the contract.
6.	Article 7 (b).	When the purpose is entering into a contract and the data subject requested to process the resource then grant the access.
7.	Articles 7 (c), 8.2 (e) and 8.4	Personal data can be accessed when there is a data access mandate.
8.	Articles 7 (d) and 8.2 (c)	Medical Professionals can Break the Glass to Read or Write on medical data.
9.	Article 7 (e).	Entities with a specific role X (e.g. social security authority) can access a specific resource type Y (e.g., personal data related to pensions) and if the purpose is the performance of a task of public interest (e.g. social security administration) or an exercise of the official authority.
10.	Article 8.1.	If the request is for a sensitive personal data then DCA=DenyOverrides. This rule only forms a CRR not an ACR.
11.	Article 8.2 (d).	Organisation X (e.g. Temple/Church) can access personal data of type Y (where Y is related to X) (e.g., Cast information) of the subjects who are the members of X.
12.	Article 8.3.	The treating Medical Professional can Read/Write on personal data for the purpose of preventive medicine, medical diagnosis, provision of care or treatment or the management of the health care service.
13.	Article 8.5.	Role X (e.g. policeman) can access the register of criminal record. The actual values of X needs to be configured by the controller depending on the application and depending on the national implementations of the EU DPD.

14.	Article 12	A data subject can Read his/her own personal data.
15.	Articles 12 (b), 14 (b).	A data subject can send "Object to Processing" with an obligation to log the request.
16.	Articles 13.1 (a),(b),(c)(d),(e)(g).	A data subject is denied access to his/her personal data if there is a national security issue, legal objection, important economic and financial issue or medical objection.
17.	Articles 13.1 (a),(b),(c)(d),(e)(g).	Authority X (e.g. Medical Professional) can Write Y (e.g. Medical Objection) to Z type of data (e.g. medical data) for the purpose P (e.g. protecting harm to the data subject).
18.	Article 26.1 (a).	Personal data can be transferred to a non EU country or to countries not having an adequate level of protection when the data subject has given consent to the transfer.
19.	Article 26.1.(b).	When the purpose is performance of a contract and the data subject and the controller are the parties of the contract and the requester is one of the authorised party of the contract then grant the transfer of the personal data mentioned in the contract.
20.	Article 26.1 (c).	When the purpose is performance of a contract / conclusion of a contract and a party of contract is the controller and the data subject is benefitted by the contract and the requester is mentioned as an authorised requester in the contract, then grant the transfer of the personal data mentioned in the contract.
21.	Article 26.1 (c).	When the purpose is implementation of pre contractual measures and there is a consent from the data subject requested to transfer the resource then grant the transfer.
22.	Article 26.1 (d).	Personal data can be transferred to a non EU country or to a country not having an adequate level of protection when there is a data transfer mandate.
23.	Article 26.1(e).	Medical Professionals can BTG to transfer medical data to a non EU country or to a country not having an adequate level of protection.
24.	Article 26.2.	Personal data can be transferred to a non EU country or to a country not having an adequate level of protection when there is a contract between the controllers (data sender and receiver controllers) to ensure an adequate safeguard.
25.	Article 25.4.	Personal data are denied to transfer to a non EU country or to a country not having an adequate level of protection when none of the conditions of the rules 18-25 is satisfied.
26.	Article 28 .3.	The Supervisory Authority can access and collect personal data for the performance of supervisory duties.
27.	Article 28 .3.	The Supervisory Authority can order the blocking/erasing /destruction of personal data, or impose a temporary ban on the processing or impose a definitive ban on processing.

Table A 1. 3: Controlled natural language rules

No. of rule	Controlled Natural Language Rule in ABNF
1.	ACR 1: If the Action:Purpose:string is not the Resource:PurposesOfCollection:string OR the Action:Purpose:string is not a "historical purpose" / "statistical purpose" / "scientific purpose" then Deny the Access to the PersonalData.
2.	ACR 2: If the Subject:E-mail:string is equal to the Resource:DataSubject'sE-mail:string OR the Subject:NHSNumber:string is equal to the Resource:DataSubject'sNHSNumber:string then Grant the SendDataUpdateRequests for the PersonalData with obligations to LogTheRequest.
3.	ACR 3: If the Environment:RequestTime:date is less than Resource:ValidityTime:date then Deny the Access to the PersonalData.
4.	ACR 4: If the Subject:Email:string is equal to the resource:DataSubject'sE-mail:string OR the Subject:NHSNumber:string is equal to the Resource:DataSubject'sNHSNumber:string then Grant the SubmitPolicy / UpdatePolicy for PersonalData.
5.	ACR 5: If the Action:Purpose:string is a "Performance of a contract" AND the Environment:RequestTime:date is less than the Environment:EndTimeOfContract:date AND the Environment:RequestTime:date is greater than the Environment:StartTimeOfContract:date AND (the Resource:DataSubject'sE-mail:string is equal to Environment:ContractParty'sE-mail:string OR the Resource:DataSubject'sNHSNumber:string is equal to Environment:ContractParty'sNHSNumber:string) Subject:RoleAndOrganisation:string is equal to Environment:AuthorisedRequester'sRoleAndOrganisation:string AND the Resource:ResourceType:string is equal to Environment:ContractReosurceType:string then Grant the Access.
6.	ACR 6: If the Action:Purpose:string is "entering into a contract" AND the Environment:SubjectRequestedToProcess:string is Resource:ResourceType:string AND (the Subject:RoleAndOrganisation:string is equal to Environment:AllowedParty'sRoleAndOrganisation:string OR Subject:E-mail:string is equal to Environment:AllowedParty'sE-mail:string) then Grant the Access to PersonalData.
7.	ACR 7: If there is a DataAccessManadate then Grant the Access to the PersonalData.
8.	ACR 8: If the Subject:Role:string is "MedicalProfessional" AND the Action:Purpose:string is "medical diagnosis" / "the provision of care and treatment" / "preventive medicine" then BreakTheGlass to Read / Write on MedicalData.
9.	ACR 9: If the Subject:Role:string is "Social Security Officer" AND the Action:Purpose:string is "social security administration" / "exercise of the official authority" then Grant the Access to the PersonalDataRelatedToPension.
10.	ACR 10: If the Subject:Role:string is the "AuthorityOfTemple" and the Resource:DataSubject'sNameAndAddress:string is equal to Environment:Member'sNameAndAddress:string then Grant Read to CastInformation.
11.	ACR 11: If the Subject:PhysicianID:string is equal to Resource:TreatingPhysicianID:string AND the Action:Purpose:string is "preventive medicine" / "medical diagnosis" / "provision of care and treatment" / "management of health care service" then Grant Read / Write to the MedicalData.
12.	ACR 12: If the Subject:Role:string is the "PoliceOfficer" then Grant Read / Write to RegisterOfCriminalRecord.
13.	ACR 13: If the Subject:E-mail:string is equal to Resource:DataSubject'sE-mail:string then Grant ObjectToProcessing to PersonalData with an obligation to LogTheRequest.
14.	ACR 14a: If the Subject:E-mail:string is equal to Resource:DataSubject'sE-mail:string AND there is a Environment:MedicalObjection:boolean / Environment:LegalObjection:boolean / Environment:NationalSecurityIssue:boolean / Environment:EconomicOrFinancialIssue:boolean then Deny Read to PersonalData.

	ACR 14b: If the Subject:E-mail:string is equal to Resource:DataSubject'sE-mail:string then Grant Read to PersonalData.
15.	ACR 15: If the Subject:Role:string is "MedicalProfessional" AND the Action:Purpose:string is "protecting harm to the data subject" then Grant WriteMedicalObjection to the MedicalData.
16.	ACR 16a: If the Environment:SubjectConsentsTo'sNameAndAddress:string is equal to Subject:Subject'sNameAndAddress:string then Grant the TRANSFER with an obligation to Transfer the sticky policy. ACR 16b: If the Action:Purpose:string is "performance of a contract" AND the Resource:ResourceType:string is equal to the Environment:ContractResourceType:string AND the Environment:SubjectOfContract'sNameAndAddress:string is equal to the Resource:DataSubject'sNameAndAddress:string AND (the Environment:ContractSignerOne'sNameAndAdderess:string is equal to the Resource:DataSubject'sNameAndAddress:string OR the Environment:ContractSingerTwo'sNameAndAddress:string is equal to the Resource:DataSubject'sNameAndAddress:string) AND (the Environment:ContractSignerOne'sNameAndAdderess:string is equal to the Environment:ControllerRepresentative'sNameAndAddress:string OR the Environment:ContractSingerTwo'sNameAndAddress:string is equal to the Environment:ControllerRepresentative'sNameAndAddress:string) AND Environment:AuthorisedRequester'sNameAndAddres:string is equal to Subject:NameAndAddress:string AND Environment:requestTime:date is less than Environment:EndTimeOfContract:date AND Environment:requestTime:date is greater than the Environment:StartTimeOfContract:date then Grant the TRANSFER with an obligation to TRANSFER sticky policy. ACR 16c:If the Resource:ResourceType:string is equal to Environment:ContractResourceType:string AND Environment:SubjectOfContract'sNameAndAddress:string is equal to Resource:Data Subject's NameAndAddress:string Action:Purpose:string is "performance of a contract" AND (ContractSignerOne'sNameAndAdderess is equal to ControllerRepresentative's NameAndAddress OR Environment:ContractSingerTwo's NameAndAddress:string is equal to Environment:ControllerRepresentative's NameAndAddress:string) AND AuthorisedRequester's NameAndAddres is equal to Subject:NameAndAddress:string AND Environment:requestTime:date is less than Environment:EndTimeOfContract:date AND Environment:requestTime:date is greater than Environment:StartTimeOfContract:date then Grant the Transfer the personal data with an obligation to TRANSFER sticky policy.

	ACR 16d: If the Resource:ResourceType:string is "Personal Data" AND Action:Purpose:string is "entering into a contract" AND Environment:SubjectRequesterToTransfer:string is equal to Resource:ResourceType:string AND Environment: AllowedPartyToTransferData:string is equal to Subject:NameAndAddress:string then Grant the Transfer the personal data with an obligation to TRANSFER sticky policy. ACR 16e:If the Resource:ResourceType:string is "Personal Data" AND there is a Environment:DataTransferMandate:boolean then Grant the Transfer the personal data with an obligation to TRANSFER sticky policy. ACR16f: If Subject:Role:string is MedicalProfessional AND Action:Purpose:string is "protecting harm to the data subject" then BTG to Transfer Medical Data with an obligation to TRANSFER sticky policy. ACR 16g:If the Resource:ResourceType:string is "Personal Data" AND there is a Environment:AdequateSafeguard:boolean then Grant the Transfer the personal data with an obligation to TRANSFER sticky policy. ACR 16h:f Resource:ResourceType:string is "Personal Data" AND Action:TransferToCountry:string is not equal to X then Deny the transfer. [X= the list of EU countries and the countries having adequate protection.]
17.	ACR 17: If Subject:Role:string is SupervisoryAuthority AND Action:Purpose:string is "performance of supervisory duties" then Grant the access/collect PersonalData.
18.	ACR 18: If Subject:Role:string is SupervisoryAuthority then Grant the Order to block/Order to erase /Order to destruct/ Impose temporary ban/ Impose definitive ban on PersonalData.
19.	CRR 19: If the Resource:ResourceType:string is "sensitive personal" data then DCA=DenyOverrides.

Appendix 2: Validation Test of Legal Policies

Table A 2. 1: Validation Tests of Legal Access Control and Conflict Resolution Policy

Test case No.	Request Context	Expected result	Obtained result	Comments
1.	Purpose of collection= [X], Purpose (current purpose)= [X]	DCA = N/A Decision = N/A	DCA = N/A Decision = N/A	These tests are testing the Legal rule "If the requested purpose of processing does not match with the original purpose of collection or is not for a historical purpose/statistical purpose / scientific purpose then Deny the request."
2.	Purpose of collection = [X], Purpose (current purpose)= [Y, where Y≠ X and Y≠"historical purpose" / Y≠ "statistical purpose" / Y≠"scientific purpose"]	DCA=DenyOverrides Decision=Deny	DCA=DenyOverrides Decision=Deny	
3.	Purpose of collection =[X], Purpose (current purpose)= [Y, where Y≠ X and Y=scientific purpose]	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
4.	Purpose of collection = [X], Purpose (current purpose) = [Y, where Y ≠ X and Y= historical purpose]	DCA=DenyOverrides Decision=Deny	DCA=DenyOverrides Decision=Deny	
5.	Purpose of collection= [X], Purpose (current purpose)= [Y, where Y≠ X and Y= scientific purpose]	DCA = N/A Decision = N/A	DCA = N/A Decision = N/A	
6.	The request time is less than the[validity time]	DCA = N/A Decision = N/A	DCA = N/A Decision = N/A	These tests are testing the Legal rule "If the validity time of the data is earlier than the request time then Deny the request."
7.	the request time is grater than the [validity time]	DCA=DenyOverrides Decision=Deny	DCA=DenyOverrides Decision=Deny	
8.	ResourceType=PersonalData, Requester is Data Subject (Identified by E-mail Attribute), Action-id=DataUpdateRequest	DCA=GrantOverrides,Decision=Grant with obligation to log the request	DCA=GrantOverrides,Decision=Grant with obligation to log the request	These tests are testing the Legal rule saying that "the data subject can send a data update request."A data subject is determined by a set of identifying attributes (such as name and address, e-mail address, NHS number). The identifying attributes are provided by the data subject during registration or while uploading the personal data and these are stored as resource attributes and passed to the request context. The legal PDP matches the identifying attributes of the requester with the identifying attributes obtained from the resource attributes to identify the requester as the data subject.
9.	ResourceType=PersonalData, Requester is Data Subject (Identified by E-mail Attribute), Action-id=DataUpdate	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
10.	ResourceType=PersonalData, Requester is Data Subject (Identified by Name and Address Attributes), Action-id=DataUpdateRequest	DCA=GrantOverrides Decision=Grant with obligation to log the request	DCA=GrantOverrides Decision=Grant with obligation to log the request	
11.	ResourceType=PersonalData, Requester is Data Subject (Identified by Name and Address Attributes), Action-id=DataUpdate	DCA=N/A Decision= N/A	DCA=N/A Decision= N/A	
12.	ResourceType=PersonalData, Requester is Data Subject (Identified by NHS number Attribute), Action-id=DataUpdateRequest	DCA=GrantOverrides Decision=Grant with obligation to log the request	DCA=GrantOverrides Decision=Grant with obligation to log the request	
13.	ResourceType=PersonalData, Requester is Data Subject (Identified by E-mail Attribute), Action-id=DataUpdate	DCA=N/A Decision= N/A	DCA=N/A Decision= N/A	
14.	ResourceType=PersonalData, Requester is NOT Data Subject, Action-id=DataUpdateRequest.	DCA=N/A Decision= N/A	DCA=N/A Decision= N/A	
15.	ResourceType=PersonalData, Requester is Data	DCA=GrantOverri	DCA=GrantOverri	Here testing the Legal

	Subject (Identified by E-mail Attribute), Action-id=SubmitPolicy	des Decision=Grant	des Decision=Grant	rule "the data subject can submit a policy / update a policy." Similar to the previous rule the data subject is identified by a set of attributes
16.	ResourceType=PersonalData, Requester is Data Subject (Identified by E-mail Attribute), Action-id=UpdatePolicy	DCA=GrantOverri des Decision=Grant	DCA=GrantOverri des Decision=Grant	
17.	ResourceType=PersonalData, Requester is Data Subject (Identified by Name and Address Attributes), Action-id= SubmitPolicy	DCA=GrantOverri des Decision=Grant	DCA=GrantOverri des Decision=Grant	
18.	ResourceType=PersonalData, Requester is Data Subject (Identified by Name and Address Attributes), Action-id= UpdatePolicy	DCA=GrantOverri des Decision=Grant	DCA=GrantOverri des Decision=Grant	
19.	ResourceType=PersonalData, Requester is Data Subject (Identified by NHS number Attribute), Action-id=SubmitPolicy	DCA=GrantOverri des Decision=Grant	DCA=GrantOverri des Decision=Grant	
20.	ResourceType=PersonalData, Requester is Data Subject (Identified by NHS Number Attribute), Action-id=UpdatePolicy	DCA=GrantOverri des Decision=Grant	DCA=GrantOverri des Decision=Grant	
21.	ResourceType=PersonalData, Requester is NOT Data Subject, Action-id=UpdatePolicy	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
22.	Requested ResourceType=PersonalData, ContractResourceType = PersonalData, SubjectOf Contract'sIDA = Data Subject's IDA, Purpose =processing of contract, requester's IDA= one of the AuthorisedRequester's IDA, one of party of contract = Data Subject, Action-id=Read, request time is within the contract's validity time.	DCA=GrantOverri des, Decision=Grant	DCA=GrantOverri des, Decision=Grant	Here testing the Legal rule , "When the purpose is performance of a contract and the contract is valid at the time the request is made, a party of contract is the data subject and the requester is mentioned as an AuthorisedRequester of the contract then grant access to the resource mentioned in the contract."
23.	Requested ResourceType=PersonalData, ContractResourceType = PersonalData, SubjectOf Contract'sIDA = Data Subject's IDA, Purpose =collecting data, requester's IDA=one of AuthorisedRequester's IDA, one of party of contract = Data Subject, Action-id = Read, request time is within the contract's validity time.	DCA=N/A, Decision=N/A	DCA=N/A, Decision=N/A	
24.	Requested ResourceType=PersonalData, ContractResourceType = PersonalData, SubjectOfContract'sIDA = Data subject's IDA, Purpose = processing of contract, requester's IDA≠one of AuthorisedRequester's IDA, one of party of contract = Data Subject, Action-id=Read, request time is within the contract's validity time.	DCA=N/A, Decision=N/A	DCA=N/A, Decision=N/A	
25.	Requested ResourceType=PersonalData, ContractResource = PersonalData, SubjectOf Contract'sIDA = Data subject's IDA, Purpose = processing of contract, requester's IDA=one of AuthorisedRequester's IDA, one of party of contract = Data Subject, Action-id=Read, request time is within the contract's validity time.	DCA=N/A, Decision=N/A	DCA=N/A, Decision=N/A	
26.	Requested ResourceType=PersonalData, ContractResourceType = PersonalData, SubjectOf Contract'sIDA = Data subject's IDA, Purpose = processing of contract, requester's IDA=one AuthorisedRequster's IDA, one of party of contract's= Data Subject, Action-id=Write, request time is within the contract's validity time.	DCA=N/A, Decision=N/A	DCA=N/A, Decision=N/A	
27.	Requested ResourceType=PersonalData, ContractResourceType = CV, SubjectOf	DCA=N/A, Decision=N/A	DCA=N/A, Decision=N/A	

	Contract'sIDA = Data subject's IDA, Purpose = processing of contract, requester's IDA=one of AuthorisedRequester's IDA, Data Subject ≠one of party of contract, Action-id=Read, request time is within the contract's validity time.			
28.	Requested ResourceType=PersonalData, ContractResourceType = PersonalData, SubjectOf Contract'sIDA≠ Data subject's IDA, Purpose = processing of contract, requester's IDA=one of AuthorisedRequester's IDA, one of party of contract = Data Subject, Action-id=Read, request time is within the contract's validity time.	DCA=N/A, Decision=N/A	DCA=N/A, Decision=N/A	
29.	Requested Purpose = entering into contract, SubjectRequestedToProcess= ResourceType [X], requested ResourceType = [X], AllowedPartyToProcessData= IDA of the Subject (requester), Action-id= Read,	DCA=GrantOverrides, Decision=Grant	DCA=GrantOverrides, Decision=Grant	Testing the rule "When the purpose is entering into a contract and the data subject requested to process the resource then grant the access."
30.	Requested Purpose = entering into contract, SubjectRequestedToProcess= ResourceType [X], requested ResourceType = [Y], AllowedPartyToProcessData= IDA of the Subject (requester), Action-id= Read,	DCA=N/A, Decision=N/A	DCA=N/A, Decision=N/A	
31.	Requested Purpose = entering into contract, SubjectRequestedToProcess= ResourceType [X], requested ResourceType = [Y], AllowedPartyToProcessData≠ IDA of the Subject (requester), Action-id= Read	DCA=N/A, Decision=N/A	DCA=N/A, Decision=N/A	
32.	Requested Purpose ≠ entering into contract, SubjectRequestedToProcess= ResourceType [X], requested ResourceType = [Y], AllowedPartyToProcessData= IDA of the Subject (requester), Action-id= Read,	DCA=N/A, Decision=N/A	DCA=N/A, Decision=N/A	
33.	ResourceType=PersonalDataRelatedToPension, Subject's (Requester's) Role = SocialSecurityAuthority, Purpose= social security administration, Action=Read	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	Testing the rule "Entities with a specific role (e.g. social security authority) can access a specific resource type (e.g., personal data related to pensions) and if the purpose is the performance of a task of public interest (e.g., social security administration) or an exercise of official authority."
34.	ResourceType= PersonalDataRelatedToPension, Subject's (Requester's) Role = SocialSecurityAuthority, Purpose=exercise of official authority, Action=Read	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
35.	ResourceType= PersonalDataRelatedToPension, Subject's (Requester's) Role = InsuranceAuthority Purpose= exercise of official authority Action=Read	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
36.	ResourceType= PersonalDataRelatedToPension, Subject's (Requester's) Role = SocialSecurityAuthority, Purpose=observe data, Action=Read	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
37.	ResourceType= PersonalDataRelatedToPension, Subject's (Requester's) Role = SocialSecurityAuthority, Purpose=exercise of official authority, Action=Write	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
38.	ResourceType=PersonalData, Subject's (Requester's) Role = SocialSecurityAuthority, Purpose=exercise of official authority,	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	

