SEVENTH FRAMEWORK PROGRAMME

ENDORSE

Legal Technical Framework for Privacy Preserving Data Management

ENDORSE

Deliverable D3.2

Privacy Rule Definition Language - Preliminary Specification

Editor:	Thomas Kurz, Christoph Rücker, Thomas Lampoltshammer, Thomas Heistracher
Deliverable nature:	Report
Dissemination level: (Confidentiality)	Public
Contractual delivery date:	01.09.11
Actual delivery date:	06.09.11
Suggested readers:	Consortium
Version:	9.0 (06. September 2011)
Total number of pages:	63
Keywords:	Rule Language, PRDL, Rule Scenarios

Abstract

This deliverable presents PRDL (Privacy Rule Definition Language) and its evolvement from a language defined by the elements and constructs needed to represent privacy policies in a computerized way into a XML-based language which can be written using syntax templates and which can be parsed into an existing rule language. Related legal XML languages are presented and discussed in brief before the complexity of the problem space is dealt with. Consequently, a glossary that covers all the important expressions is developed, it is enriched with annotations and amendments related to the involvement of contributing project partners. In the following, the meta models representing the language are explained in detail. Finally, this deliverable includes 'divers examples' of rules that were written basing on the syntax templates provided here as well as the according XML and Drools representations.

Disclaimer

Impressum

Legal Technical Framework for Privacy Preserving Data Management

ENDORSE

WP3 Architecture

D3.2 Privacy Rule Definition Language Preliminary Specification

Editor: Thomas Kurz, SUAS, Christoph Rücker, SUAS, Thomas Lampoltshammer, SUAS, Thomas Heistracher, SUAS

Work-package leader: Pierfranco Ferronato, Soluta

Estimation of PM spent on the Deliverable: 4

Executive summary

The definition of a *Privacy Rule Definition Language* (PRDL) is one of the main goals of the ENDORSE project. There have been many initiatives for expressing privacy rules and legal restrictions into a computable way. Some of them are described and evaluated in this deliverable. The attempt of PRDL and consequently of this deliverable is to present a collaborative result towards a multi-stakeholder language. PRDL should be sufficiently expressive to define privacy policies for SMEs, it should link the wording of the data privacy laws of different European countries, it should be represented in natural language and therefore should be easy to understand. After all, it should be automatically or semiautomatically executable by a rule engine. All these requirements, documented in D2.4, *Language Requirements Specification*, reflect the potentially broad application area of such a PRDL, but also show the challenges we are facing. The preliminary PRDL specification deliverable at this early stage of the ENDORSE project should catalyse the interdisciplinary discussions within and outside of ENDORSE and act as a first basis for feedback and adaptations. This early prototyping includes a basic language definition, the implementation of an editor as well as a first implementation in the Drools rule engine.

After the introduction, we start with an evaluation and short description of XML-based languages, which seem to be suitable to fulfil the requirements described in D2.4. The focus here was on open-source languages and toolsets rather than proprietary software solutions. The evaluation of these languages lead into a comparison of their benefits and goals with benefits and goals of the foreseen PRDL. Chapter four of this document is the core part of the current PRDL specification – the meta model and glossary. Beside the meta rule model, which defines the concepts and objects of the PRDL syntax, a so-called meta access model was defined to identify the main stakeholder objects and their relationship in a data access use case. The two meta models as well as the glossary are the current version and will be modified continuously by project partners in an online version at http://ict-endorse.eu/wiki/index.php/PRDL and http://ict-endorse.eu/wiki/index.php/Glossary. Based on these language elements, chapter five outlines the first syntax templates together with the first examples of possible PRDL statements in the Drools rule language. This mapping supports an easier translation into computable rules later on in the project. An outlook to the planned activities of PRDL development concludes the preliminary specification. The appendix includes the current PRDL XSD schema as well as a short rule analysis from D2.1 and the privacy statement of Docticare which is to be represented by PRDL rules.