	Action=Read			
39.	ResourceType=PersonalData, requester = Anyone, DataAccessMandate=true, Action=Read	DCA= GrantOverrides Decision=Grant	DCA= GrantOverrides Decision=Grant	Testing the rule "Personal data can be accessed when there is a data access mandate"
40.	Requester=PoliceOfficer assigned by PolicyAuthority, ResourceType=PersonalData, Action=Read, DataAccessMandate=true	DCA= GrantOverrides Decision=Grant	DCA= GrantOverrides Decision=Grant	
41.	Requester=PoliceOfficer assigned by SecurityAuthority, ResourceType= PersonalData, Action=Read, DataAccessMandate=false	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
42.	requester=friend of DataSubject, ResourceType=PersonalData DataAccessMandate=true	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
43.	Requester=TreatingMedicalProfessional,ResourceType=MedicalData, Purpose=medical diagnosis, Action=Read	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	Testing rule "The treating Medical Professional can Read/Write medical data for the purpose of preventive medicine, medical diagnosis, provision of care or treatment or the management of the health care service."
44.	Requester=TreatingMedicalProfessional,ResourceType=MedicalData, Purpose=medical diagnosis, Action=Write	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
45.	Requester=TreatingMedicalProfessional,ResourceType=PersonalData, Purpose=medical diagnosis, Action=Read	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
46.	Requester=TreatingMedicalProfessional,ResourceType=MedicalData, Purpose=Keeping Record, Action=Read	DCA=DenyOverrides Decision=N/A	DCA=DenyOverrides Decision=N/A	
47.	Requester=MedicalProfessional,ResourceType= MedicalData, Action=Read	DCA=GrantOverrides Decision=BTG	DCA=GrantOverrides Decision=BTG	Testing rule, "Medical Professional can Break the Glass to Read or Write medical data."
48.	Requester=MedicalProfessional,ResourceType= MedicalData, Action=BTG	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
49.	Requester=MedicalProfessional,ResourceType=PersonalData, Purpose=medical diagnosis, Action=Read	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
50.	Requester=MedicalProfessional,ResourceType= MedicalData, Action=Read, BTG=true	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
51.	Requester≠MedicalProfessional,ResourceType= MedicalData, Purpose=medical diagnosis, Action=Read, BTG=true	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
52.	Requester=MedicalProfessional,ResourceType= MedicalData, Action=Write, BTG=true	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
53.	Requester=Authority of temple X, resourceType = Religion's cast information, data subject= one of member of temple X	DCA=GrantOverrides. Decision=Grant	DCA=GrantOverrides. Decision=Grant	Testing rule "Organisation X (e.g. Temple/Church) can access personal data of type Y (where Y is related to X) (e.g., Cast information) of the subjects who are members of X."
54.	Requester=Authority of temple X, resourceType = Religion's cast information, data subject≠ one of member of temple X	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
55.	Requester=PolicyOfficer , Action-id=read, ResourceType= register of criminal record,	DCA=GrantOverrides. Decision=Grant	DCA=GrantOverrides. Decision=Grant	Testing rule "Role X (e.g. police officer) can access the register of criminal record."
56.	Requester=PolicyOfficer, Action-id=read, ResourceType= register of educational record,	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	

57.	Requester=DataSubject, ResourceType= PersonalData, Action=ObjectToProcessing	DCA=GrantOverrides Decision=Grant Obligation=LogTheRequest	DCA=GrantOverrides Decision=Grant Obligation=LogTheRequest	Testing rule "The data subject can send "*Object to Processing*" with an obligation to log the request". Data subject is always identified by a set of attributes as explained earlier which is not shown again for simplicity.
58.	Requester ≠ DataSubject, ResourceType= PersonalData, Action=ObjectToProcessing	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
59.	Requester=DataSubject, ResourceType=MedicalData LegalObjection=ture	DCA=DenyOverrides Decision=Deny	DCA=DenyOverrides Decision=Deny	Testing rule, "A data subject can read his/her personal data if there is no medical objection, legal objection, no national security issue and no economic or financial issue."It is mentionable that the data subject is identified based on the identity attributes as before which is not shown again for simplicity.
60.	Requester=DataSubject, ResourceType=PersonalData LegalObjection=ture	DCA=DenyOverrides Decision=Deny	DCA=DenyOverrides Decision=Deny	
61.	Requester=DataSubject, ResourceType=PersonalData NationalSecurityIssue=ture	DCA=DenyOverrides Decision=Deny	DCA=DenyOverrides Decision=Deny	
62.	Requester=DataSubject, ResourceType=MedicalData MedicalObjection=ture	DCA=DenyOverrides Decision=Deny	DCA=DenyOverrides Decision=Deny	
63.	Requester=DataSubject, ResourceType=PersonalData Economic/FinancialIssue=ture	DCA=DenyOverrides Decision=Deny	DCA=DenyOverrides Decision=Deny	
64.	Requester=DataSubject, ResourceType=PersonalData	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
65.	Requester=DataSubject, ResourceType=MedicalData	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
66.	Requester=DataSubject,ResourceType=Personal Data, LegalObjection=false	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
67.	Requester=MedicalProfessional,ResourceType= MedicalData, Purpose=protect harm to data subject, Action=Issue MedicalObjection	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	Testing rule "Authority X (e.g. Medical Professional) can issue Y (e.g. Medical Objection) to Z type of data (e.g. Medical Data) for the purpose P (e.g. protecting harm to the data subject).
68.	Requester=MedicalProfessional,ResourceType= MedicalData, Purpose=medical diagnosis, Action= Issue MedicalObjection	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
69.	Requester=anyone, ResourceType=PersonalDataFromRegister, TransferToCountry=UK, Action=Transfer	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	Testing rule "Personal data can be transferred to a non EU country or to a country not having an adequate level of protection if a. the subject consents to the transfer OR b. the purpose is performance of a contract or entering into a contract and the parties of the contract are the data subject and the controller and
70.	Requester=anyone, ResourceType=PersonalData, TransferToCountry=UK, Action=Transfer, SubjectConsentsToTransfer =IDA of the requester	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
71.	Requester=anyone, ResourceType=PersonalData, TransferToCountry=UK, Action=Transfer SubjectConsentsToTransferTo ≠IDA of the requester	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
72.	Requester=anyone, ResourceType=PersonalData,	DCA=GrantOverrides	DCA=GrantOverrides	

	TransferToCountry=Ghana, Action=Transfer SubjectConsentsToTransferTo = IDA of the requester	Decision=Grant	Decision=Grant	the requester is an authorised requester and the contract is valid at the time the request is made OR c. the purpose is performance of a contract or conclusion of a contract, the contract is valid when the request is made, the parties of the contract are the controller and the third party and contract's beneficiary (i.e. SubjectOfContract) is the data subject and the requester an authorised requester OR d. When the purpose is implementation of pre contractual measures and the data subject requested to transfer the resource then grant the transfer. e. there is a data transfer mandate OR f. there is an adequate safeguard provided by the requester controller; otherwise g. deny the transfer."
73.	Requester=anyone, ResourceType=PersonalData, TransferToCountry=Ghana, Action=Transfer SubjectConsentsToTransferTo ≠IDA of the requester	DCA=DenyOverrides Decision=Deny	DCA=DenyOverrides Decision=Deny	
74.	TransferToCountry=Ghana, Action=Transfer,Purpose=performance of a contract, one of PartyOfContract=DataSubject, one of PartyOfContract =Controller, Requested ResourceType=PersonalData,ContractResourceType = PersonalData, SubjectOf Contract'sIDA = Data Subject's IDA, Subject's (requester's) IDA= one of AuthorisedParty's IDA, request time is within the validity time of contract	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
75.	TransferToCountry=Italy, Action=Transfer,Purpose=performance of a contract, one of PartyOfContract=DataSubject, one of PartyOfContract =Controller, Requested ResourceType=PersonalData,ContractResourceType = PersonalData, SubjectOf Contract'sIDA = Data subject's IDA, Subject's (requester's)IDA = AuthorisedRequester's IDA, request time is within the validity time of contract	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
76.	TransferToCountry=Ghana, Action=Transfer,Purpose=performance of a contract, one of PartyOfContract ≠DataSubject, one of PartyOfContract =Controller, Requested ResourceType=PersonalData,ContractResourceType = PersonalData, SubjectOf Contract'sIDA ≠ Data Subject's IDA, Subject's (requester's)IDA = AuthorisedRequester's IDA, request time is within the validity time of contract	DCA=DenyOverrides Decision=Deny	DCA=DenyOverrides Decision=Deny	
77.	TransferToCountry=Ghana, Action=Transfer,Purpose=performance of a contract, one of PartyOfContract=DataSubject, one of PartyOfContract ≠Controller, Requested ResourceType=PersonalData,ContractResourceType = PersonalData, SubjectOf Contract'sIDA = Data Subject's IDA, Subject's (requester's)IDA = AuthorisedRequester's IDA, request time is within the validity time of contract	DCA=DenyOverrides Decision=Deny	DCA=DenyOverrides Decision=Deny	
78.	TransferToCountry=UK, Action=Transfer,Purpose=performance of a contract, one of PartyOfContract=DataSubject, one of PartyOfContract ≠Controller, Requested ResourceType=PersonalData, ContractResourceType = PersonalData, SubjectOf Contract'sIDA = Data Subject's IDA, Subject's (requester's)IDA = AuthorisedRequester's IDA, request time is within the validity time of contract	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
79.	TransferToCountry=UK, Action=Transfer,Purpose=performance of a contract, one of PartyOfContract≠DataSubject, one of PartyOfContract =Controller, Requested	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	

	ResourceType=PersonalData,ContractResourceType = PersonalData, SubjectOf Contract'sIDA ≠ Data Subject's IDA, Subject's (requester's)IDA = AuthorisedRequester's IDA, request time is within the validity time of contract			
80.	TransferToCountry=UK, Action=Transfer,Purpose=collecting data, one of PartyOfContract=DataSubject, one of PartyOfContract =Controller, Requested ResourceType=PersonalData,ContractResourceType = PersonalData, SubjectOf Contract'sIDA = Data Subject's IDA, Subject's (requester's)IDA = AuthorisedRequester's IDA, request time is within the validity time of contract	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
81.	TransferToCountry=China, Action=Transfer,Purpose=collecting data, one of PartyOfContract=DataSubject, one of PartyOfContract =Controller, Requested ResourceType=PersonalData, ContractResourceType = PersonalData, SubjectOfContract'sIDA = Data subject's IDA, Subject's (requester's)IDA = AuthorisedRequester's IDA, request time is within the validity time of contract	DCA=DenyOverrides Decision=Deny	DCA=DenyOverrides Decision=Deny	
82.	TransferToCountry=UK, Action=Transfer,Purpose=performance of a contract, one of PartyOfContract Controller, Requested ResourceType=PersonalData, ContracrtResourceType = PersonalData, SubjectOf Contract'sIDA = Data subject's IDA, Subject's (requester's)IDA = AuthorisedRequester's IDA, request time is within the validity time of contract	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
83.	TransferToCountry=China, Action=Transfer,Purpose=performance of a contract, one of PartyOfContract=Controller, Requested ResourceType=PersonalData,ContractResourceType = PersonalData, SubjectOf Contract'sIDA = Data Subject's IDA, Subject's (requester's)IDA = AuthorisedRequester's IDA, request time is within the validity time of contract	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
84.	TransferToCountry=UK, Action=Transfer,Purpose=performance of a contract, one of PartyOfContract Controller, SubjectOfContract≠DataSubject,Requested ResourceType=PersonalData, ContractResourceType = PersonalData, Subject's (requester's)IDA = AuthorisedRequester's IDA, request time is within the validity time of contract	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
85.	Requester=anyone, ResourceType=PersonalData, ContractResourceType = PersonalData TransferToCountry=India, Action=Transfer, Purpose=performance of a contract, One Of PartyOfContract= Controller, SubjectOfContract≠DataSubject, Subject's (requester's)IDA = AuthorisedRequester's IDA,	DCA=DenyOverrides Decision=Deny	DCA=DenyOverrides Decision=Deny	

	request time is within the validity time of contract			
86.	TransferToCountry=UK, Action=Transfer,Purpose=performance of a contract, one of PartyOfContract≠ Controller, SubjectOfContract=DataSubject, Requested ResourceType=PersonalData,ContractResourceType = PersonalData, SubjectOf Contract'sIDA = Data Subject's IDA, Subject's (requester's)IDA = AuthorisedRequester's IDA, request time is within the validity time of contract	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
87.	TransferToCountry=India, Action=Transfer,Purpose=performance of a contract, one of PartyOfContract≠ Controller, SubjectOfContract's IDA=DataSubject's IDA,Requested ResourceType=PersonalData, ContractResourceType = PersonalData, SubjectOf Contract'sIDA = Data subject's IDA	DCA=DenyOverrides Decision=Deny	DCA=DenyOverrides Decision=Deny	
88.	TransferToCountry=China, Action=Transfer,Purpose=performance of a contract, one of PartyOfContract= Controller, Requested ResourceType=PersonalData,ContractResourceType = PersonalData, SubjectOfContract'sIDA = Data Subject's IDA, Subject's (requester's)IDA ≠ AuthorisedRequester's IDA, request time is within the validity time of contract	DCA=DenyOverrides Decision=Deny	DCA=DenyOverrides Decision=Deny	
89.	Requester=anyone, ResourceType=PersonalData, TransferToCountry=India, Action=Transfer, adequateSafeguard=true	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
90.	Requester=anyone, ResourceType=PersonalData, TransferToCountry=Germany, Action=Transfer, adequateSafeguard=true	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
91.	Requester=anyone, ResourceType=PersonalData, TransferToCountry=India, Action=Transfer, adequateSafeguard=false	DCA=DenyOverrides Decision=Deny	DCA=DenyOverrides Decision=Deny	
92.	Requester=anyone, ResourceType=PersonalData, TransferToCountry=India, Action=Transfer	DCA=DenyOverrides Decision=Deny	DCA=DenyOverrides Decision=Deny	
93.	Requester=anyone, ResourceType=PersonalData, TransferToCountry=Italy, Action=Transfer	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
94.	Requester=anyone, ResourceType=PersonalData, TransferToCountry=India, Action=Transfer, DataTransferMandate=true	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
95.	Requester=anyone, ResourceType=PersonalData, TransferToCountry=India, Action=Transfer, DataTransferMandate=false	DCA=DenyOverrides Decision=Deny	DCA=DenyOverrides Decision=Deny	
96.	Requester=MedicalProfessional,ResourceType=MedicalData, TransferToCountry= Germany, Action=Transfer	DCA=GrantOverrides Decision=BTG	DCA=GrantOverrides Decision=BTG	Testing rule "Medical Professional can BTG to transfer medical data to a non EU country or to a country not having an
97.	Requester=MedicalProfessional,ResourceType=PersonalData, TransferToCountry=India,	DCA=GrantOverrides	DCA=GrantOverrides	

	Action=Transfer	Decision=BTG	Decision=BTG	adequate level of protection."
98.	Requester≠MedicalProfessional,ResourceType= MedicalData, TransferToCountry=UK, Action=Transfer	DCA=DenyOverrides Decision=N/A	DCA=DenyOverrides Decision=N/A	
99.	Requester=MedicalProfessional,ResourceType= MedicalData, TransferToCountry=India,Action=Transfer , BTG=true	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
100.	Requester≠MedicalProfessional,ResourceType= MedicalData, ,TransferToCountry=UK, Action=Transfer, BTG=true	DCA=DenyOverrides Decision=N/A	DCA=DenyOverrides Decision=N/A	
101.	requester =SupervisoryAuthority, ResourceType=PersonalData, Purpose=Performance of supervisory Duty, Action= Read	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	Testing rule "The Supervisory Authority can access and collect personal data for the performance of supervisory duties."
102.	requester =SupervisoryAuthority, ResourceType=PersonalData, Purpose=data observation, Action= Read	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
103.	requester =SupervisoryAuthority, ResourceType=PersonalData, Purpose=Performance of supervisory Duty, Action= Collect	DCA=GrantOverrides Decision=Grant	DCA=GrantOverrides Decision=Grant	
104.	requester =SupervisoryAuthority, ResourceType=PersonalData, Purpose=Performance of supervisory Duty, Action= Modify	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
105.	requester =SupervisoryAuthority, ResourceType=PersonalData, Purpose=Performance of supervisory Duty, Action= Order To Block	DCA=GrantOverrides Decision=Grant with obligation to Log The Order.	DCA=GrantOverrides Decision=Grant with obligation to Log The Order.	Testing rule "The Supervisory Authority can order the blocking/erasing /destruction of personal data, or impose a temporary ban on the processing or impose a definitive ban on processing."
106.	requester =SupervisoryAuthority, ResourceType=PersonalData, Purpose=Performance of supervisory Duty, Action= Destroy Data	DCA=N/A Decision=N/A	DCA=N/A Decision=N/A	
107.	requester=friend of DataSubject, ResourceType=MedicalData	DCA=DenyOverrides Decision=N/A	DCA=DenyOverrides Decision=N/A	Testing rule "If the request is for accessing sensitive personal data then DCA=DenyOverrides."
108.	requester=friend of DataSubject, ResourceType=CriminalRecord ,Action=Read	DCA=DenyOverrides Decision=N/A	DCA=DenyOverrides Decision=N/A	

Appendix 3: Extracted Legal ACR in XACML

Here some examples of implementation of Legal access control rules in XACML are presented.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- New document created with EditiX at Thu Aug 25 20:14:07 BST 2011 -->
<PolicySet xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os /home/kaniz/policies/access_control-xacml-2.0-policy-schema-os.xsd" PolicySetId="lawpolicy" PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:first-applicable" xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<Target xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"></Target>
 <Policy PolicyId="LawPolicyNo1" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
            <Description>If the requested purpose of processing does not match with the original purpose of data collection or is not for a historical purpose/statistical purpose / scientific purpose OR the validity time of data is earlier than the requested time then deny the request.</Description>
             <Target></Target>
             <Rule RuleId="LawPolicyNo1Rule1" Effect="Deny">
                        <Description>If the requested purpose of processing does not match with the original purpose of data collection or is not for a historical purpose/statistical purpose / scientific purpose OR the validity time of data is earlier than the requested time then deny the request </Description>
                        <Target/>
                        <Condition>
                                    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:not">
                                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
                                                            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                                        <ResourceAttributeDesignator AttributeId="Purpose"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">historical purpose</AttributeValue>
                                                                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">statistical purpose</AttributeValue>
                                                                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">performance of contract</AttributeValue>
                                                                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">entering into contract</AttributeValue>

                                                                                    </Apply>
                                                                        </Apply>
                                                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                                                    <ResourceAttributeDesignator
AttributeId="PurposeOfDataCollection" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                                    <ActionAttributeDesignator
AttributeId="Purpose" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                        </Apply>
                                                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-less-than">
                                                                                    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-only">
                                                                                                <EnvironmentAttributeDesignator
AttributeId="currentDateTime" DataType="http://www.w3.org/2001/XMLSchema#date"/>
                                                                                    </Apply>
                                                                                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-only">
                                                                                                <ResourceAttributeDesignator
AttributeId="Validity" DataType="http://www.w3.org/2001/XMLSchema#date"/>
                                                                                    </Apply>
                                                                        </Apply>
                                                            </Apply>
```

```
                                        </Apply>
                                </Condition>
                        </Rule>
 </Policy>
<Policy PolicyId="LawPolicyNo4" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
                        <Target></Target>
                        <Rule RuleId="LawPolicyNo4Rule1" Effect="Permit">
                                <Description>If the purpose of data processing is performance of a contract and data subject is a subject of contract and party of the contract, requester is mentioned as an authorised requester in the contract and the request time is within the validity time then grant access to the resource mentioned in the contract </Description>
                                <Target/>
                                <Condition>
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                        <ResourceAttributeDesignator AttributeId="Purpose" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">performance of contract</AttributeValue>
                                                        </Apply>
                                                </Apply>
                                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                        <EnvironmentAttributeDesignator AttributeId="ContractResourceType" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                        <ResourceAttributeDesignator AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                </Apply>
                                                <!-- checking that the data subject is a subject of the contract -->
                                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
                                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                                <EnvironmentAttributeDesignator AttributeId="SubjectOfContract'sE-mail" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                <SubjectAttributeDesignator AttributeId="DataSubject'sE-mail" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                        </Apply>
                                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                                <EnvironmentAttributeDesignator AttributeId="SubjectOfContract'sNHSNumber" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                <SubjectAttributeDesignator AttributeId="DataSubject'sNHSNumber" DataType="http://www.w3.org/2001/XMLSchema#string" />

                                                        </Apply>
                                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
                                                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and"
                                                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">

                                                <EnvironmentAttributeDesignator AttributeId="SubjectOfContract'sRole" DataType="http://www.w3.org/2001/XMLSchema#string" />
```

```
            <SubjectAttributeDesignator AttributeId="DataSubject'sRole"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                    </Apply>
                                                                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">

            <EnvironmentAttributeDesignator AttributeId="SubjectOfContract'sOrganisation"
DataType="http://www.w3.org/2001/XMLSchema#string" />

            <SubjectAttributeDesignator AttributeId="DataSubject'sOrganisation"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                    </Apply>
                                                                </Apply>
                                                            </Apply>
                                                            <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">

                                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                                                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">

            <EnvironmentAttributeDesignator AttributeId="SubjectOfContract'sName"
DataType="http://www.w3.org/2001/XMLSchema#string" />

            <SubjectAttributeDesignator AttributeId="DataSubject'sName"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                    </Apply>
                                                                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">

            <EnvironmentAttributeDesignator AttributeId="SubjectOfContract'sAddress"
DataType="http://www.w3.org/2001/XMLSchema#string" />

            <SubjectAttributeDesignator AttributeId="DataSubject'sAddress"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                    </Apply>
                                                                </Apply>
                                                            </Apply>
                                                        </Apply>

                                                        <!-- checking that the data subject is one of the party of contract -->
                                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
                                                            <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
                                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                                    <EnvironmentAttributeDesignator
AttributeId="ContractPartyOne'sE-mail" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                    <ResourceAttributeDesignator
AttributeId="DataSubject'sE-mail" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                </Apply>
                                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                                    <EnvironmentAttributeDesignator
AttributeId="ContractPartyTwo'sE-mail" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                    <ResourceAttributeDesignator
AttributeId="DataSubject'sE-mail" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                </Apply>
```

```
                                                    </Apply>
                                                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
                                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                            <EnvironmentAttributeDesignator
AttributeId="ContractPartyOne'sNHSNumber" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                            <ResourceAttributeDesignator
AttributeId="DataSubject'sNHSNumber" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                        </Apply>
                                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                            <EnvironmentAttributeDesignator
AttributeId="ContractPartyTwo'sNHSNumber" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                            <ResourceAttributeDesignator
AttributeId="DataSubject'sNHSNumber" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                        </Apply>
                                                    </Apply>
                                                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
                                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                                            <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">

                <EnvironmentAttributeDesignator AttributeId="ContractPartyOne'sRole"
DataType="http://www.w3.org/2001/XMLSchema#string" />

                <ResourceAttributeDesignator AttributeId="DataSubject'sRole"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                            </Apply>
                                                            <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">

                <EnvironmentAttributeDesignator AttributeId="ContractPartyOne'sOrganisation"
DataType="http://www.w3.org/2001/XMLSchema#string" />

                <ResourceAttributeDesignator AttributeId="DataSubject'sOrganisation"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                            </Apply>
                                                        </Apply>
                                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                                            <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">

                <EnvironmentAttributeDesignator AttributeId="ContractPartyTwo'sRole"
DataType="http://www.w3.org/2001/XMLSchema#string" />

                <ResourceAttributeDesignator AttributeId="DataSubject'sName"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                            </Apply>
                                                            <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">

                <EnvironmentAttributeDesignator AttributeId="ContractPartyTwo'sOrganisation"
DataType="http://www.w3.org/2001/XMLSchema#string" />

                <ResourceAttributeDesignator AttributeId="DataSubject'sAddress"
```

```
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                </Apply>
                                                        </Apply>
                                                </Apply>

                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
                                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">

            <EnvironmentAttributeDesignator AttributeId="ContractPartyOne'sName"
DataType="http://www.w3.org/2001/XMLSchema#string" />

            <ResourceAttributeDesignator AttributeId="DataSubject'sName"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                </Apply>
                                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">

            <EnvironmentAttributeDesignator AttributeId="ContractPartyOne'sAddress"
DataType="http://www.w3.org/2001/XMLSchema#string" />

            <ResourceAttributeDesignator AttributeId="DataSubject'sAddress"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                </Apply>
                                                        </Apply>
                                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">

            <EnvironmentAttributeDesignator AttributeId="ContractPartyTwo'sName"
DataType="http://www.w3.org/2001/XMLSchema#string" />

            <ResourceAttributeDesignator AttributeId="DataSubject'sName"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                </Apply>
                                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">

            <EnvironmentAttributeDesignator AttributeId="ContractPartyTwo'sAddress"
DataType="http://www.w3.org/2001/XMLSchema#string" />

            <ResourceAttributeDesignator AttributeId="DataSubject'sAddress"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                </Apply>
                                                        </Apply>
                                                </Apply>
                                        </Apply>
                                        <!-- cheking that the requester is an Authorised Requester -->

                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                        <EnvironmentAttributeDesignator
AttributeId="AuthorisedRequester'sE-mail" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                        <SubjectAttributeDesignator AttributeId="E-
```

```
mail" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                </Apply>
                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                        <EnvironmentAttributeDesignator
AttributeId="AuthorisedRequester'sNHSNumber" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                        <SubjectAttributeDesignator
AttributeId="NHSNumber" DataType="http://www.w3.org/2001/XMLSchema#string" />

                                                </Apply>
                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
                                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:and"
                                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">

            <EnvironmentAttributeDesignator AttributeId="AuthorisedRequester'sRole"
DataType="http://www.w3.org/2001/XMLSchema#string" />

            <SubjectAttributeDesignator AttributeId="Role" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                </Apply>
                                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">

            <EnvironmentAttributeDesignator AttributeId="AuthorisedRequester'sOrganisation"
DataType="http://www.w3.org/2001/XMLSchema#string" />

            <SubjectAttributeDesignator AttributeId="Organisation"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                </Apply>
                                                        </Apply>
                                                </Apply>
                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">

                                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">

            <EnvironmentAttributeDesignator AttributeId="AuthorisedRequester'sName"
DataType="http://www.w3.org/2001/XMLSchema#string" />

            <SubjectAttributeDesignator AttributeId="Name" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                </Apply>
                                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">

            <EnvironmentAttributeDesignator AttributeId="AuthorisedRequester'sAddress"
DataType="http://www.w3.org/2001/XMLSchema#string" />

            <SubjectAttributeDesignator AttributeId="Address" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                </Apply>
                                                        </Apply>
                                                </Apply>
                                        </Apply>
                                        <!--checking the contract is still valid -->
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
```

```
                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-less-than">
                                                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-only">
                                                    <ResourceAttributeDesignator
AttributeId="StartDate" DataType="http://www.w3.org/2001/XMLSchema#date"/>
                                                    </Apply>
                                                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-only">
                                                        <EnvironmentAttributeDesignator
AttributeId="CurrentDate" DataType="http://www.w3.org/2001/XMLSchema#date"/>
                                                    </Apply>
                                                </Apply>
                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-less-than">
                                                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-only">
                                                        <EnvironmentAttributeDesignator
AttributeId="CurrentDate" DataType="http://www.w3.org/2001/XMLSchema#date"/>
                                                    </Apply>
                                                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-only">
                                                        <ResourceAttributeDesignator
AttributeId="EndDate" DataType="http://www.w3.org/2001/XMLSchema#date"/>
                                                    </Apply>
                                                </Apply>
                                            </Apply>
                                        </Apply>
                                    </Condition>
                                </Rule>
</Policy>
<Policy PolicyId="LawPolicyNo8" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-
overrides">
            <Description>The treating Medical Professional can Read/Write personal data for the purpose of preventive
medicine, medical diagnosis, provision of care or treatment or the management of health care service. </Description>
            <Target></Target>
            <Rule RuleId="LawPolicyNo8Rule1l" Effect="Permit">
                    <Description>The treating Medical Professional can Read/Write personal data for the purpose of
preventive medicine, medical diagnosis, provision of care or treatment or the management of health care service.
</Description>
                    <Target/>
                    <Condition>
                            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                    <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
equal"/>
                                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">MedicalData</AttributeValue>
                                    <ResourceAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="ResourceType"/>
                                </Apply>
                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                    <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
equal"/>
                                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">MedicalProfessional</AttributeValue>
                                    <SubjectAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="Role"/>
                                </Apply>
```

```
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-
member-of">
                                                <ResourceAttributeDesignator AttributeId="Purpose"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">medical diagnosis</AttributeValue>
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">provision of care and treatment</AttributeValue>
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">preventive medicine</AttributeValue>
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">management of health care service</AttributeValue>
                                                </Apply>
                                        </Apply>
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-
member-of">
                                                <SubjectAttributeDesignator AttributeId="PhysicianID"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                <ResourceAttributeDesignator AttributeId="TreatingPhysicianID"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                        </Apply>
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-
member-of">
                                                <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">READ</AttributeValue>
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">WRITE</AttributeValue>
                                                </Apply>
                                        </Apply>
                                </Apply>
                        </Condition>
                </Rule>
 </Policy>
<Policy PolicyId="LawPolicyNo9" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-
overrides">
                 <Target></Target>
                <Rule RuleId="LawPolicyNo9Rule1" Effect="Deny">
                <Description>Medical professionals can BTG (break the glass) to medical data for purpose of medical diagnosis/
the provision of care and treatment / preventive medicine.</Description>
                <Target/>
                <Condition>
                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                        <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">MedicalData</AttributeValue>
                                        <ResourceAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="ResourceType"/>
                                </Apply>
                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                        <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">MedicalProfessional</AttributeValue>
                                        <SubjectAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="Role"/>
```

```
                                    </Apply>
                                    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                        <ResourceAttributeDesignator AttributeId="Purpose"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">medical diagnosis</AttributeValue>
                                            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">provision of care and treatment</AttributeValue>
                                            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">preventive medicine</AttributeValue>
                                            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">management of health care service</AttributeValue>
                                        </Apply>
                                    </Apply>
                                    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function: string-at-least-one-member-of">
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">READ</AttributeValue>
                                            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">WRITE</AttributeValue>
                                        </Apply>
                                        <ActionAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
                                    </Apply>
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                        <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:boolean-equal"/>
                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#boolean">false</AttributeValue>
                                        <EnvironmentAttributeDesignator
AttributeId="BTG:(Role(S),ResourceType(R),urn:oasis:names:tc:xacml:1.0:action:action-id(A))"
DataType="http://www.w3.org/2001/XMLSchema#boolean" MustBePresent="false"/>
    </Apply>
                                </Apply>
            </Condition>
 </Rule>
            <Rule RuleId="LawPolicyNo9Rule2" Effect="Permit">
                    <Description>Medical professionals can BTG (break the glass) to medical data for purpose of medical
diagnosis/ the provision of care and treatment / preventive medicine. </Description>
                                    <Target/>
                                    <Condition>
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">MedicalData</AttributeValue>
                                                <ResourceAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="ResourceType"/>
                                            </Apply>
                                            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
                                                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">MedicalProfessional</AttributeValue>
                                                    <SubjectAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="Role"/>
                                            </Apply>
                                            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-
```

```
least-one-member-of">
                                                <ResourceAttributeDesignator AttributeId="Purpose"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">medical diagnosis</AttributeValue>
                                                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">provision of care and treatment</AttributeValue>
                                                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">preventive medicine</AttributeValue>
                                                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">management of health care service</AttributeValue>
                                                </Apply>
                                            </Apply>
                    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function: string-at-least-one-member-of">
                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">READ</AttributeValue>
                                                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">WRITE</AttributeValue>
                                                </Apply>
                                            <ActionAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
                    </Apply>
                                            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">READ</AttributeValue>
                                                <ActionAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
                                            </Apply>
                                            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:boolean-equal"/>
                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#boolean">true</AttributeValue>
                                                <EnvironmentAttributeDesignator
AttributeId="BTG:(Role(S),ResourceType(R),urn:oasis:names:tc:xacml:1.0:action:action-id(A))"
DataType="http://www.w3.org/2001/XMLSchema#boolean" MustBePresent="false"/>
                    </Apply>
                                            </Apply>
                                </Condition>
                        </Rule>
                        <Rule RuleId="LawPolicyNo9Rule3" Effect="Permit">
                                <Description>Medical professionals can BTG (break the glass) to medical data for purpose
of medical diagnosis/ the provision of care and treatment / preventive medicine. </Description>
                                <Target/>
                                <Condition>
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                                <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
equal"/>
                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">BreakTheGlass</AttributeValue>
                                                <ActionAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
```

```
                                        </Apply>
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                                <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
equal"/>
                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">MedicalData</AttributeValue>
                                                <ResourceAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="ResourceType"/>
                                        </Apply>
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-
member-of">
                                                <SubjectAttributeDesignator AttributeId="Role"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">MedicalProfessional</AttributeValue>
                                                </Apply>
                                        </Apply>
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-
member-of">
                                                <ResourceAttributeDesignator AttributeId="Purpose"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">medical diagnosis</AttributeValue>
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">provision of care and treatment</AttributeValue>
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">preventive medicine</AttributeValue>
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">management of health care service</AttributeValue>
                                                </Apply>
                                        </Apply>
                                </Apply>
                        </Condition>
                </Rule>
                <Obligations>
                        <Obligation ObligationId="BreakTheGlassObligation" FulfillOn="Permit">
                <AttributeAssignment AttributeId="BTG:(Role(S),ResourceType(R),urn:oasis:names:tc:xacml:1.0:action:action-
id(A))" DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </Obligation>
         </Obligations>
</Policy>
-----------------------------------------------------------------------------------------------
<Policy PolicyId="LawPolicyNo11A" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-
overrides">
                <Target></Target>
                <Rule RuleId="LawPolicyNo11ARule1" Effect="Permit">
                <Description>Role X (e.g. policeman) can access the register of criminal record providing role X is assigned by
the authority Y (e.g. policy authority). </Description>
                        <Target>
                        <Subjects>
                                <Subject>
                                        <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">PoliceOfficer</AttributeValue>
                <SubjectAttributeDesignator AttributeId="1.2.826.0.1.3344810.1.1.14"
DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
                        </SubjectMatch>
                                </Subject>
```

```
                    </Subjects>
                     <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>
</Policy>
-----------------------------------------------------------------------------------------------------------------
<Policy PolicyId="LawPolicy11B" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-
overrides">
            <Target></Target>
            <Rule RuleId="LawPolicy11BRule1" Effect="Permit">
 <Target>
  <Subjects>
             <Subject>
            <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:x500Name-match">
            <AttributeValue DataType="urn:oasis:names:tc:xacml:1.0:data-
type:x500Name">CN=PoliceAuthority,OU=Police,O=UKC,ST=Some-State,C=GB</AttributeValue>
     <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
DataType="urn:oasis:names:tc:xacml:1.0:data-type:x500Name"/>
    </SubjectMatch>
   </Subject>
  </Subjects>
  <Actions>
   <Action>
    <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
     <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">Assign</AttributeValue>
     <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
    </ActionMatch>
    <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">PoliceOfficer</AttributeValue>
            <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
    </ActionMatch>
   </Action>
  </Actions>
 </Target>
 </Rule>
</Policy>

-----------------------------------------------------------------------------------------------------------------
<Policy PolicyId="LawPolicyNo7" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-
overrides">
            <Description>A data subject can read his/her personal data if there is no legal objection, no national security
issue and no economic or financial issue or no Medical Objection.</Description>
            <Target></Target>
            <Rule RuleId="LawPolicyNo7Rule1" Effect="Deny">
                    <Description>IF Requester is Data Subject, Identified by {E-mail} OR {NHSNumber} OR {Name and
Address} Attribute, AND legalObjection=true, Effect= Deny </Description>
                    <Target></Target>
                    <Condition>
              <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
```

```
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
                                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-
least-one-member-of">
                                                        <SubjectAttributeDesignator AttributeId="E-mail"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                        <ResourceAttributeDesignator
AttributeId="DataSubject'sE-mail" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                </Apply>
                                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-
least-one-member-of">
                                                        <SubjectAttributeDesignator AttributeId="NHSNumber"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                        <ResourceAttributeDesignator
AttributeId="DataSubject'sNHSNumber" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                </Apply>
                                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                                <SubjectAttributeDesignator
AttributeId="Name" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                <ResourceAttributeDesignator
AttributeId="DataSubject'sName" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                        </Apply>
                                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                                <SubjectAttributeDesignator
AttributeId="Address" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                                <ResourceAttributeDesignator
AttributeId="DataSubject'sAddress" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                        </Apply>
                                                </Apply>
                                        </Apply>
                                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                                        <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:boolean-equal"/>
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#boolean">true</AttributeValue>
                                                        <ResourceAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#boolean" AttributeId="LegalObjection"/>
                                                </Apply>
                                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                                        <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">READ</AttributeValue>
                                                        <ActionAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
                                                </Apply>
                                        </Apply>
                                </Condition>
                        </Rule>

                        <Rule RuleId="LawPolicyNo7Rule2" Effect="Deny">
                        <Description>IF Requester is Data Subject, Identified by {E-mail} OR {NHSNumber} OR {Name and
Address} Attribute, AND NationalSecurityIssue=true, Effect= Deny </Description>
                        <Target></Target>
                        <Condition>
                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
```

```
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-
least-one-member-of">
                                            <SubjectAttributeDesignator AttributeId="E-mail"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                            <ResourceAttributeDesignator
AttributeId="DataSubject'sE-mail" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                        </Apply>
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-
least-one-member-of">
                                            <SubjectAttributeDesignator AttributeId="NHSNumber"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                            <ResourceAttributeDesignator
AttributeId="DataSubject'sNHSNumber" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                        </Apply>
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                            <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                <SubjectAttributeDesignator
AttributeId="Name" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                <ResourceAttributeDesignator
AttributeId="DataSubject'sName" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                            </Apply>
                                            <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                <SubjectAttributeDesignator
AttributeId="Address" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                <ResourceAttributeDesignator
AttributeId="DataSubject'sAddress" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                            </Apply>
                                        </Apply>
                                    </Apply>
                                    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                        <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:boolean-
equal"/>
                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#boolean">true</AttributeValue>
                                        <ResourceAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#boolean" AttributeId="NationalSecurityIssue"/>
                                    </Apply>
                                    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                        <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
equal"/>
                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">READ</AttributeValue>
                                        <ActionAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
                                    </Apply>
                                </Apply>
                                </Condition>
                            </Rule>
                            <Rule RuleId="LawPolicyNo7Rule3" Effect="Deny">
                                <Description>IF Requester is Data Subject, Identified by {E-mail} OR {NHSNumber} OR
{Name and Address} Attribute, AND EconomicOrFinancialIssue=true, Effect= Deny </Description>
                                <Target>
                                </Target>
                                <Condition>
                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
                                            <Apply
```

```
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                <SubjectAttributeDesignator AttributeId="E-
mail" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                <ResourceAttributeDesignator
AttributeId="DataSubject'sE-mail" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                            </Apply>
                                            <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                <SubjectAttributeDesignator
AttributeId="NHSNumber" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                <ResourceAttributeDesignator
AttributeId="DataSubject'sNHSNumber" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                            </Apply>
                                            <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                    <SubjectAttributeDesignator
AttributeId="Name" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                    <ResourceAttributeDesignator
AttributeId="DataSubject'sName" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                </Apply>
                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                    <SubjectAttributeDesignator
AttributeId="Address" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                    <ResourceAttributeDesignator
AttributeId="DataSubject'sAddress" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                </Apply>
                                            </Apply>
                                        </Apply>
                                    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                        <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:boolean-
equal"/>
                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#boolean">true</AttributeValue>
                                        <ResourceAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#boolean" AttributeId="EconomicOrFinancialIssue"/>
                                    </Apply>
                                    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                        <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
equal"/>
                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">READ</AttributeValue>
                                        <ActionAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
                                    </Apply>
                                </Apply>
                            </Condition>
                        </Rule>
                        <Rule RuleId="LawPolicyNo7Rule2" Effect="Deny">
                            <Description>IF Requester is Data Subject, Identified by {E-mail} OR {NHSNumber} OR {Name and
Address} Attribute, AND MedicalObjection=true, Effect= Deny </Description>
                            <Target>
                            </Target>
                            <Condition>
                          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-
```

```
least-one-member-of">
                                                <SubjectAttributeDesignator AttributeId="E-mail"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                <ResourceAttributeDesignator
AttributeId="DataSubject'sE-mail" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                        </Apply>
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-
least-one-member-of">
                                                <SubjectAttributeDesignator AttributeId="NHSNumber"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                <ResourceAttributeDesignator
AttributeId="DataSubject'sNHSNumber" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                        </Apply>
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                        <SubjectAttributeDesignator
AttributeId="Name" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                        <ResourceAttributeDesignator
AttributeId="DataSubject'sName" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                </Apply>
                                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-
least-one-member-of">
                                                <SubjectAttributeDesignator AttributeId="Address"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                <ResourceAttributeDesignator
AttributeId="DataSubjectAddress" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                        </Apply>
                                </Apply>
                        </Apply>
                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:boolean-equal"/>
                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#boolean">true</AttributeValue>
                                <ResourceAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#boolean" AttributeId="MedicalObjection"/>
                        </Apply>
                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">READ</AttributeValue>
                                <ActionAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
                        </Apply>
                </Apply>
        </Condition>
        </Rule>
        <Rule RuleId="LawPolicyNo7Rule5" Effect="Permit">
                <Description>IF Requester is Data Subject, Identified by {E-mail} OR {NHSNumber} OR {Name and
Address} Attribute, Effect= Permit </Description>
                <Target>
                </Target>
                <Condition>
                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-
least-one-member-of">
                                        <SubjectAttributeDesignator AttributeId="E-mail"
DataType="http://www.w3.org/2001/XMLSchema#string" />
```

```
                                                    <ResourceAttributeDesignator AttributeId="DataSubjectE-
mail" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                </Apply>
                                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-
least-one-member-of">
                                                    <SubjectAttributeDesignator AttributeId="NHSNumber"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                    <ResourceAttributeDesignator
AttributeId="DataSubjectNHSNumber" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                </Apply>
                                                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                        <SubjectAttributeDesignator
AttributeId="Name" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                        <ResourceAttributeDesignator
AttributeId="DataSubjectName" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                    </Apply>
                                                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                        <SubjectAttributeDesignator
AttributeId="Address" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                        <ResourceAttributeDesignator
AttributeId="DataSubjectAddress" DataType="http://www.w3.org/2001/XMLSchema#string" />
                                                    </Apply>
                                                </Apply>
                                            </Apply>
                                            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
                                                <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">READ</AttributeValue>
                                                <ActionAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
                                            </Apply>
                                        </Apply>
                                    </Condition>
                                </Rule>
                            </Policy>
                        </PolicySet>
```

Appendix 4: Extracted Legal ACR in PERMIS

Here some examples of implementation of Legal access control rules in PERMIS are presented.

```
<?xml version="1.0" encoding="UTF-8"?>
<X.509_PMI_RBAC_Policy OID="LawPolicy1" DenyBased="true" EnableNotApplicable="false">
<SubjectPolicy>
 <SubjectDomainSpec ID="everywhere">
  <Include LDAPDN=""/>
 </SubjectDomainSpec>
```

```
</SubjectPolicy>
<RoleHierarchyPolicy>
 <RoleSpec Type="permisRole" OID="1.2.826.0.1.3344810.1.1.14">
 </RoleSpec>
</RoleHierarchyPolicy>
<SOAPolicy>
 <SOASpec ID="anyone" LDAPDN=""/>
</SOAPolicy>
<RoleAssignmentPolicy>
 <RoleAssignment>
  <SubjectDomain ID="everywhere"/>
   <RoleList>
   </RoleList>
  <Delegate Depth="0"/>
  <SOA ID="anyone"/>
  <Validity/>
 </RoleAssignment>
 </RoleAssignmentPolicy>
 <TargetPolicy>
  <TargetDomainSpec ID="PersonalData">
   <Include URL=""/>
   <ObjectClass Name="PersonalData"/>
  </TargetDomainSpec>
 </TargetPolicy>
 <ActionPolicy>
  <Action Name="READ" ID="READ">
   <TargetDomain ID="PersonalData"/>
  </Action>
 </ActionPolicy>
 <TargetAccessPolicy>

<!-- If the requested purpose of processing does not match with the original purpose of data collection or is not for a
historical purpose/statistical purpose / scientific purpose OR the validity time of data is earlier than the requested time then
deny the request
    -->
 <TargetAccess>
  <RoleList>
  </RoleList>
  <TargetList>
   <Target>
    <TargetDomain ID="PersonalData"/>
    <DeniedAction ID="READ"/>
   </Target>
  </TargetList>
  <IF>
  <NOT>
              <OR>
             <EQ>
                          <Environment Parameter="Purpose" Type="String" />
                          <Constant Value="historical purpose" Type="String" />
                          </EQ>
                          <EQ>
                          <Environment Parameter="Purpose" Type="String" />
                          <Constant Value="statistical purpose" Type="String" />
                          </EQ>
                          <EQ>
                          <Environment Parameter="Purpose" Type="String" />
                          <Constant Value="performance of contract" Type="String" />
                          </EQ>
```

```
                    <EQ>
                    <Environment Parameter="Purpose" Type="String" />
                    <Constant Value="entering into contract" Type="String" />
                    </EQ>
                    <EQ>
                    <Environment Parameter="Purpose" Type="String" />
                    <Environment Parameter="PurposeOfDataCollection" Type="String" />
                    </EQ>
                    <GT>
                    <Environment Parameter="ValidityTime" Type="Time" />
  <Environment Parameter="CurrentTime" Type="Time" />
                    </GT>
            </OR>
            </NOT>
 </IF>
 </TargetAccess>
</TargetAccessPolicy>
</X.509_PMI_RBAC_Policy>

-----------------------------------------------------------------------

<?xml version="1.0" encoding="UTF-8"?>
<X.509_PMI_RBAC_Policy OID="LawPolicy4" EnableNotApplicable="true">
 <SubjectPolicy>
  <SubjectDomainSpec ID="everywhere">
   <Include LDAPDN=""/>
  </SubjectDomainSpec>
 </SubjectPolicy>
 <RoleHierarchyPolicy>
  <RoleSpec Type="permisRole" OID="1.2.826.0.1.3344810.1.1.14">
  </RoleSpec>
 </RoleHierarchyPolicy>
 <SOAPolicy>
  <SOASpec ID="anyone" LDAPDN=""/>
 </SOAPolicy>
 <RoleAssignmentPolicy>
  <RoleAssignment>
   <SubjectDomain ID="everywhere"/>
   <RoleList>
   </RoleList>
   <Delegate Depth="0"/>
   <SOA ID="anyone"/>
   <Validity/>
  </RoleAssignment>
 </RoleAssignmentPolicy>
<TargetPolicy>
 <TargetDomainSpec ID="PersonalData">
  <Include URL=""/>
  <ObjectClass Name="PersonalData"/>
 </TargetDomainSpec>
</TargetPolicy>
 <ActionPolicy>
  <Action Name="READ" ID="READ">
   <TargetDomain ID="PersonalData"/>
  </Action>
 </ActionPolicy>
 <TargetAccessPolicy>

  <!-- If the purpose of data processing is performance of contract,
```

```
  data subject is a subject or contract and party of the contract, requester is mentioned
  as a authorised requester in the contract and the request time is within
  the validity time then grant the access. -->

<TargetAccess>
   <RoleList>
   </RoleList>
   <TargetList>
    <Target>
     <TargetDomain ID="PersonalData"/>
     <AllowedAction ID="READ"/>
    </Target>
   </TargetList>
   <IF>
            <AND>
             <EQ>
 <Environment Parameter="Purpose" Type="String" />
             <Constant Value="Performance of a contract" Type="String" />
 </EQ>
 <!-- checking that the data subject is one of the parties -->
             <OR>
       <EQ>
             <Environment Parameter="ContractPartyOne'sE-mail" Type="String" />
             <Environment Parameter="DataSubjectE-mail" Type="String" />
             </EQ>
             <EQ>
             <Environment Parameter="ContractPartyOne'sNHSNumber" Type="String" />
             <Environment Parameter="DataSubjectNHSNumber" Type="String" />
             </EQ>
             <AND>
              <EQ>
       <Environment Parameter="ContractPartyOne'sName" Type="String" />
             <Environment Parameter="DataSubjectName" Type="String" />
              </EQ>
              <EQ>
              <Environment Parameter="ContractPartyOne'sAddress" Type="String" />
              <Environment Parameter="DataSubjectAddress" Type="String" />
              </EQ>
            </AND>
     <AND>
                <EQ>
                     <Environment Parameter="ContractPartyOne'sRole" Type="String" />
                     <Environment Parameter="DataSubjectRole" Type="String" />
                </EQ>
                     <EQ>
                     <Environment Parameter="ContractPartyOne'sOrganisation" Type="String" />
                     <Environment Parameter="DataSubjectAddress" Type="String" />
                     </EQ>
                      </AND>
                      <EQ>
                      <Environment Parameter="ContractPartyTwo'sE-mail" Type="String" />
                      <Environment Parameter="DataSubject'sE-mail" Type="String" />
                      </EQ>
                      <EQ>
                      <Environment Parameter="ContractPartyTwo'sNHSNumber" Type="String" />
                      <Environment Parameter="DataSubject'sNHSNumber" Type="String" />
                      </EQ>
                      <AND>
                 <EQ>
```

```
                    <Environment Parameter="ContractPartyTwo'sName" Type="String" />
                    <Environment Parameter="DataSubject'sName" Type="String" />
                </EQ>
                <EQ>
                    <Environment Parameter="ContractPartyTwo'sAddress" Type="String" />
                    <Environment Parameter="DataSubject'sAddress" Type="String" />
                </EQ>
            </AND>
            <AND>
                <EQ>
                    <Environment Parameter="ContractPartyTwo'sRole" Type="String" />
                    <Environment Parameter="DataSubject'sRole" Type="String" />
                </EQ>
                <EQ>
                    <Environment Parameter="ContractPartyTwo'sOrganisation" Type="String" />
                    <Environment Parameter="DataSubject'sOrganisation" Type="String" />
                </EQ>
            </AND>
        </OR>
        <!-- checking that the data subject is a subject of the contract -->
    <OR>
            <AND>
             <EQ>
             <Environment Parameter="DataSubject'sRole" Type="String" />
             <Environment Parameter="SubjectOfContract'sRole" Type="String" />
             </EQ>
             <EQ>
             <Environment Parameter="DataSubject'sOrganisation" Type="String" />
             <Environment Parameter="SubjectOfContract'sOrganisation" Type="String" />
             </EQ>
                    </AND>
                    <AND>
             <EQ>
             <Environment Parameter="DataSubject'sName" Type="String" />
             <Environment Parameter="SubjectOfContract'sName" Type="String" />
             </EQ>
             <EQ>
             <Environment Parameter="DataSubject'sAddress" Type="String" />
             <Environment Parameter="SubjectOfContract'sAddress" Type="String" />
             </EQ>
                    </AND>
            <EQ>
            <Environment Parameter="DataSubject'sE-mail" Type="String" />
            <Environment Parameter="SubjectOfContract'sE-mail" Type="String" />
            </EQ>
            <EQ>
            <Environment Parameter="DataSubject'sNHSNumber" Type="String" />
            <Environment Parameter="SubjectOfContract'sNHSNumber" Type="String" />
            </EQ>
</OR>
<!-- checking that the requester is an authorised requester -->
<OR>
            <AND>
             <EQ>
             <Environment Parameter="Subject'sName" Type="String" />
             <Environment Parameter="AuthorisedRequester'sName" Type="String" />
             </EQ>
             <EQ>
             <Environment Parameter="Subject'sAddress" Type="String" />
```

```
            <Environment Parameter="AuthorisedRequester'sAddress" Type="String" />
            </EQ>
          </AND>
          <AND>
            <EQ>
            <Environment Parameter="Subject'sRole" Type="String" />
            <Environment Parameter="AuthorisedRequester'sRole" Type="String" />
            </EQ>
            <EQ>
            <Environment Parameter="Subject'sOrganisation" Type="String" />
            <Environment Parameter="AuthorisedRequester'sOrganisation" Type="String" />
            </EQ>
          </AND>
           <EQ>
           <Environment Parameter="Subject'sE-mail" Type="String" />
           <Environment Parameter="AuthorisedRequester'sE-mail" Type="String" />
           </EQ>
           <EQ>
           <Environment Parameter="Subject'sNHSNumber" Type="String" />
           <Environment Parameter="AuthorisedRequester'sNHSNumber" Type="String" />
           </EQ>
  </OR>
           <GT>
           <Environment Parameter="EndTime" Type="Time" />
      <Environment Parameter="CurrentTime" Type="Time" />
           </GT>
      <LT>
            <Environment Parameter="StartTime" Type="Time" />
       <Environment Parameter="CurrentTime" Type="Time" />
           </LT>
           <EQ>
           <Environment Parameter="ResourceType" Type="String" />
           <Environment Parameter="ContractResourceType" Type="String" />
           </EQ>
  </AND>
 </IF>
</TargetAccess>
</TargetAccessPolicy>
</X.509_PMI_RBAC_Policy>

-----------------------------------------------------------------------------------------------------------------------
<!--this rule to to deny access, another white list policy is used to allow access when the requester is a data subject and no other denying condition is true"// this line needs to be deleted to run the policy//-->

<?xml version="1.0" encoding="UTF-8"?>
<X.509_PMI_RBAC_Policy OID="LawPolicy7" DenyBased="true" EnableNotApplicable="true">
 <SubjectPolicy>
  <SubjectDomainSpec ID="everywhere">
   <Include LDAPDN=""/>
  </SubjectDomainSpec>
 </SubjectPolicy>
 <RoleHierarchyPolicy>
  <RoleSpec Type="permisRole" OID="1.2.826.0.1.3344810.1.1.14">
  </RoleSpec>
 </RoleHierarchyPolicy>
 <SOAPolicy>
  <SOASpec ID="anyone" LDAPDN=""/>
 </SOAPolicy>
 <RoleAssignmentPolicy>
```

```
  <RoleAssignment>
   <SubjectDomain ID="everywhere"/>
   <RoleList>
   </RoleList>
   <Delegate Depth="0"/>
   <SOA ID="anyone"/>
   <Validity/>
  </RoleAssignment>
 </RoleAssignmentPolicy>
<TargetPolicy>
 <TargetDomainSpec ID="PersonalData">
  <Include URL=""/>
  <ObjectClass Name="PersonalData"/>
 </TargetDomainSpec>
</TargetPolicy>
 <ActionPolicy>
  <Action Name="READ" ID="READ">
   <TargetDomain ID="PersonalData"/>
  </Action>
 </ActionPolicy>
 <TargetAccessPolicy>

<!-- Data Subject can READ his/her personal data if there is no legal Objection -->

            <TargetAccess>
 <RoleList>
 </RoleList>
  <TargetList>
   <Target>
    <TargetDomain ID="PersonalData"/>
    <DeniedAction ID="READ"/>
   </Target>
  </TargetList>
  <IF>
            <AND>
            <EQ>
             <Environment Parameter="LegalObjection" Type="Boolean" />
             <Constant Value="True" Type="Boolean" />
             </EQ>
               <OR>
                        <EQ>
                        <Environment Parameter="E-mail" Type="String" />
                        <Environment Parameter="DataSubjectE-mail" Type="String" />
                        </EQ>
                        <EQ>
                        <Environment Parameter="NHSNumber" Type="String" />
                        <Environment Parameter="DataSubjectNHSNumber" Type="String" />
                        </EQ>
            <AND>
                           <EQ>
                           <Environment Parameter="Name" Type="String" />
                           <Environment Parameter="DataSubjectName" Type="String" />
                           </EQ>
                           <EQ>
                           <Environment Parameter="Address" Type="String" />
                           <Environment Parameter="DataSubjectAddress" Type="String" />
                           </EQ>
                 </AND>
              </OR>
```

```
      </AND>
 </IF>
 </TargetAccess>
<!-- Data Subject can READ his/her personal data if there is no national security issue -->

<TargetAccess>
 <RoleList>
 </RoleList>
  <TargetList>
   <Target>
    <TargetDomain ID="PersonalData"/>
    <DeniedAction ID="READ"/>
   </Target>
 </TargetList>
 <IF>
              <AND>
               <EQ>
               <Environment Parameter="NationalSecurityIssue" Type="Boolean" />
               <Constant Value="True" Type="Boolean" />
               </EQ>
            <OR>
                        <EQ>
                        <Environment Parameter="E-mail" Type="String" />
                        <Environment Parameter="DataSubjectE-mail" Type="String" />
                        </EQ>
                        <EQ>
                        <Environment Parameter="NHSNumber" Type="String" />
                        <Environment Parameter="DataSubjectNHSNumber" Type="String" />
                        </EQ>
                        <AND>
                     <EQ>
                     <Environment Parameter="Name" Type="String" />
                           <Environment Parameter="DataSubjectName" Type="String" />
                           </EQ>
                           <EQ>
                           <Environment Parameter="Address" Type="String" />
                           <Environment Parameter="DataSubjectAddress" Type="String" />
                          </EQ>
                        </AND>
                </OR>
               </AND>
              </IF>
</TargetAccess>
<!-- Data Subject can READ his/her personal data if there is no economic or financial issue -->
<TargetAccess>
 <RoleList>
 </RoleList>
  <TargetList>
   <Target>
    <TargetDomain ID="PersonalData"/>
    <DeniedAction ID="READ"/>
   </Target>
  </TargetList>
  <IF>
   <AND>
            <EQ>
            <Environment Parameter="EconomicOrFinancialIssue" Type="Boolean" />
            <Constant Value="True" Type="Boolean" />
             </EQ>
```

```
            <OR><EQ>
                    <Environment Parameter="E-mail" Type="String" />
                    <Environment Parameter="DataSubjectE-mail" Type="String" />
                    </EQ>
                    <EQ>
                    <Environment Parameter="NHSNumber" Type="String" />
                    <Environment Parameter="DataSubjectNHSNumber" Type="String" />
                    </EQ>
                    <AND>
                      <EQ>
                      <Environment Parameter="Name" Type="String" />
                      <Environment Parameter="DataSubjectName" Type="String" />
                      </EQ>
                      <EQ>
                      <Environment Parameter="Address" Type="String" />
                      <Environment Parameter="DataSubjectAddress" Type="String" />
                      </EQ>
                    </AND>
             </OR>
     </AND>
 </IF>
</TargetAccess>
<!-- Data Subject can READ his/her personal data if there is no medical Objection -->
<TargetAccess>
 <RoleList>
 </RoleList>
 <TargetList>
  <Target>
   <TargetDomain ID="PersonalData"/>
   <DeniedAction ID="READ"/>
  </Target>
 </TargetList>
 <IF>
  <AND>
          <EQ>
          <Environment Parameter="MedicalObjection" Type="Boolean" />
          <Constant Value="True" Type="Boolean" />
          </EQ>
          <OR><EQ>
            <Environment Parameter="E-mail" Type="String" />
            <Environment Parameter="DataSubjectE-mail" Type="String" />
       </EQ>
            <EQ>
            <Environment Parameter="NHSNumber" Type="String" />
            <Environment Parameter="DataSubjectNHSNumber" Type="String" />
            </EQ>
           <AND>
            <EQ>
            <Environment Parameter="Name" Type="String" />
            <Environment Parameter="DataSubjectName" Type="String" />
            </EQ>
            <EQ>
            <Environment Parameter="Address" Type="String" />
            <Environment Parameter="DataSubjectAddress" Type="String" />
            </EQ>
           </AND>
          </OR>
  </AND>
 </IF>
```

```
 </TargetAccess>
</TargetAccessPolicy>
</X.509_PMI_RBAC_Policy>

<!-- this policy is used as a white list policy to allow the data subject read his/her policy if no other denying condition is true-->

<?xml version="1.0" encoding="UTF-8"?>
<X.509_PMI_RBAC_Policy OID="LawPolicy1" DenyBased="true" EnableNotApplicable="false">
 <SubjectPolicy>
  <SubjectDomainSpec ID="everywhere">
   <Include LDAPDN=""/>
  </SubjectDomainSpec>
 </SubjectPolicy>
 <RoleHierarchyPolicy>
  <RoleSpec Type="permisRole" OID="1.2.826.0.1.3344810.1.1.14">
  </RoleSpec>
 </RoleHierarchyPolicy>
 <SOAPolicy>
  <SOASpec ID="anyone" LDAPDN=""/>
 </SOAPolicy>
 <RoleAssignmentPolicy>
  <RoleAssignment>
   <SubjectDomain ID="everywhere"/>
   <RoleList>
    <Role Type="permisRole"/>
   </RoleList>
   <Delegate Depth="0"/>
   <SOA ID="anyone"/>
   <Validity/>
  </RoleAssignment>
 </RoleAssignmentPolicy>
<TargetPolicy>
 <TargetDomainSpec ID="PersonalData">
  <Include URL=""/>
  <ObjectClass Name="PersonalData"/>
 </TargetDomainSpec>
</TargetPolicy>
 <ActionPolicy>
  <Action Name="READ" ID="READ">
   <TargetDomain ID="PersonalData"/>
  </Action>
 </ActionPolicy>
 <TargetAccessPolicy>
 <!-- DataSubject can READ his/her personal data. -->
 <TargetAccess>
 <RoleList>
 </RoleList>
 <TargetList>
  <Target>
   <TargetDomain ID="PersonalData"/>
   <DeniedAction ID="READ"/>
  </Target>
 </TargetList>
 <IF>
   <NOT>
              <OR> <EQ>
                     <Environment Parameter="Purpose" Type="String" />
                <Constant Value="historical purpose" Type="String" />
                     </EQ>
```

```
                    <EQ>
                     <Environment Parameter="Purpose" Type="String" />
                     <Constant Value="statistical purpose" Type="String" />
                     </EQ>
                <EQ>
                    <Environment Parameter="Purpose" Type="String" />
                    <Constant Value="performance of contract" Type="String" />
                    </EQ>
                    <EQ>
                    <Environment Parameter="Purpose" Type="String" />
                    <Constant Value="entering into contract" Type="String" />
                    </EQ>
                    <EQ>
                    <Environment Parameter="Purpose" Type="String" />
                    <Environment Parameter="PurposeOfDataCollection" Type="String" />
                    </EQ>
                     <GT>
                    <Environment Parameter="ValidityTime" Type="Time" />
          <Environment Parameter="CurrentTime" Type="Time" />
        </GT>
          </OR>
         </NOT>
</IF>
</TargetAccess>
</TargetAccessPolicy>
</X.509_PMI_RBAC_Policy>

-------------------------------------------------------------------------------------------------------------------
<?xml version="1.0" encoding="UTF-8"?>
<X.509_PMI_RBAC_Policy OID="LawPolicy8" EnableNotApplicable="true">
 <SubjectPolicy>
  <SubjectDomainSpec ID="everywhere">
   <Include LDAPDN=""/>
  </SubjectDomainSpec>
 </SubjectPolicy>
 <RoleHierarchyPolicy>
  <RoleSpec Type="permisRole" OID="1.2.826.0.1.3344810.1.1.14">
   <SupRole Value="MedicalProfessional"/>
  </RoleSpec>
 </RoleHierarchyPolicy>
 <SOAPolicy>
  <SOASpec ID="anyone" LDAPDN=""/>
 </SOAPolicy>
 <RoleAssignmentPolicy>
  <RoleAssignment>
   <SubjectDomain ID="everywhere"/>
   <RoleList>
    <Role Type="permisRole"/>
   </RoleList>
   <Delegate Depth="0"/>
   <SOA ID="anyone"/>
   <Validity/>
  </RoleAssignment>
 </RoleAssignmentPolicy>
 <TargetPolicy>
 <TargetDomainSpec ID="MedicalData">
  <Include URL=""/>
  <ObjectClass Name="MedicalData"/>
 </TargetDomainSpec>
```

```
</TargetPolicy>
 <ActionPolicy>
  <Action Name="READ" ID="READ">
   <TargetDomain ID="MedicalData"/>
  </Action>
  <Action Name="Update" ID="Update">
   <TargetDomain ID="MedicalData"/>
  </Action>
  <Action Name="UpdateRequest" ID="UpdateRequest">
   <TargetDomain ID="MediclaData"/>
  </Action>
 </ActionPolicy>
 <TargetAccessPolicy>

  <!-- DataSubject can READ his/her personal data.
   -->
 <TargetAccess>
  <RoleList>
   <Role Type="permisRole" Value="MedicalProfessional"/>
  </RoleList>
   <TargetList>
    <Target>
     <TargetDomain ID="MedicalData"/>
     <AllowedAction ID="READ"/>
    </Target>
   </TargetList>
   <IF>
    <AND>
    <OR>
     <EQ>
     <Environment Parameter="Purpose" Type="String" />
     <Constant Value="medical diagnosis" Type="String" />
     </EQ>
            <EQ>
            <Environment Parameter="Purpose" Type="String" />
            <Constant Value="provision of care and treatment" Type="String" />
            </EQ>
            <EQ>
            <Environment Parameter="Purpose" Type="String" />
            <Constant Value="preventive medicine" Type="String" />
            </EQ>
            <EQ>
            <Environment Parameter="Purpose" Type="String" />
            <Constant Value="management of health care service" Type="String" />
            </EQ>
            <EQ>
            <Environment Parameter="Purpose" Type="String" />
            <Environment Parameter="PurposeOfDataCollection" Type="String" />
            </EQ>
      </OR>
            <EQ>
            <Environment Parameter="PhysicianID" Type="String" />
            <Environment Parameter="TreatingPhysicianID" Type="String" />
            </EQ>
    </AND>
   </IF>
  </TargetAccess>
 </TargetAccessPolicy>
</X.509_PMI_RBAC_Policy>
```

```
---------------------------------------------------------------------------------------------------------------------------
<?xml version="1.0" encoding="UTF-8"?>
<X.509_PMI_RBAC_Policy OID="LawPolicy9" EnableNotApplicable="false">
 <SubjectPolicy>
  <SubjectDomainSpec ID="everywhere">
   <Include LDAPDN=""/>
  </SubjectDomainSpec>
 </SubjectPolicy>
 <RoleHierarchyPolicy>
  <RoleSpec Type="permisRole" OID="1.2.826.0.1.3344810.1.1.14">
   <SupRole Value="MedicalProfessional"/>
  </RoleSpec>
 </RoleHierarchyPolicy>
 <SOAPolicy>
  <SOASpec ID="anyone" LDAPDN=""/>
 </SOAPolicy>
 <RoleAssignmentPolicy>
  <RoleAssignment>
   <SubjectDomain ID="everywhere"/>
   <RoleList>
    <Role Type="permisRole"/>
   </RoleList>
   <Delegate Depth="0"/>
   <SOA ID="anyone"/>
   <Validity/>
  </RoleAssignment>
 </RoleAssignmentPolicy>
 <TargetPolicy>
 <TargetDomainSpec ID="MedicalData">
  <Include URL=""/>
  <ObjectClass Name="MedicalData"/>
 </TargetDomainSpec>
</TargetPolicy>
 <ActionPolicy>
  <Action Name="READ" ID="READ">
   <TargetDomain ID="MedicalData"/>
  </Action>
  <Action Name="WRITE" ID="WRITE">
   <TargetDomain ID="MedicalData"/>
  </Action>
  <Action Name="BreakTheGlass" ID="BreakTheGlass">
   <TargetDomain ID="MedicalData"/>
  </Action>
 </ActionPolicy>
 <TargetAccessPolicy>

  <!-- DataSubject can READ his/her personal data.
   -->
 <TargetAccess>
   <RoleList>
    <Role Type="permisRole" Value="MedicalProfessional"/>
   </RoleList>
   <TargetList>
    <Target>
     <TargetDomain ID="MedicalData"/>
     <AllowedAction ID="BreakTheGlass"/>
    </Target>
   </TargetList>
```

```
    <IF>
     <OR>
              <EQ>
              <Environment Parameter="Purpose" Type="String" />
              <Constant Value="medical diagnosis" Type="String" />
              </EQ>
              <EQ>
              <Environment Parameter="Purpose" Type="String" />
              <Constant Value="provision of care and treatment" Type="String" />
              </EQ>
              <EQ>
              <Environment Parameter="Purpose" Type="String" />
              <Constant Value="preventive medicine" Type="String" />
              </EQ>
              <EQ>
              <Environment Parameter="Purpose" Type="String" />
              <Constant Value="management of health care service" Type="String" />
              </EQ>
     </OR>
     </IF>
     <Obligations>
      <Obligation ObligationId="BreakTheGlassObligation" FulfillOn="Permit">
       <AttributeAssignment AttributeId="BTG:(Role(S),ResourceType(R),urn:oasis:names:tc:xacml:1.0:action:action-id(A))"
        DataType="http://www.w3.org/2001/XMLSchema#boolean"></AttributeAssignment>
      </Obligation>
     </Obligations>
    </TargetAccess>
              <TargetAccess>
    <RoleList>
     <Role Type="permisRole" Value="MedicalProfessional"/>
    </RoleList>
    <TargetList>
     <Target>
      <TargetDomain ID="MedicalData"/>
       <AllowedAction ID="READ"/>
       <AllowedAction ID="WRITE"/>
     </Target>
     </TargetList>
     <IF>
      <AND>
               <EQ>
               <Environment Parameter="BTG:(Role(S),ResourceType(R),urn:oasis:names:tc:xacml:1.0:action:action-id(A))"
Type="Boolean"/>
        <Constant Type="Boolean" Value="true"/>
      </EQ>
              <OR>
              <EQ>
       <Environment Parameter="Purpose" Type="String" />
              <Constant Value="medical diagnosis" Type="String" />
              </EQ>
              <EQ>
               <Environment Parameter="Purpose" Type="String" />
              <Constant Value="provision of care and treatment" Type="String" />
              </EQ>
               <EQ>
               <Environment Parameter="Purpose" Type="String" />
               <Constant Value="preventive medicine" Type="String" />
               </EQ>
              <EQ>
```

```
          <Environment Parameter="Purpose" Type="String" />
          <Constant Value="management of health care service" Type="String" />
          </EQ>
    </OR>
   </AND>
  </IF>
 </TargetAccess>
</TargetAccessPolicy>
</X.509_PMI_RBAC_Policy>
```

Appendix 5: Example Policies and Request Context for Validation Tests

Policies for use case scenario 1 presented in Section 5.5.1

Appendix 4.1: CRP constructed with Isseur's CRR, DataSubject's CRR, Controller's CRR and the default CRR.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- New document created with EditiX at Thu Sep 09 13:42:12 BST 2010 -->
<PolicySet xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
/home/kaniz/Desktop/access_control-xacml-2.0-policy-schema-os.xsd"
PolicySetId="CRRofKentHealthCentre" PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-
combining-algorithm:permit-overrides" xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <Target xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os">
                <Subjects/><Resources/><Actions/>
        </Target>
        <Policy PolicyId="IssuerCRR1" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-
combining-algorithm:deny-overrides">
                <Target/>
                <Rule RuleId="IssuerCRR1" Effect="Permit">
                        <Description>If the ResourceType is MedicalData and the requester is a
MedicalProfessional DCA=grantOverrides.</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">MedicalProfessional</AttributeValue>
                                                <SubjectAttributeDesignator
AttributeId="Role" 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">MedicalData</AttributeValue>
         <ResourceAttributeDesignator AttributeId="ResourceType"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
```

```
  </ResourceMatch>
 </Resource>
</Resources>
<Actions/>
</Target>
</Rule>
<Obligations><Obligation
ObligationId="http://sec.cs.kent.ac.uk/masterpdp/conflictresolution/permit-overrides"
FulfillOn="Permit"/></Obligations>
</Policy>
<Policy PolicyId="IssuerCRR2" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:deny-overrides">
 <Target/>
  <Rule RuleId="IssuerCRR2" Effect="Permit">
                        <Description>If the ResourceType is MedicalData and the requester is a
MedicalProfessional DCA=grantOverrides.</Description>
   <Target><Subjects/>
    <Resources><Resource><ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
equal">
        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Prescription</AttributeValue>
                                                    <ResourceAttributeDesignator
AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string"/>
         </ResourceMatch></Resource>    </Resources>
         <Actions/>
        </Target>
         </Rule>
               <Obligations><Obligation
ObligationId="http://sec.cs.kent.ac.uk/masterpdp/conflictresolution/permit-overrides"
FulfillOn="Permit"/></Obligations>
   </Policy>
   <Policy PolicyId="IssuerCRR" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-
combining-algorithm:deny-overrides">
               <Target/>
        <Rule RuleId="IssuerCRR" Effect="Permit">
                        <Description>If the ResourceType is MedicalData and the requester is a
MedicalProfessional DCA=grantOverrides.</Description>
         <Target><Subjects/>
      <Resources><Resource><ResourceMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                                                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">MedicalData</AttributeValue>
                                                    <ResourceAttributeDesignator
AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch></Resource>    </Resources>
           <Actions/>
          </Target>
         <Condition> <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:not">
                 <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-
member-of">
                        <SubjectAttributeDesignator AttributeId="Role"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
                              <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
bag">
                         <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">MedicalProfessional</AttributeValue>
                         </Apply></Apply></Apply>
    </Condition>
  </Rule>
        <Obligations><Obligation
ObligationId="http://sec.cs.kent.ac.uk/masterpdp/conflictresolution/permit-overrides"
```

```
FulfillOn="Permit"/></Obligations>
 </Policy>
 <Policy PolicyId="DataSubjectCRR" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-
combining-algorithm:deny-overrides">
   <Target/>
         <Rule RuleId="DataSubjectCRR" Effect="Permit">
         <Description>If the ResourceType is MedicalData DCA=denyOverrides.</Description>
          <Target> <Subjects></Subjects>
                <Resources><Resource><ResourceMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">MedicalData</AttributeValue>
                                                        <ResourceAttributeDesignator
AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string"/>
                </ResourceMatch></Resource></Resources>
                <Actions/>
           </Target>
         </Rule>
                <Obligations><Obligation
ObligationId="http://sec.cs.kent.ac.uk/masterpdp/conflictresolution/deny-overrides"
FulfillOn="Permit"/>
                </Obligations>
        </Policy>
        <Policy PolicyId="ControllerCRR" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-
combining-algorithm:deny-overrides">
                <Target/>
                <Rule RuleId="ControllerCRR" Effect="Permit"><Description>If the ResourceType
is MedicalData DCA=denyOverrides.</Description>
                  <Target/>
                </Rule>
                <Obligations>
                        <Obligation
ObligationId="http://sec.cs.kent.ac.uk/masterpdp/conflictresolution/permit-overrides"
FulfillOn="Permit"/>
                </Obligations>
         </Policy>
         <Policy PolicyId="DefaultCRR" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-
combining-algorithm:deny-overrides">
                <Target/>

                <Rule RuleId="DefaultCRR" Effect="Permit">
                        <Description>If the ResourceType is MedicalData
DCA=denyOverrides.</Description>
                        <Target/
                </Rule>
                <Obligations>
                        <Obligation
ObligationId="http://sec.cs.kent.ac.uk/masterpdp/conflictresolution/permit-overrides"
FulfillOn="Permit"/>
                </Obligations>
         </Policy>

</PolicySet>
```

Appendix 4.2: ACP of Issuer for use case 1 of Section 5.5.1

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- New document created with EditiX at Thu Sep 09 13:42:12 BST 2010 -->
<PolicySet xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
/home/kaniz/Desktop/access_control-xacml-2.0-policy-schema-os.xsd"
```

```
PolicySetId="IssuerPolicySet" PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-
combining-algorithm:permit-overrides" xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <Target xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os">
                <Subjects/><Resources/><Actions/>
        </Target>
        <Policy PolicyId="IssuerPolicy1" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-
combining-algorithm:deny-overrides">
                <Target/>
                <Rule RuleId="IssuerRule1" Effect="Permit">
                        <Description>MedicalProfessional can change the value of
MedicalObjection attribute of the Medical Data or of each Medical Records in the Medical Data
separately for the purpose of saving the vital interest of the data subject.</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">MedicalProfessional</AttributeValue>
         <SubjectAttributeDesignator AttributeId="Role"
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">MedicalObligation</AttributeValue>
         <ResourceAttributeDesignator AttributeId="ResourceAttributeType"
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">Write</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></Policy> </PolicySet>
```

Appendix 4.3: ACP of DataSubject for use case 1 of Section 5.5.1

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- New document created with EditiX at Thu Sep 09 13:42:12 BST 2010 -->
<PolicySet xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
/home/kaniz/Desktop/access_control-xacml-2.0-policy-schema-os.xsd"
PolicySetId="DataSubjectPolicySet" PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-
combining-algorithm:permit-overrides" xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <Target xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os">
                <Subjects/><Resources/><Actions/>
        </Target>
        <Policy PolicyId="DataSubjectPolicy1"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
                <Target/>
                <Rule RuleId="DataSubjectRule1" Effect="Permit">
                        <Description>Dr. D of Kent Health Center can READ /WRITE my
MedicalData.</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">MedicalProfessional</AttributeValue>
                                        <SubjectAttributeDesignator AttributeId="Role"
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">Dr. D</AttributeValue>
                                <SubjectAttributeDesignator AttributeId="Name"
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">KentHealthCenter</AttributeValue>
                                <SubjectAttributeDesignator AttributeId="Organisation"
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">MedicalData</AttributeValue>
                                <ResourceAttributeDesignator AttributeId="ResourceType"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
                    </ResourceMatch></Resource></Resources>
                    <Actions></Actions>
                    </Target>
                            <Condition> <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:not">
                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Read</AttributeValue>
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Write</AttributeValue>
                                                </Apply></Apply></Apply>
                    </Condition>
                        </Rule>
                </Policy>
        <Policy PolicyId="DataSubjectPolicy2"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
                <Target/>
                <Rule RuleId="DataSubjectRule2" Effect="Permit">
                        <Description>•  Researchers are allowed to view the Medical Data and
Prescription if the data can be anonymised.</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">Researcher</AttributeValue>
                                        <SubjectAttributeDesignator AttributeId="Role"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
                            </SubjectMatch></Subject>
                        </Subjects>
                        <Resources></Resources>
                        <Actions></Actions>
                        </Target>
                                <Condition> <Apply
```

```
FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Read</AttributeValue>
                                        </Apply></Apply>
                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                <ResourceAttributeDesignator
AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string"/>
                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">MedicalData</AttributeValue>
                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Prescription</AttributeValue>
                                                </Apply></Apply>
                                        </Apply>
                        </Condition>
                </Rule>
                        <Obligations><Obligation ObligationId="Anonymise data"
FulfillOn="Permit"/>
                        </Obligations>
        </Policy>
</PolicySet>
```

Appendix 4.4: ACP of Controller for use case 1 of Section 5.5.1

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- New document created with EditiX at Thu Sep 09 13:42:12 BST 2010 -->
<PolicySet xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
/home/kaniz/Desktop/access_control-xacml-2.0-policy-schema-os.xsd"
PolicySetId="ControllerPolicySet" PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-
combining-algorithm:permit-overrides" xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <Target xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os">
                <Subjects/><Resources/><Actions/>
        </Target>
        <Policy PolicyId="ControllerPolicy1"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
                <Target/>
                <Rule RuleId="ControllerRule1" Effect="Permit">
                        <Description>Administrative Officers can read and write administrative
data. </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">AdministrativeOfficer</AttributeValue>
                                        <SubjectAttributeDesignator AttributeId="Role"
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">KentHealthCenter</AttributeValue>
                                <SubjectAttributeDesignator AttributeId="Organisation"
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">AdministrativeData</AttributeValue>
                                <ResourceAttributeDesignator AttributeId="ResourceType"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
                        </ResourceMatch></Resource></Resources>
                        <Actions></Actions>
                        </Target>
                                <Condition>
                                  <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
                                            <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Read</AttributeValue>
                                                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Write</AttributeValue>
                                                </Apply></Apply>
                    </Condition>
                        </Rule>
                        <Rule RuleId="ControllerRule2" Effect="Deny">
                        <Description>Administrative Officers can’t access the medical data.
</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">AdministrativeOfficer</AttributeValue>
                                        <SubjectAttributeDesignator AttributeId="Role"
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">KentHealthCenter</AttributeValue>
                                <SubjectAttributeDesignator AttributeId="Organisation"
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">MedicalData</AttributeValue>
                                <ResourceAttributeDesignator AttributeId="ResourceType"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
                        </ResourceMatch></Resource></Resources>
                        <Actions></Actions>
                        </Target>
                                <Condition>
```

```
                                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Read</AttributeValue>
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Write</AttributeValue>
                                                </Apply></Apply>
                        </Condition>
                            </Rule>
                            <Rule RuleId="ControllerRule3" Effect="Permit">
                            <Description>Financial Officers can access the billing and payment
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">FinancialOfficer</AttributeValue>
                                                <SubjectAttributeDesignator AttributeId="Role"
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">KentHealthCenter</AttributeValue>
                                        <SubjectAttributeDesignator AttributeId="Organisation"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
                                    </SubjectMatch></Subject>
                            </Subjects>
                            <Resources></Resources>
                            <Actions></Actions>
                            </Target>
                                    <Condition>
                                      <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
                                                  <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Read</AttributeValue>
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Write</AttributeValue>
                                                  </Apply></Apply>
                                                  <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                <ResourceAttributeDesignator
AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string"/>
                                                  <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">BillingInformation</AttributeValue>
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">PaymentInformation</AttributeValue>
                                                  </Apply></Apply>
```

```
                    </Condition>
                        </Rule>
                        <Rule RuleId="ControllerRule4" Effect="Deny">
                        <Description>Financial Officers can't access the medical data or
administrative data.</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">FinancialOfficer</AttributeValue>
                                        <SubjectAttributeDesignator AttributeId="Role"
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">KentHealthCenter</AttributeValue>
                                        <SubjectAttributeDesignator AttributeId="Organisation"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
                                </SubjectMatch></Subject>
                        </Subjects>
                        <Resources></Resources>
                        <Actions></Actions>
                        </Target>
                                <Condition>
                                  <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
                                          <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Read</AttributeValue>
                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Write</AttributeValue>
                                        </Apply></Apply>
                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                <ResourceAttributeDesignator
AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string"/>
                                          <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">MedicalData</AttributeValue>
                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">AdministrativeData</AttributeValue>
                                          </Apply></Apply>
                    </Condition>
                        </Rule>
                        <Rule RuleId="ControllerRule4" Effect="Deny">
                        <Description>Medical professionals can't access the billing and payment
information or the patient's financial 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">MedicalProfessional</AttributeValue>
```

```
                                        <SubjectAttributeDesignator AttributeId="Role"
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">KentHealthCenter</AttributeValue>
                                    <SubjectAttributeDesignator AttributeId="Organisation"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
                            </SubjectMatch></Subject>
                        </Subjects>
                        <Resources></Resources>
                        <Actions></Actions>
                        </Target>
                                <Condition>
                                  <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
                                            <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Read</AttributeValue>
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Write</AttributeValue>
                                            </Apply></Apply>
                                            <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                                <ResourceAttributeDesignator
AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string"/>
                                            <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">BillingInformation</AttributeValue>
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">FinancialInformation</AttributeValue>
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Patient'sFinancialInformation</AttributeValu
e>
                                    </Apply></Apply>
                    </Condition>
                        </Rule>

                        </Policy>

</PolicySet>
```

Appendix 4.5: Request Contexts (RC)

RC No.	Request Contexts
1.	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header/> <soapenv:Body> <XACMLAuthzDecisionQuery xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01" ID="A2010-12-13T12.58.12" Version="2.0" IssueInstant="2010-12-13T12:58:12.209Z"> <xacml-context:Request xmlns:xacml- context="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">

	<xacml-context:Attribute AttributeId="Role" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>MedicalProfessional</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="Name" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Dr. D</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="Organisation" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>KentHealthCenter</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Subject> <xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="rid" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>rid-mrm123</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>MedicalData</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Resource> <xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Read</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Action> <xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"/> </xacml-context:Request> </XACMLAuthzDecisionQuery> </soapenv:Body> </soapenv:Envelope>
2.	This RC is same as RC1 except for the value of action-id attribute which will be as follows <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Write</xacml-context:AttributeValue> </xacml-context:Attribute>
3.	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header/> <soapenv:Body> <XACMLAuthzDecisionQuery xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01" ID="A2010-12-13T12.58.12" Version="2.0" IssueInstant="2010-12-13T12:58:12.209Z"> <xacml-context:Request xmlns:xacml- context="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="Role" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>MedicalProfessional</xacml-context:AttributeValue>

```
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="Name"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>Dr. S</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="Organisation"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>LondonHospital</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-
id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="rid"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>rid-mrm123</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="ResourceType"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>MedicalData</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>Read</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"/>
</xacml-context:Request>
</XACMLAuthzDecisionQuery>
</soapenv:Body>
</soapenv:Envelope>
```

4.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<XACMLAuthzDecisionQuery
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
ID="A2010-12-13T12.58.12"
Version="2.0"
IssueInstant="2010-12-13T12:58:12.209Z">
<xacml-context:Request xmlns:xacml-
context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="NHSNumber"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>MRM061281</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-
id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="rid"
DataType="http://www.w3.org/2001/XMLSchema#string">
```

<table>
<tr><td></td><td>
<xacml-context:AttributeValue>rid-mrm123</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="ResourceType"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>MedicalData</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="DataSubject'sNHSNumber"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>MRM061281</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>UpdatePolicy</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"/>
</xacml-context:Request>
</XACMLAuthzDecisionQuery>
</soapenv:Body>
</soapenv:Envelope>
</td></tr>
<tr><td>5.</td><td>Same as RC3.</td></tr>
<tr><td>6.</td><td>Same as RC3 except the value of action-id will be “Write” instead of “Read”.</td></tr>
<tr><td>7.</td><td>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<XACMLAuthzDecisionQuery
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
ID="A2010-12-13T12.58.12"
Version="2.0"
IssueInstant="2010-12-13T12:58:12.209Z">
<xacml-context:Request xmlns:xacml-
context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="Role"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>MedicalProfessional</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="Name"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>Dr. S</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="Organisation"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>LondonHospital</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-
id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="rid"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>rid-mrm123</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="ResourceAttributeType"
DataType="http://www.w3.org/2001/XMLSchema#string">
</td></tr>
</table>

	<xacml-context:AttributeValue>MedicalObjection</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Resource> <xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Write</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Action> <xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"/> </xacml-context:Request> </XACMLAuthzDecisionQuery> </soapenv:Body> </soapenv:Envelope>
8.	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header/> <soapenv:Body> <XACMLAuthzDecisionQuery xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01" ID="A2010-12-13T12.58.12" Version="2.0" IssueInstant="2010-12-13T12:58:12.209Z"> <xacml-context:Request xmlns:xacml- context="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="NHSNumber" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>MRM061281</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Subject> <xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource- id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="rid" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>rid-mrm123</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>MedicalData</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="DataSubject'sNHSNumber" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>MRM061281</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="MedicalObjection" DataType="http://www.w3.org/2001/XMLSchema#boolean"> <xacml-context:AttributeValue>true</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Resource> <xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Read</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Action> <xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"/>

	</xacml-context:Request> </XACMLAuthzDecisionQuery> </soapenv:Body> </soapenv:Envelope>
9.	Same as RC 8 removing MedicalObjection attribute
10.	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header/> <soapenv:Body> <XACMLAuthzDecisionQuery xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01" ID="A2010-12-13T12.58.12" Version="2.0" IssueInstant="2010-12-13T12:58:12.209Z"> <xacml-context:Request xmlns:xacml- context="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="Role" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>MedicalProfessional</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="Name" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Dr. D</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="Organisation" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>LondonHospital</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Subject> <xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="rid" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>rid-mrm123</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="ResourceAttributeType" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>MedicalObjection</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Resource> <xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Write</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Action> <xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"/> </xacml-context:Request> </XACMLAuthzDecisionQuery> </soapenv:Body> </soapenv:Envelope>
11.	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header/> <soapenv:Body> <XACMLAuthzDecisionQuery

```
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
ID="A2010-12-13T12.58.12"
Version="2.0"
IssueInstant="2010-12-13T12:58:12.209Z">
<xacml-context:Request xmlns:xacml-
context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="Role"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>AdministrativeOfficer</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="Organisation"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>KentHealthCenter</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-
id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="rid"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>rid-mrm123</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="ResourceType"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>MedicalData</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>Read</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"/>
</xacml-context:Request>
</XACMLAuthzDecisionQuery>
</soapenv:Body>
</soapenv:Envelope>
```

12.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<XACMLAuthzDecisionQuery
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
ID="A2010-12-13T12.58.12"
Version="2.0"
IssueInstant="2010-12-13T12:58:12.209Z">
<xacml-context:Request xmlns:xacml-
context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="Role"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>MedicalProfessional</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="Name"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>Dr. L</xacml-context:AttributeValue>
```

```
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="Organisation"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>KentHealthCenter</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-
id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="ResourceType"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>BillingInformation</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>Read</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"/>
</xacml-context:Request>
</XACMLAuthzDecisionQuery>
</soapenv:Body>
</soapenv:Envelope>
```

13.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<XACMLAuthzDecisionQuery
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
ID="A2010-12-13T12.58.12"
Version="2.0"
IssueInstant="2010-12-13T12:58:12.209Z">
<xacml-context:Request xmlns:xacml-
context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="NHSNumber"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>MRM010777</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-
id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="rid"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>rid-mrm123</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="ResourceType"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>MedicalData</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="DataSubject'sNHSNumber"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>MRM061281</xacml-context:AttributeValue>
```

```
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>Read</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"/>
</xacml-context:Request>
</XACMLAuthzDecisionQuery>
</soapenv:Body>
</soapenv:Envelope>
```

14.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<XACMLAuthzDecisionQuery
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
ID="A2010-12-13T12.58.12"
Version="2.0"
IssueInstant="2010-12-13T12:58:12.209Z">
<xacml-context:Request xmlns:xacml-
context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="Role"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>MedicalProfessional</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="Name"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>Dr. T</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="Organisation"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>HospitalH</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-
id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="rid"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>rid-mrm123</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="ResourceType"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>MedicalData</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>Read</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"/>
</xacml-context:Request>
```

	</XACMLAuthzDecisionQuery> </soapenv:Body> </soapenv:Envelope>
15.	Same as RC 14 except the value of action-id will be as follows: <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>BreakTheGlass</xacml-context:AttributeValue> </xacml-context:Attribute>
16.	Same as RC 14 since the BTG variable is handled by the software it does not need to be in the RC.
17.	Same as RC 16 expect the value of action-id will be "Write".
18.	Same as RC 13 since the BTG variable is handled by the software it does not need to be in the RC.
19.	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header/> <soapenv:Body> <XACMLAuthzDecisionQuery xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01" ID="A2010-12-13T12.58.12" Version="2.0" IssueInstant="2010-12-13T12:58:12.209Z"> <xacml-context:Request xmlns:xacml- context="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="Role" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Researcher</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="Organisation" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>UniversityOfKent</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Subject> <xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="rid" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>rid-mrm123</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>MedicalData</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Resource> <xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Read</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Action> <xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"/> </xacml-context:Request> </XACMLAuthzDecisionQuery> </soapenv:Body> </soapenv:Envelope>

20.	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header/> <soapenv:Body> <XACMLAuthzDecisionQuery xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01" ID="A2010-12-13T12.58.12" Version="2.0" IssueInstant="2010-12-13T12:58:12.209Z"> <xacml-context:Request xmlns:xacml- context="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="Organisation" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>GHealthCenter</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Subject> <xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource- id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="rid" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>rid-mrm123</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>MedicalData</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Resource> <xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>TRANSFER</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="TransferToCountry" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Germany</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Action> <xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"/> </xacml-context:Request> </XACMLAuthzDecisionQuery> </soapenv:Body> </soapenv:Envelope>
21.	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header/> <soapenv:Body> <XACMLAuthzDecisionQuery xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01" ID="A2010-12-13T12.58.12" Version="2.0" IssueInstant="2010-12-13T12:58:12.209Z"> <xacml-context:Request xmlns:xacml- context="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="Organisation" DataType="http://www.w3.org/2001/XMLSchema#string">

```
<xacml-context:AttributeValue>GHealthCenter</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-
id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="rid"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>rid-mrm123</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="ResourceType"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>MedicalData</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>TRANSFER</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="TransferToCountry"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>Germany</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="SubjectConsentsToTransferTo"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>GHealthCenter</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Environment>
</xacml-context:Request>
</XACMLAuthzDecisionQuery>
</soapenv:Body>
</soapenv:Envelope>
```

22.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<XACMLAuthzDecisionQuery
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
ID="A2010-12-13T12.58.12"
Version="2.0"
IssueInstant="2010-12-13T12:58:12.209Z">
<xacml-context:Request xmlns:xacml-
context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="Role"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>Employee</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="Organisation"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>PharmacyB</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-
```

	id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Prescription</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Resource> <xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>TRANSFER</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="TransferToCountry" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>United Kingdom</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Action> <xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="Contract-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>http://localhost/contract.xml</xacml- context:AttributeValue> </xacml-context:Attribute> </xacml-context:Environment> </xacml-context:Request> </XACMLAuthzDecisionQuery> </soapenv:Body> </soapenv:Envelope> Note: This RC is sent to PEP which then added the contract related information with the Contract-id.
23.	Same as RC19 except the value of ResourceType is “Prescription”.
24.	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header/> <soapenv:Body> <XACMLAuthzDecisionQuery xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01" ID="A2010-12-13T12.58.12" Version="2.0" IssueInstant="2010-12-13T12:58:12.209Z"> <xacml-context:Request xmlns:xacml- context="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="Role" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Employee</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="Organisation" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>PharmacyB</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Subject> <xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string">

```
<xacml-context:AttributeValue>Prescription</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>TRANSFER</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="TransferToCountry"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>United Kingdom</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="Contract-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>http://localhost/contract.xml</xacml-
context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="RequestTime"
DataType="http://www.w3.org/2001/XMLSchema#date">
<xacml-context:AttributeValue>2015-06-12</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="ContractEndTime"
DataType="http://www.w3.org/2001/XMLSchema#date">
<xacml-context:AttributeValue>2015-06-11</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Environment>
</xacml-context:Request>
</XACMLAuthzDecisionQuery>
</soapenv:Body>
</soapenv:Envelope>
```

Note: PEP adds the contract related information with the Contract-id.

25.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<XACMLAuthzDecisionQuery
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
ID="A2010-12-13T12.58.12"
Version="2.0"
IssueInstant="2010-12-13T12:58:12.209Z">
<xacml-context:Request xmlns:xacml-
context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="Role"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>Employee</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="Organisation"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>HIC</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-
id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="ResourceType"
DataType="http://www.w3.org/2001/XMLSchema#string">
```

```
<xacml-context:AttributeValue>BillingInformation</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>Read</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="Contract-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>http://localhost/contract2.xml</xacml-
context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Environment>
</xacml-context:Request>
</XACMLAuthzDecisionQuery>
</soapenv:Body>
</soapenv:Envelope>
```

Note: PEP adds the contract related information with the Contract-id.

26.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<XACMLAuthzDecisionQuery
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
ID="A2010-12-13T12.58.12"
Version="2.0"
IssueInstant="2010-12-13T12:58:12.209Z">
<xacml-context:Request xmlns:xacml-
context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="Role"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>SeniorOfficer</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="Organisation"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>BankB</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-
id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="ResourceType"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>BankDetails</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>TRANSFER</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="Contract-id"
```

```
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>http://localhost/contract3.xml</xacml-
context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="TransferToCountry"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>United Kingdom</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Environment>
</xacml-context:Request>
</XACMLAuthzDecisionQuery>
</soapenv:Body>
</soapenv:Envelope>
```

Note: PEP adds the contract related information with the Contract-id.

27.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<XACMLAuthzDecisionQuery
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
ID="A2010-12-13T12.58.12"
Version="2.0"
IssueInstant="2010-12-13T12:58:12.209Z">
<xacml-context:Request xmlns:xacml-
context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="E-mailAddress"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>mrm@mail.com</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-
id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="ResourceType"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>BankDetails</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="rid"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>rid-mrm123-b</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="DataSubject'sE-mailAddress"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>mrm@mail.com</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>Read</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
</xacml-context:Environment>
</xacml-context:Request>
</XACMLAuthzDecisionQuery>
</soapenv:Body>
</soapenv:Envelope>
```

	Note: PEP adds the contract related information with the Contract-id.
28.	Same as RC26 except that the value of Role is “Junior Officer”.
29.	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header/> <soapenv:Body> <XACMLAuthzDecisionQuery xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01" ID="A2010-12-13T12.58.12" Version="2.0" IssueInstant="2010-12-13T12:58:12.209Z"> <xacml-context:Request xmlns:xacml- context="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="Role" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Staff</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="Organisation" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>jobs.com</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Subject> <xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>CV</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="rid" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>rid-mrm123-j</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Resource> <xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Read</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Action> <xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> </xacml-context:Environment> </xacml-context:Request> </XACMLAuthzDecisionQuery> </soapenv:Body> </soapenv:Envelope>
30.	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header/> <soapenv:Body> <XACMLAuthzDecisionQuery xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01" ID="A2010-12-13T12.58.12" Version="2.0" IssueInstant="2010-12-13T12:58:12.209Z"> <xacml-context:Request xmlns:xacml- context="urn:oasis:names:tc:xacml:2.0:context:schema:os">

	<xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="Role" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Staff</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="Organisation" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>jobs.com</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Subject> <xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>DegreeCertificate</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="rid" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>rid-mrm123-u</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Resource> <xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>TRANSFER</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Action> <xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="TransferToCountry" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>United Kingdom</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Environment> </xacml-context:Request> </XACMLAuthzDecisionQuery> </soapenv:Body> </soapenv:Envelope>
31.	Same as Rc 30 but with an environment attribute as follows: <xacml-context:Attribute AttributeId="SubjectConsentsToTransferTo" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>GHealthCenter</xacml-context:AttributeValue> </xacml-context:Attribute>
32.	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header/> <soapenv:Body> <XACMLAuthzDecisionQuery xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01" ID="A2010-12-13T12.58.12" Version="2.0" IssueInstant="2010-12-13T12:58:12.209Z"> <xacml-context:Request xmlns:xacml- context="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="E-mailAddress" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>mrm@mail.com</xacml-context:AttributeValue>

	</xacml-context:Attribute> </xacml-context:Subject> <xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>DegreeCertificate</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="rid" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>rid-mrm123-u</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="DataSubject'sE-mailAddress" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>mrm@mail.com</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Resource> <xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Read</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Action> <xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> </xacml-context:Environment> </xacml-context:Request> </XACMLAuthzDecisionQuery> </soapenv:Body> </soapenv:Envelope>
33.	Same as RC32 except that the value of action-id will be “Update”.
34.	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header/> <soapenv:Body> <XACMLAuthzDecisionQuery xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01" ID="A2010-12-13T12.58.12" Version="2.0" IssueInstant="2010-12-13T12:58:12.209Z"> <xacml-context:Request xmlns:xacml- context="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="Role" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Staff</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="Organisation" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>CompanyB</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Subject> <xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue> </xacml-context:Attribute>

	<xacml-context:Attribute AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>CV</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="rid" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>rid-mrm123-j</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Resource> <xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Read</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Action> <xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> </xacml-context:Environment> </xacml-context:Request> </XACMLAuthzDecisionQuery> </soapenv:Body> </soapenv:Envelope>
35.	Same as RC29 except that the value of action-id is “Write”.
36.	Same as RC29 except that the value of action-id is “Update”.
37.	Same as RC34 except that the value of Organisation is “companyC”.
38.	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header/> <soapenv:Body> <XACMLAuthzDecisionQuery xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01" ID="A2010-12-13T12.58.12" Version="2.0" IssueInstant="2010-12-13T12:58:12.209Z"> <xacml-context:Request xmlns:xacml- context="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="Role" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Staff</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="Organisation" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>CompanyC</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Subject> <xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>CV</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="rid" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>rid-mrm123-j</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Resource> <xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">

	<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>TRANSFER</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Action> <xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="TransferToCountry" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Germany</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Environment> </xacml-context:Request> </XACMLAuthzDecisionQuery> </soapenv:Body> </soapenv:Envelope>
39.	Same as RC34 except that the value of Organisation is "CompanyX".
40.	<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header/> <soapenv:Body> <XACMLAuthzDecisionQuery xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01" ID="A2010-12-13T12.58.12" Version="2.0" IssueInstant="2010-12-13T12:58:12.209Z"> <xacml-context:Request xmlns:xacml- context="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="E-mailAddress" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>mrm@mail.com</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Subject> <xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource- id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="ResourceType" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>CV</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="rid" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>rid-mrm123-j</xacml-context:AttributeValue> </xacml-context:Attribute> <xacml-context:Attribute AttributeId="DataSubject'sE-mailAddress" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>mrm@mail.com</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Resource> <xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> <xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"> <xacml-context:AttributeValue>Read</xacml-context:AttributeValue> </xacml-context:Attribute> </xacml-context:Action> <xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"> </xacml-context:Environment> </xacml-context:Request>

	</XACMLAuthzDecisionQuery> </soapenv:Body> </soapenv:Envelope>
41.	Same as RC40 except that the value of action-id is “Update”.
42.	Same as RC36 except that the value of Organisation is “CompanyC”.
43.	Same as RC30 except that the value of Organisation is “CompanyC”.

Appendix 4.6: Policies and Request Contexts of Section 5.5.4

Appendix 4.6.1 Controller Policy of the system for use case scenario 4

```
--Controller PDP Policy for use case scenario 4--
<?xml version="1.0" encoding="UTF-8"?>
<!-- New document created with EditiX at Thu Sep 09 13:42:12 BST 2010 -->
<PolicySet xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
/home/kaniz/Desktop/access_control-xacml-2.0-policy-schema-os.xsd"
PolicySetId="controllerpolicy" PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-
combining-algorithm:permit-overrides" xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <Target xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os">
        </Target>
        <Policy PolicyId="MemberSubmit" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-
combining-algorithm:deny-overrides">
                <Target/>
                <Rule RuleId="MemberSubmit" Effect="Permit">
                        <Description>Anyone with Member role can submit policy </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">member</AttributeValue>
                                                        <SubjectAttributeDesignator
AttributeId="Role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
                                                </SubjectMatch>
                  </Subject></Subjects>
                        <Resources/>
                        <Actions><Action>
                                                <ActionMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">SUBMIT</AttributeValue>
                                                        <ActionAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
                                                </ActionMatch>
                        </Action></Actions>
                        </Target>
                </Rule>
        </Policy>
        <Policy PolicyId="MemberTransfer"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
         <Target/>
         <Rule RuleId="MemberTransfer" Effect="Permit">
                        <Description>Anyone with member Role can TRANSFER policy </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">member</AttributeValue>
                                                    <SubjectAttributeDesignator
AttributeId="Role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
                                                </SubjectMatch>
                                            </Subject>
                                </Subjects>
                                <Resources/>
                                <Actions><Action>
                                                <ActionMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                                                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">TRANSFER</AttributeValue>
                                                    <ActionAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
                                                </ActionMatch>
                            </Action></Actions>
                           </Target>
                </Rule>
                <Obligations>
                        <Obligation
ObligationId="http://sec.cs.kent.ac.uk/obligations/AttachStickyPolicy" FulfillOn="Permit">

        </Obligation>
        </Obligations>
         </Policy>
</PolicySet>
```

Appendix 4.6.2 Request Context to Read personal data by MyFriend Role

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<XACMLAuthzDecisionQuery
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
ID="A2010-12-13T12.58.12"
Version="2.0"
IssueInstant="2010-12-13T12:58:12.209Z">
<xacml-context:Request xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="Role"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>MyFriend</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="rid"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>rid-1</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
```

```
<xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>Read</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"/>
</xacml-context:Request>
</XACMLAuthzDecisionQuery>
</soapenv:Body>
</soapenv:Envelope>
```

Appendix 4.6.3 Request Context to SUBMIT an XACML policy

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<XACMLAuthzDecisionQuery
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
ID="A2010-12-13T12.58.12"
Version="2.0"
IssueInstant="2010-12-13T12:58:12.209Z">
<xacml-context:Request xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="Role"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>member</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="rid"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>rid-1</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>SUBMIT</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"/>
</xacml-context:Request>
<Extensions><sp:StickyPolicy
PolicyID="sticky-policy-1"
PolicyLanguage="XACML"
PolicyType="Authorisation"
TimeOfCreation="2010-08-09T00:00:00Z"
xmlns:sp="http://sec.cs.kent.ac.uk/stickypolicy">
<sp:PolicyAuthor>
<sp:AuthorType>DataSubject</sp:AuthorType>
</sp:PolicyAuthor>
<sp:PolicyResourceTypes>
<sp:ResourceType>personal:preferences</sp:ResourceType>
</sp:PolicyResourceTypes>
<sp:PolicyContents><PolicySet PolicySetId="dspolicy"
PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:first-applicable"
xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os access_control-xacml-2.0-
```

```
policy-schema-os.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os">
<Target xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os">
</Target>
 <Policy PolicyId="Policy1" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:deny-overrides">
 <Target></Target>
 <Rule RuleId="Mypolicy" Effect="Permit">
 <Description>MyFriend can Read </Description>
 <Target>
 </Target>
 <Condition>
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                 <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-
member-of">
                 <SubjectAttributeDesignator AttributeId="Role"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                 <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">MyFriend</AttributeValue>
                 </Apply></Apply>
                 <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-
member-of">
                 <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                 <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                 <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Read</AttributeValue>
                 </Apply></Apply>
                 </Apply>
 </Condition>
 </Rule><Obligations>
        <Obligation ObligationId="LogTheRequest" FulfillOn="Permit"></Obligation>
    </Obligations>
</Policy>
</PolicySet>
</sp:PolicyContents>
</sp:StickyPolicy>
</Extensions>
</XACMLAuthzDecisionQuery>
</soapenv:Body>
</soapenv:Envelope>
```

Appendix 4.6.4 Request Context to SUBMIT a PERMIS policy

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<XACMLAuthzDecisionQuery
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
ID="A2010-12-13T12.58.12"
Version="2.0"
IssueInstant="2010-12-13T12:58:12.209Z">
<xacml-context:Request xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="Role"
DataType="http://www.w3.org/2001/XMLSchema#string">
```

```
<xacml-context:AttributeValue>member</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="rid"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>rid-1</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>SUBMIT</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"/>
</xacml-context:Request>
<Extensions><sp:StickyPolicy
PolicyID="sticky-policy-2"
PolicyLanguage="PERMIS"
PolicyType="Authorisation"
TimeOfCreation="2010-08-09T00:00:00Z"
xmlns:sp="http://sec.cs.kent.ac.uk/stickypolicy">
<sp:PolicyAuthor>
<sp:AuthorType>DataSubject</sp:AuthorType>
</sp:PolicyAuthor>
<sp:PolicyResourceTypes>
<sp:ResourceType>personal:preferences</sp:ResourceType>
</sp:PolicyResourceTypes>
<sp:PolicyContents><X.509_PMI_RBAC_Policy OID="Policyv2">
  <SubjectPolicy>
    <SubjectDomainSpec ID="everywhere">
      <Include LDAPDN=""/>
    </SubjectDomainSpec>
  </SubjectPolicy>
  <RoleHierarchyPolicy>
    <RoleSpec Type="permisRole" OID="1.2.826.0.1.3344810.1.1.14">
      <SupRole Value="MyFriend"/>
    </RoleSpec>
  </RoleHierarchyPolicy>
  <SOAPolicy>
    <SOASpec ID="anyone" LDAPDN=""/>
  </SOAPolicy>
  <RoleAssignmentPolicy>
    <RoleAssignment>
      <SubjectDomain ID="everywhere"/>
      <RoleList>
        <Role Type="permisRole"/>
      </RoleList>
      <Delegate Depth="0"/>
      <SOA ID="anyone"/>
      <Validity/>
    </RoleAssignment>
  </RoleAssignmentPolicy>
  <TargetPolicy>
    <TargetDomainSpec ID="PersonalData">
      <Include URL="http://records.kent.ac.uk/PersonalData/"/>
```

```
    </TargetDomainSpec>
  </TargetPolicy>
  <ActionPolicy>
    <Action Name="Write" ID="Write">
     <TargetDomain ID="PersonalData"/>
    </Action>
  </ActionPolicy>
  <TargetAccessPolicy>
    <TargetAccess>
      <RoleList>
        <Role Type="permisRole" Value="MyFriend"/>
      </RoleList>
      <TargetList>
        <Target>
          <TargetDomain ID="PersonalData"/>
          <AllowedAction ID="Write"/>
        </Target>
      </TargetList>
     <Obligations>
          <Obligation ObligationId="SendE-mail" FulfillOn="Permit"></Obligation>
     </Obligations>
    </TargetAccess>
  </TargetAccessPolicy>
</X.509_PMI_RBAC_Policy>
</sp:PolicyContents>
</sp:StickyPolicy>
</Extensions>
</XACMLAuthzDecisionQuery>
</soapenv:Body>
</soapenv:Envelope>
```

Appendix 4.6.5 Request Context to TRANSFER sticky policy

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<XACMLAuthzDecisionQuery
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
ID="A2010-12-13T12.58.12"
Version="2.0"
IssueInstant="2010-12-13T12:58:12.209Z">
<xacml-context:Request xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oid:1.2.826.0.1.3344810.1.1.14"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>MyFriend</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>http://records.kent.ac.uk/PersonalData/.*</xacml-
context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="rid"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>rid-1</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
```

```
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>Write</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"/>
</xacml-context:Request>

</XACMLAuthzDecisionQuery>
</soapenv:Body>
</soapenv:Envelope>
```

Appendix 4.6.6 Request Context to TRANSFER sticky policy

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<XACMLAuthzDecisionQuery
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
ID="A2010-12-13T12.58.12"
Version="2.0"
IssueInstant="2010-12-13T12:58:12.209Z">
<xacml-context:Request xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="Role"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>member</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="rid"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>rid-1</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>TANSFER</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"/>
</xacml-context:Request>
</XACMLAuthzDecisionQuery>
</soapenv:Body>
</soapenv:Envelope>
```

Appendix 4.6.7 Response obtained with a Grant to TRANSFER sticky policy containing all the policies for "rid-1"

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
 <soapenv:Body>
 <urn:Response IssueInstant="2013-06-25T00:13:21.300+01:00"
ID="_d386a01909eafcc8c6d55e203a940aa5"
Version="2.0" InResponseTo="A2010-12-13T12.58.12"
```

```
xmlns:urn="urn:oasis:names:tc:SAML:2.0:protocol">
 <urn:Status>
  <urn:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
 </urn:Status>
 <urn1:Assertion IssueInstant="2013-06-25T00:13:21.300+01:00"
ID="_66e06da0a5776ed30f0ee540bce72c5b"
Version="2.0" xmlns:urn1="urn:oasis:names:tc:SAML:2.0:assertion">
 <urn1:Statement xsi:type="urn:XACMLAuthzDecisionStatementType"
xmlns:urn="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:assertion:cd-01"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <xacml-context:Response xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
  <xacml-context:Result ResourceId="ou=some,o=service,c=gb">
  <xacml-context:Decision>Permit</xacml-context:Decision>
  <xacml-context:Status>
  <xacml-context:StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
  </xacml-context:Status>
  <xacml:Obligations xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os">
   <xacml:Obligation
ObligationId="http://sec.cs.kent.ac.uk/obligations/stickypolicyobligation">
   <xacml:AttributeAssignment
AttributeId="http://sec.cs.kent.ac.uk/obligations/stickypolicyobligation/stickypolicy"
DataType="http://www.w3.org/2001/XMLSchema#string">
<![CDATA[<?xml version="1.0" encoding="UTF-8"?>
<sp:StickyPolicy xmlns:sp="http://sec.cs.kent.ac.uk/stickypolicy" PolicyID="sticky-policy-2"
PolicyLanguage="PERMIS" PolicyType="Authorisation" TimeOfCreation="2010-08-
09T01:00:00.000+01:00" >
<sp:PolicyAuthor><sp:AuthorType>DataSubject</sp:AuthorType></sp:PolicyAuthor>
<sp:PolicyResourceTypes><sp:ResourceType>personal:preferences</sp:ResourceType>
</sp:PolicyResourceTypes><sp:PolicyContents>
<X.509_PMI_RBAC_Policy
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
OID="Policyv2">
  <SubjectPolicy>
    <SubjectDomainSpec ID="everywhere">
      <Include LDAPDN=""/>
    </SubjectDomainSpec>
  </SubjectPolicy>
  <RoleHierarchyPolicy>
    <RoleSpec OID="1.2.826.0.1.3344810.1.1.14" Type="permisRole">
      <SupRole Value="MyFriend"/>
    </RoleSpec>
  </RoleHierarchyPolicy>
  <SOAPolicy>
    <SOASpec ID="anyone" LDAPDN=""/>
  </SOAPolicy>
  <RoleAssignmentPolicy>
    <RoleAssignment>
      <SubjectDomain ID="everywhere"/>
      <RoleList>
        <Role Type="permisRole"/>
      </RoleList>
      <Delegate Depth="0"/>
      <SOA ID="anyone"/>
      <Validity/>
    </RoleAssignment>
  </RoleAssignmentPolicy>
  <TargetPolicy>
    <TargetDomainSpec ID="PersonalData">
      <Include URL="http://records.kent.ac.uk/PersonalData/"/>
    </TargetDomainSpec>
  </TargetPolicy>
```

```
  <ActionPolicy>
    <Action ID="Write" Name="Write">
     <TargetDomain ID="PersonalData"/>
    </Action>
  </ActionPolicy>
  <TargetAccessPolicy>
    <TargetAccess>
      <RoleList>
        <Role Type="permisRole" Value="MyFriend"/>
      </RoleList>
      <TargetList>
        <Target>
          <TargetDomain ID="PersonalData"/>
          <AllowedAction ID="Write"/>
        </Target>
      </TargetList>
      <Obligations>
          <Obligation ObligationId="SendE-mail" FulfillOn="Permit"></Obligation>
     </Obligations>
    </TargetAccess>
  </TargetAccessPolicy>
</X.509_PMI_RBAC_Policy></sp:PolicyContents></sp:StickyPolicy>]]>
</xacml:AttributeAssignment>
<xacml:AttributeAssignment
AttributeId="http://sec.cs.kent.ac.uk/obligations/stickypolicyobligation/stickypolicy"
DataType="http://www.w3.org/2001/XMLSchema#string">
<![CDATA[<?xml version="1.0" encoding="UTF-8"?>
<sp:StickyPolicy xmlns:sp="http://sec.cs.kent.ac.uk/stickypolicy" PolicyID="sticky-policy-1"
PolicyLanguage="XACML" PolicyType="Authorisation" TimeOfCreation="2010-08-
09T01:00:00.000+01:00" >
<sp:PolicyAuthor><sp:AuthorType>DataSubject</sp:AuthorType>
</sp:PolicyAuthor><sp:PolicyResourceTypes>
<sp:ResourceType>personal:preferences</sp:ResourceType></sp:PolicyResourceTypes>
<sp:PolicyContents><PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:first-applicable"
PolicySetId="dspolicy"
xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
access_control-xacml-2.0-policy-schema-os.xsd">
<Target xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os">
</Target>
 <Policy PolicyId="Policy1"
 RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
<Target/>
<Rule Effect="Permit" RuleId="Mypolicy">
<Description>myfriend can Read </Description>
<Target>
</Target>
<Condition>
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">

<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
<SubjectAttributeDesignator AttributeId="Role"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">MyFriend</AttributeValue>
</Apply>
</Apply>
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
<ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
```

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Read</AttributeValue>
</Apply>
</Apply></Apply>
</Condition>
</Rule><Obligations>
        <Obligation ObligationId="LogTheRequest" FulfillOn="Permit"></Obligation>
    </Obligations>
</Policy>
</PolicySet></sp:PolicyContents></sp:StickyPolicy>]]>
</xacml:AttributeAssignment>
            </xacml:Obligation>
          </xacml:Obligations>
         </xacml-context:Result>
       </xacml-context:Response>
      </urn1:Statement>
    </urn1:Assertion>
   </urn:Response>
 </soapenv:Body>
</soapenv:Envelope>
```

Appendix 4.6.8 Response obtained with combined obligations

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
 <soapenv:Body>
   <urn:Response IssueInstant="2013-06-25T17:26:03.230+01:00"
ID="_411dbd7174a948240c26ee6864c9cf15" Version="2.0" InResponseTo="A2010-12-13T12.58.12"
xmlns:urn="urn:oasis:names:tc:SAML:2.0:protocol">
    <urn:Status>
      <urn:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </urn:Status>
    <urn1:Assertion IssueInstant="2013-06-25T17:26:03.230+01:00"
ID="_42c220ccd62fdbaf2ba46f95f5a92a71" Version="2.0"
xmlns:urn1="urn:oasis:names:tc:SAML:2.0:assertion">
      <urn1:Statement xsi:type="urn:XACMLAuthzDecisionStatementType"
xmlns:urn="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:assertion:cd-01"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
       <xacml-context:Response xmlns:xacml-
context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
         <xacml-context:Result ResourceId="ou=some,o=service,c=gb">
          <xacml-context:Decision>Permit</xacml-context:Decision>
          <xacml-context:Status>
            <xacml-context:StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
          </xacml-context:Status>
          <xacml:Obligations xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os">
            <xacml:Obligation ObligationId="LogTheRequest" FulfillOn="Permit"/>
            <xacml:Obligation ObligationId="SendE-mail" FulfillOn="Permit"/>
          </xacml:Obligations>
         </xacml-context:Result>
       </xacml-context:Response>
      </urn1:Statement>
    </urn1:Assertion>
   </urn:Response>
 </soapenv:Body>
</soapenv:Envelope>
```

Appendix 6: WSDL of ConVS

The wsdl of ConVS is presented here.

```
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:ns1="http://org.apache.axis2/xsd"
xmlns:ns="http://ws.apache.org/axis2" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:ax24="http://io.java/xsd" xmlns:ax21="http://sax.xml.org/xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
targetNamespace="http://ws.apache.org/axis2">
<wsdl:documentation>ConVS</wsdl:documentation>
<wsdl:types>
<xs:schema xmlns:ax26="http://io.java/xsd" xmlns:ax23="http://sax.xml.org/xsd" attributeFormDefault="qualified"
elementFormDefault="qualified" targetNamespace="http://ws.apache.org/axis2">
<xs:import namespace="http://sax.xml.org/xsd"/>
<xs:import namespace="http://io.java/xsd"/>
<xs:complexType name="Exception">
<xs:sequence>
<xs:element minOccurs="0" name="Exception" nillable="true" type="xs:anyType"/>
</xs:sequence>
</xs:complexType>
<xs:element name="ParserConfigurationException">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="ParserConfigurationException" nillable="true" type="xs:anyType"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="SAXException">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="SAXException" nillable="true" type="ax21:SAXException"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="IOException">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="IOException" nillable="true" type="ax24:IOException"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="IllegalArgumentException">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="IllegalArgumentException" nillable="true" type="xs:anyType"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="MarshalException">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="MarshalException" nillable="true" type="xs:anyType"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="XMLSignatureException">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="XMLSignatureException" nillable="true" type="xs:anyType"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="ValidateContract">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="args0" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="args1" nillable="true" type="xs:string"/>
```

```
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="ValidateContractResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="return" nillable="true" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
<xs:schema xmlns:ax25="http://ws.apache.org/axis2" attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://io.java/xsd">
<xs:import namespace="http://ws.apache.org/axis2"/>
<xs:complexType name="IOException">
<xs:complexContent>
<xs:extension base="ax25:Exception">
<xs:sequence/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:schema>
<xs:schema xmlns:ax22="http://ws.apache.org/axis2" attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://sax.xml.org/xsd">
<xs:import namespace="http://ws.apache.org/axis2"/>
<xs:complexType name="SAXException">
<xs:complexContent>
<xs:extension base="ax22:Exception">
<xs:sequence>
<xs:element minOccurs="0" name="cause" nillable="true" type="xs:anyType"/>
<xs:element minOccurs="0" name="exception" nillable="true"/>
<xs:element minOccurs="0" name="message" nillable="true" type="xs:string"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:schema>
</wsdl:types>
<wsdl:message name="ValidateContractRequest">
<wsdl:part name="parameters" element="ns:ValidateContract"/>
</wsdl:message>
<wsdl:message name="ValidateContractResponse">
<wsdl:part name="parameters" element="ns:ValidateContractResponse"/>
</wsdl:message>
<wsdl:message name="ParserConfigurationException">
<wsdl:part name="parameters" element="ns:ParserConfigurationException"/>
</wsdl:message>
<wsdl:message name="SAXException">
<wsdl:part name="parameters" element="ns:SAXException"/>
</wsdl:message>
<wsdl:message name="IOException">
<wsdl:part name="parameters" element="ns:IOException"/>
</wsdl:message>
<wsdl:message name="IllegalArgumentException">
<wsdl:part name="parameters" element="ns:IllegalArgumentException"/>
</wsdl:message>
<wsdl:message name="MarshalException">
<wsdl:part name="parameters" element="ns:MarshalException"/>
</wsdl:message>
<wsdl:message name="XMLSignatureException">
<wsdl:part name="parameters" element="ns:XMLSignatureException"/>
</wsdl:message>
<wsdl:portType name="ConVSPortType">
<wsdl:operation name="ValidateContract">
<wsdl:input message="ns:ValidateContractRequest" wsaw:Action="urn:ValidateContract"/>
<wsdl:output message="ns:ValidateContractResponse" wsaw:Action="urn:ValidateContractResponse"/>
<wsdl:fault message="ns:ParserConfigurationException" name="ParserConfigurationException"
```

```
wsaw:Action="urn:ValidateContractParserConfigurationException"/>
<wsdl:fault message="ns:SAXException" name="SAXException" wsaw:Action="urn:ValidateContractSAXException"/>
<wsdl:fault message="ns:IOException" name="IOException" wsaw:Action="urn:ValidateContractIOException"/>
<wsdl:fault message="ns:IllegalArgumentException" name="IllegalArgumentException"
wsaw:Action="urn:ValidateContractIllegalArgumentException"/>
<wsdl:fault message="ns:MarshalException" name="MarshalException" wsaw:Action="urn:ValidateContractMarshalException"/>
<wsdl:fault message="ns:XMLSignatureException" name="XMLSignatureException"
wsaw:Action="urn:ValidateContractXMLSignatureException"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ConVSSoap11Binding" type="ns:ConVSPortType">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
<wsdl:operation name="ValidateContract">
<soap:operation soapAction="urn:ValidateContract" style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
<wsdl:fault name="MarshalException">
<soap:fault use="literal" name="MarshalException"/>
</wsdl:fault>
<wsdl:fault name="IllegalArgumentException">
<soap:fault use="literal" name="IllegalArgumentException"/>
</wsdl:fault>
<wsdl:fault name="IOException">
<soap:fault use="literal" name="IOException"/>
</wsdl:fault>
<wsdl:fault name="SAXException">
<soap:fault use="literal" name="SAXException"/>
</wsdl:fault>
<wsdl:fault name="XMLSignatureException">
<soap:fault use="literal" name="XMLSignatureException"/>
</wsdl:fault>
<wsdl:fault name="ParserConfigurationException">
<soap:fault use="literal" name="ParserConfigurationException"/>
</wsdl:fault>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="ConVSSoap12Binding" type="ns:ConVSPortType">
<soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
<wsdl:operation name="ValidateContract">
<soap12:operation soapAction="urn:ValidateContract" style="document"/>
<wsdl:input>
<soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap12:body use="literal"/>
</wsdl:output>
<wsdl:fault name="MarshalException">
<soap12:fault use="literal" name="MarshalException"/>
</wsdl:fault>
<wsdl:fault name="IllegalArgumentException">
<soap12:fault use="literal" name="IllegalArgumentException"/>
</wsdl:fault>
<wsdl:fault name="IOException">
<soap12:fault use="literal" name="IOException"/>
</wsdl:fault>
<wsdl:fault name="SAXException">
<soap12:fault use="literal" name="SAXException"/>
</wsdl:fault>
<wsdl:fault name="XMLSignatureException">
<soap12:fault use="literal" name="XMLSignatureException"/>
</wsdl:fault>
<wsdl:fault name="ParserConfigurationException">
<soap12:fault use="literal" name="ParserConfigurationException"/>
```

```
</wsdl:fault>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="ConVSHttpBinding" type="ns:ConVSPortType">
<http:binding verb="POST"/>
<wsdl:operation name="ValidateContract">
<http:operation location="ConVS/ValidateContract"/>
<wsdl:input>
<mime:content type="text/xml" part="ValidateContract"/>
</wsdl:input>
<wsdl:output>
<mime:content type="text/xml" part="ValidateContract"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="ConVS">
<wsdl:port name="ConVSHttpSoap11Endpoint" binding="ns:ConVSSoap11Binding">
<soap:address location="http://localhost:8080/axis2/services/ConVS.ConVSHttpSoap11Endpoint/"/>
</wsdl:port>
<wsdl:port name="ConVSHttpSoap12Endpoint" binding="ns:ConVSSoap12Binding">
<soap12:address location="http://localhost:8080/axis2/services/ConVS.ConVSHttpSoap12Endpoint/"/>
</wsdl:port>
<wsdl:port name="ConVSHttpEndpoint" binding="ns:ConVSHttpBinding">
<http:address location="http://localhost:8080/axis2/services/ConVS.ConVSHttpEndpoint/"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

Appendix 7: PEP and ConVS Communication

The PHD PEP code that calls the ConVS and extends the request context based on the response obtained from ConVS is presented below.

```
<?php
$location="http://localhost:8080/axis2/services/ContractValidationService";
$uri="http://ws.apache.org/axis2";
$client = new SoapClient(null, array(
   'location' => urldecode($location),
   'uri'   => urldecode($uri),
   'trace'  => 1 ));
//getting the signature files
$str1="signature_final1.xml";
$str2="signature_final2.xml";


//calling ConVS functions

try {
  $result = $client->__soapCall("ValidateContract",
            array
            (
                new SoapParam($str1,"Signature_file1"),
                new SoapParam($str2,"Signature_file2")
            )
__
        );
} catch (Exception $e) {
```

```
  echo 'Caught exception: ', $e->getMessage(), "\n";
}


echo "I found this as a returned value.";
echo $result;
echo ".........";
$string = $result;
        //$string = 'NameOfSubjectOfContract: kaniz;AddressOfSubjectOfContract:
canterbury;ResourceType: cv of kaniz;NameOfPartyOne: O2;AddressOfPartyOne:canterbury;
NameOfPartyTwo: D; AddressOfPartyTwo: kent';
        //Splitting the attbutes and values. Attribute and values are separated by : and the
pairs are separated by ;.
        $array = preg_split("/[\s]*[;][\s]*/", $string);
        print_r($array);

//Constructing XACML elemnet from the result.
        $mkelement = "";
        for ($i=0; $i <= 6; $i++) {
                $element = preg_split("/[\s]*[:][\s]*/", $array[$i]);

                $mkelement .= '<xacml-context:Attribute AttributeId="' .      $element[0]. '"
DataType="http://www.w3.org/2001/XMLSchema#string">' . "\n" . '<xacml-context:AttributeValue>';
                $mkelement .= $element[1];
                $mkelement .= '</xacml-context:AttributeValue>'. "\n" . '</xacml-
context:Attribute>';

                echo $mkelement;
        }
//writing the result into a file which can be used for tesing purpose
        $fd = fopen("myfile5.xml", "w");
        fwrite($fd, $mkelement);
        fclose($fd);

//reading the file containing Request Context to put the XACML elements made from the response
from ConVS
        $contents = file("RequestByO2ExtendedMetaData.xml");
        $newcontents = "";
        foreach($contents as $line) {
                echo $line;
                echo "\r";

//extending the request context with the XACML elements made from the response from ConVS

if(rtrim(ltrim($line)) !="</xacml-context:Environment>") {

                        $newcontents .= $line;
                }
                else {

                        $newcontents .= $mkelement . "\n"; // don't add the newline if you've
already added it in $newthread before
                        $newcontents .= $line;
                }
        }

//Writing the extented request context to a file which can be sent to AuthorisationSystem
directly by PHP client code or SOAP UI client
        $fd = fopen("RequestByO2ExtendedContract.xml", "w");
        fwrite($fd, $newcontents);
```

```
        fclose($fd);
?>
```

The request context sent to the PEP before calling the ConVS is presented below

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<XACMLAuthzDecisionQuery
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
ID="A2010-12-13T12.58.12"
Version="2.0"
IssueInstant="2010-12-13T12:58:12.209Z">
<xacml-context:Request xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="Name"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>O2</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="Address"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>High Street, Canterbury, Kent</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="rid"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>rid-1234</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>READ</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="Contract-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>http://localhost/contract.xml</xacml-context:AttributeValue>
</xacml-context:Environment>
</xacml-context:Request>
</XACMLAuthzDecisionQuery>
</soapenv:Body>
</soapenv:Envelope>
```

The request context obtained from the PEP after calling the ConVS is presented below

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<XACMLAuthzDecisionQuery
xmlns="urn:oasis:names:tc:xacml:2.0:profile:saml2.0:v2:schema:protocol:cd-01"
ID="A2010-12-13T12.58.12"
Version="2.0"
```

```
IssueInstant="2010-12-13T12:58:12.209Z">
<xacml-context:Request xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Subject xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="Name"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>O2</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="Address"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>High Street, Canterbury, Kent</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Subject>
<xacml-context:Resource xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>ou=some,o=service,c=gb</xacml-context:AttributeValue>
</xacml-context:Attribute>
<xacml-context:Attribute AttributeId="rid"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>rid-1234</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Resource>
<xacml-context:Action xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>READ</xacml-context:AttributeValue>
</xacml-context:Attribute>
</xacml-context:Action>
<xacml-context:Environment xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
<xacml-context:Attribute AttributeId="NameOfSubjectOfContract"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>kaniz</xacml-context:AttributeValue>
</xacml-context:Attribute><xacml-context:Attribute AttributeId="AddressOfSubjectOfContract"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>canterbury</xacml-context:AttributeValue>
</xacml-context:Attribute><xacml-context:Attribute AttributeId="ResourceType"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>cv of kaniz</xacml-context:AttributeValue>
</xacml-context:Attribute><xacml-context:Attribute AttributeId="NameOfPartyOne"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>kaniz</xacml-context:AttributeValue>
</xacml-context:Attribute><xacml-context:Attribute AttributeId="AddressOfPartyOne"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>UKC, canterbury, Kent</xacml-context:AttributeValue>
</xacml-context:Attribute><xacml-context:Attribute AttributeId="NameOfPartyTwo"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>O2</xacml-context:AttributeValue>
</xacml-context:Attribute><xacml-context:Attribute AttributeId="AddressOfPartyTwo"
DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>High Street, Canterbury, Kent</xacml-context:AttributeValue>
</xacml-context:Attribute><xacml-context:Attribute AttributeId="Contract-
id"DataType="http://www.w3.org/2001/XMLSchema#string">
<xacml-context:AttributeValue>http://localhost/contract.xml</xacml-context:AttributeValue>
</xacml-context:Environment>
</xacml-context:Request>

</XACMLAuthzDecisionQuery>
</soapenv:Body>
</soapenv:Envelope>
```

Appendix 8: Automation of CNL to XACML

A portion of java code for XACMLConverter is given below.

```
package converter;

import java.io.File;
import java.io.IOException;
import java.util.Vector;

import javax.xml.bind.JAXBContext;
import javax.xml.bind.JAXBException;
import javax.xml.bind.Marshaller;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.xpath.XPathExpressionException;

import org.xml.sax.SAXException;

import converter.model.Apply;
import converter.model.Condition;
import converter.model.Obligations;
import converter.model.Policy;
import converter.model.Rule;

public class XMLConverter {

	public XMLConverter() {

	}

	public void convert(String inputFile, String convertedFile) throws ParserConfigurationException, SAXException,
IOException, XPathExpressionException, JAXBException {

		// create helper classes
		XMLInputHelper inputHelper = new XMLInputHelper(inputFile);
		XMLPolicyFactory factory = new XMLPolicyFactory();

		// start creating policy
		Policy policy = factory.createPolicy();

		// get ruleId
		String ruleId = inputHelper.getNodeValue("//rule-definition/rule-id/STRING/text()");
		System.out.println("RuleId = " + ruleId);

		Rule rule = factory.createRule();
		rule.setRuleId(ruleId);
		policy.setRule(rule);

		String grantOrDeny = inputHelper.getNodeValue("/rule-definition/rule-statement/GrantOrDeny/text()");
		assertTrue(grantOrDeny != null, "We always expect grant or deny");
		rule.setEffect(grantOrDeny);

		Condition condition = new Condition();
		rule.setCondition(condition);
```

```
		//all rules start with 'and'
		Apply conditionStartApply = factory.createApplyForFunction("and");
		condition.setApply(conditionStartApply);

		//now check if there is any 'operator'
		String[] operatorList = inputHelper.getNodeValueList("/rule-definition/rule-
statement/conditions/operator/text()");

		Apply operatorApply = null;
		if (operatorList.length > 0) {
			//we assume they are all 'or' or all 'and'... so a quick check on that!
			for (int i = 0; i < operatorList.length - 1; i++) {
				if ( ! operatorList[i].equalsIgnoreCase(operatorList[i + 1])) {
					this.notifyNotImplemented("We don't expect different kinds of operators.
We expected all operators to be same");
				}
			}
			operatorApply = factory.createApplyForFunction(operatorList[0].toLowerCase());
		} else {
			//if there is no opeator, it means we will add stuffs to first 'and'
			operatorApply = conditionStartApply;
		}

		boolean alreadyFoundNegativeRelOperator = false;//once we find a negative rel operator (like is-not-equal-
to), we expect all other rel operator will be negative as well.

		for (int iCondition = 0; iCondition < operatorList.length + 1; iCondition++) { // + 1 coz even if there is no
operator, it means there is one condition. for one 'or' there will be two coditions
			//now possible options:
			// 1) With 'relational operator'	: If the Subject:E-mail:string is equal to the
resource:DataSubject’sE-mail:string
			//		2) 'there is' booleanAttribute	: If there is a DataAccessManadate

			//Lets look for relational operator
			//we first need to build the xpath: if we are in the second part of the operator (such OR), we will
have more /conditions/ in xpath
			String conditionsChain = "";
			for (int iConditionBuilder = 0; iConditionBuilder < iCondition; iConditionBuilder++ ) {
				conditionsChain += "conditions/";
			}
			String relationalOperator = inputHelper.getNodeValue("/rule-definition/rule-
statement/conditions/" + conditionsChain + "/condition/relationalOperator/text()");
			if (relationalOperator != null) {
				String relOpFunctionName = "";
				if ("is equal to".equalsIgnoreCase(relationalOperator)) {
					if (alreadyFoundNegativeRelOperator) {
						throw new IllegalStateException("We already found a negative
relational operator, cannot apply " + relationalOperator);
					}
					relOpFunctionName = "string-at-least-one-member-of";
				} else if ("is less than".equalsIgnoreCase(relationalOperator)) {
					if (alreadyFoundNegativeRelOperator) {
						throw new IllegalStateException("We already found a negative
relational operator, cannot apply " + relationalOperator);
					}
					relOpFunctionName = "date-less-than";
				} else if ("is not the".equalsIgnoreCase(relationalOperator) ||"is not
a".equalsIgnoreCase(relationalOperator) ||
						"is not equal to".equalsIgnoreCase(relationalOperator) ||"is not
the".equalsIgnoreCase(relationalOperator)) {
					alreadyFoundNegativeRelOperator = true;
					relOpFunctionName = "string-at-least-one-member-of";
```

```
                                        } else {
                                                this.notifyNotImplemented("relational operator = " + relationalOperator);
                                        }
                                        Apply relationalOperatorApply =
factory.createApplyForFunction(relOpFunctionName);

                                        //now check if it contains attribute or value or reservedAttribute
                                        int relOperandCount = 2;//is-equal-to, is-not-the etc has 2 operators.
                                        for (int iRelOperand = 0; iRelOperand < relOperandCount; iRelOperand++) {
                                                boolean hasAttribute = inputHelper.getNodeValue("/rule-definition/rule-
statement/conditions/" + conditionsChain + "/condition/attributes[" + (iRelOperand + 1) + "]/attribute/category/text()") != null;
                                                boolean hasValue = inputHelper.getNodeValue("/rule-definition/rule-
statement/conditions/" + conditionsChain + "/condition/attributes[" + (iRelOperand + 1) + "]/values/value/STRING/text()") !=
null;
                                                if (hasAttribute) {
                                                        String firstAttrId = inputHelper.getNodeValue("/rule-
definition/rule-statement/conditions/" + conditionsChain + "condition/attributes[" + (iRelOperand + 1) +
"]/attribute/name/STRING/text()");
                                                        String firstCategoryId = inputHelper.getNodeValue("/rule-
definition/rule-statement/conditions/" + conditionsChain + "condition/attributes[" + (iRelOperand + 1) +
"]/attribute/category/text()");
                                                        String firstAttrType = inputHelper.getNodeValue("/rule-
definition/rule-statement/conditions/" + conditionsChain + "condition/attributes[" + (iRelOperand + 1) + "]/attribute/type/text()");

                                                        assertTrue(firstAttrId != null, "Condition must have name");
                                                        assertTrue(firstCategoryId != null, "Condition must have
category");
                                                        assertTrue(firstAttrType != null, "Condition must have subject
type");

                                                        factory.addAttributeDesignator(relationalOperatorApply,
firstCategoryId, firstAttrId, firstAttrType);
                                                } else if (hasValue) {
                                                        int valueCount = inputHelper.getNodeCount("/rule-
definition/rule-statement/conditions/" + conditionsChain + "/condition/attributes[" + (iRelOperand + 1) + "]/values/value");
                                                        Vector<String> valueList = new Vector<String>();
                                                        for (int iValue = 0; iValue < valueCount; iValue++) {
                                                                String value = inputHelper.getNodeValue("/rule-
definition/rule-statement/conditions/" + conditionsChain + "/condition/attributes/values/value[" + (iValue + 1) +
"]/STRING/text()");
                                                                valueList.add(value);
                                                        }
                                                        Apply valueBag = factory.createStringBagApply(valueList);
                                                        relationalOperatorApply.getApply().add(valueBag);
                                                } else {
                                                        this.notifyNotImplemented("only attribute / value ... not
reservedAttribute");
                                                }
                                        }

                                        operatorApply.getApply().add(relationalOperatorApply);


                                } else {
                                        //now we will look for 'there is'
                                        this.notifyNotImplemented("different relational operator in condition = " +
relationalOperator);
                                }

                        }//for loop for conditions


                        //lets get actions now
```

```
                int actionCount = inputHelper.getNodeCount("/rule-definition/rule-statement/actions/action");

                Vector<String> actionList = new Vector<String>();
                for (int iAction = 0; iAction < actionCount; iAction++) {
                        String actionName = inputHelper.getNodeValue("/rule-definition/rule-statement/actions/action["
+ (iAction + 1) + "]/word/text()");
                        actionList.add(actionName);
                }
                Apply actionApply = factory.createAction(actionList);

                //now finally let's handle operator apply (and special case with NOT)
                if (operatorApply != conditionStartApply) {
                        if (alreadyFoundNegativeRelOperator) {
                                Apply notApply = factory.createApplyForFunction("not");
                                notApply.getApply().add(operatorApply);
                                conditionStartApply.getApply().add(notApply);
                        } else {
                                conditionStartApply.getApply().add(operatorApply);
                        }
                }
                conditionStartApply.getApply().add(actionApply);

                //resourceType
                int resourceTypeCount = inputHelper.getNodeCount("/rule-definition/rule-statement/ResourceType");
                if (resourceTypeCount > 0) {
                        Vector<String> resourceTypeList = new Vector<String>();
                        for (int iResType = 0; iResType < resourceTypeCount; iResType++) {
                                String resourceTypeName = inputHelper.getNodeValue("/rule-definition/rule-
statement/ResourceType[" + (iResType + 1) + "]/word/text()");
                                resourceTypeList.add(resourceTypeName);
                        }
                        Apply resourceTypesApply = factory.createResourceTypes(resourceTypeList);
                        conditionStartApply.getApply().add(resourceTypesApply);
                }

                //obligations

                String obligation = inputHelper.getNodeValue("/rule-definition/rule-
statement/obligations/obligation/STRING/text()");
                if (obligation != null) {
                        Obligations obligations = factory.createObligationsWithId(obligation, rule.getEffect());
                        policy.setObligations(obligations);
                }

                // prepare marshaling to xml
                JAXBContext jaxbContext = JAXBContext.newInstance(Policy.class);
                Marshaller jaxbMarshaller = jaxbContext.createMarshaller();

                // output pretty printed
                jaxbMarshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);

                // write to file
                File file = new File(convertedFile);
                if (file.exists()) { //clear file
                        file.delete();
                }
                jaxbMarshaller.marshal(policy, file);

                // print to system.out as well
                System.out.println("============ " + convertedFile + " ============");
                jaxbMarshaller.marshal(policy, System.out);

        }
```

```
	private void notifyNotImplemented(String message) {
		throw new IllegalStateException("support for this case is not done yet: " + message);
	}

	private void assertTrue(boolean condition, String message) {
		if (condition == false) {
			throw new AssertionError(message);
		}
	}


	public static void main(String[] args) {

		try {
			XMLConverter converter = new XMLConverter();
			converter.convert("intermediate.xml", "policy.xml");
		} catch (Exception e) {
			e.printStackTrace();
		}

	}

}
```

Appendix 9: StickyPAD schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns="tas3:to:be:decided:namespace"
 targetNamespace="tas3:to:be:decided:namespace"
 xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
 xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
 <!-- Use a schema on local file store as there seem to be problems with the
ones available on the net. -->
<xs:import namespace="http://www.w3.org/2000/09/xmldsig#" schemaLocation="http://www.w3.org/TR/2002/REC-
xmldsig-core-20020212/xmldsig-core-schema.xsd"/>
<xs:element name="StickyPad" type="StickyPADType"/>
<xs:complexType name="StickyPADType">
 <xs:annotation>
  <xs:documentation>
  This is the TAS3 Sticky Policy and Data/resource type definition.
  Version 8. 2 December 2010.
  The DataResource can be any data or resource which requires a sticky policy.
  Resource Types holds the type(s) of resource that are contained
  in the DataResource e.g. it could be a computer system or an email message
  or some PII.
  Any number of policies can be stuck to a DataResource.
  The XML signature is optional because applications may choose to
  secure the PAD using alternate means, e.g. SSL/TLS.
  </xs:documentation>
 </xs:annotation>
 <xs:sequence>
		<xs:choice>
```

```
    <xs:element ref="DataResource"/>
    <xs:element ref="DataResourceRef"/>
   </xs:choice>
   <xs:element name="DataResourceType" type="ResourceTypes"/>
   <xs:element ref="StickyPolicy" maxOccurs="unbounded"/>
   <xs:element ref="ds:Signature" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ResourceTypes">
 <xs:sequence>
  <xs:element name="ResourceType" type="xs:anyURI" maxOccurs="unbounded"/>
 </xs:sequence>
</xs:complexType>

<xs:element name="StickyPolicy" type="StickyPolicyType"/>
<xs:complexType name="StickyPolicyType">
 <xs:annotation>
  <xs:documentation>
   The Policy ID specifies the globally unique ID of the policy.
   The PolicyLanguage specifies the language the policy is written in.
   The PolicyAuthor specifies attributes of the person who wrote the policy.
   Time of Creation specifies when the policy was written.
   Expiry time specifies after what time the policy should be ignored. Infinity is the default.
   Resource Type specifies the type(s) of resource this policy refers to.
   The PolicyContents contains the policy written in the language
   specified in PolicyLanguage.
   The PolicyType specifies what type of policy this is.
  </xs:documentation>
 </xs:annotation>
 <xs:sequence>
  <xs:element name="PolicyAuthor" type="PolicyAuthorType" form="qualified"/>
  <xs:element name="PolicyResourceTypes" type="ResourceTypes" />
  <xs:element ref="PolicyContents"/>
 </xs:sequence>
 <xs:attribute name="PolicyID" type="xs:anyURI" use="required"/>
 <xs:attribute name="PolicyLanguage" type="xs:anyURI" use="required"/>
 <xs:attribute name="PolicyType" type="xs:anyURI" use="required"/>
 <xs:attribute name="TimeOfCreation" type="xs:dateTime" use="required"/>
 <xs:attribute name="ExpiryTime" type="xs:dateTime" use="optional"/>
</xs:complexType>

<xs:element name="PolicyContents" type="AnyXMLType"/>
<xs:element name="DataResource" type="AnyXMLType"/>
<xs:element name="DataResourceRef" type="xs:anyURI"/>

<xs:complexType name="AnyXMLType" mixed="true">
 <xs:sequence>
  <xs:any minOccurs="0" maxOccurs="unbounded" namespace="##any" processContents="lax">
   <xs:annotation>
    <xs:documentation>
     Any xml content is allowed in this element.
    </xs:documentation>
   </xs:annotation>
  </xs:any>
 </xs:sequence>
</xs:complexType>

<xs:complexType name="PolicyAuthorType">
```

```
            <xs:annotation>
                        <xs:documentation>
                         The Author is identified by any number of Author Attributes.
                         AuthorType indicates the author's relationship to the Resource in this StickyPAD
                        </xs:documentation>
            </xs:annotation>
   <xs:sequence>
    <xs:element name="AuthorAttribute" type="AuthorAttributeType" minOccurs="1" maxOccurs="unbounded"/>
    <xs:element name="AuthorType" type="xs:anyURI"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="AuthorAttributeType">
   <xs:attribute name="AttributeId" type="xs:anyURI" use="required"/>
   <xs:attribute name="Issuer" type="xs:anyURI" use="optional"/>
   <xs:attribute name="IssueInstant" type="xs:dateTime" use="optional"/>
   <xs:attribute name="Value" type="xs:string" use="required"/>
</xs:complexType>

</xs:schema>
```