List of authors

Participant	Author
SUAS	Thomas Kurz
SUAS	Christoph Rücker
SUAS	Thomas Lampoltshammer
SUAS	Thomas Heistracher
Internal Review	**Author**
WIT	Jason Finnegan
Uni Zaragoza	Pedro Bueso
Acknowledgements	**Author**
WIT	Mark McLaughlin

Table of Contents

1. Introduction and Context

This deliverable describes the Privacy Rule Definition Language (PRDL) preliminary specification in the context of the ENDORSE project. The aim of the ENDORSE project is to provide interconnection between policy makers, the organisations which handle the data and last but not least the data subjects via pioneered conceptions regarding legally compliant data handling. In order to achieve this goal ENDORSE will introduce a PRDL that provides solutions for organisations that want to migrate their existing privacy policies into computationally executable rules enforced by the ENDORSE framework.

As already stated in D2.4, the focus of the language development efforts lays on the two deliverables, namely D3.2 PRDL preliminary specification as well as D4.2 Rule Engine preliminary implementation. The document at hand is the former mentioned D3.2 built upon the Language Requirements specification of D2.4. As it turned out after analysing the results of D2.4, the most suitable way to express the language is via the use of a XML-based language. Therefore, this deliverable is examining the state-of-the-art XML-based access rule languages to identify possible candidates in order to adapt their specifications insofar as they are matching the language requirements specification. Besides the specifications concerning the technical implementation, also the meta-models which will be used to map personal data and the related law-based access rules are introduced and their elements are described by the presented glossary. Along with the meta-models, syntax templates are presented and extended by a set of rules given in XML.

The remainder of this document is structured as follows. Chapter 2 introduces several XML-based languages as candidates for PRDL. These candidates are described and unique approaches and features are emphasized. Chapter 3 then summarizes and discusses benefits and drawbacks of the candidates regarding the categories language, system, approach and security issues. Chapter 4 is split in two parts. The first one contains the explanation of the preliminary meta-models of the PRDL while the second part is dedicated to the description of the meta rule model, including all necessary elements in order to represent a rule for handling data requests. The findings on which these models are based on were derived from the scenario deliverables D2.1 and D2.3 as well as the language specification D.2.4. In Chapter 5 the general syntax of the PRDL is delineated so that rule statements regarding privacy and data protection can be handled properly. It will be discussed what suitable conceptualisation could be pursued in order to achieve the essential requirements of PRDL as there are ambiguity, traceability and accountability, taking into consideration the evolutionary state of law.

2. Comparison of XML-Based Languages

In this chapter, a comparison and brief discussion of XML-based languages is given. This is done to provide an overview of existing approaches and to focus on some key characteristics that are helpful for the orientation of the readers and the understanding of the chapters following.

2.1. Author-X (X-Author)

Author-X is Java-based and dedicated to access control and security policy for XML documents. Regarding access control Author-X provides specification capabilities on variable detail levels. Furthermore, "push and pull" functionalities for document release are included as well. In addition to that, it is possible to update documents in a distributed environment via hash functions and digital signatures. Besides, tools are delivered to support security government related to underlying policies [1,2,3].

2.2. FASTER

There exist relations in some points between Author-X and *Faster*, for instance features like the possibility of defining a tiered access control schema, limit to XML sources as well as black and white lists regarding authorizations. Due to the fixed hierarchy used for representation of security requirements regarding managed content it becomes extremely complicated to incorporate new resources frequently. Besides, the before mentioned security requirements may change over time. Also, the system is designed to be server-sided which can be identified as a drawback as well. Moreover, there is no support for content protection except limited user views on the content on the server [1, 4, 5].

2.3. XrMl

The *eXtensible rights Markup Language* (XrMl) is dedicated to digital rights specification. Due to its complexity and specificity it is hard to use it in other scenarios which demand simple specifications. Therefore, its accuracy leads to inflexibility. For expressing rights concerning digital resources, XrML supports different levels of trust regarding resources to be protected as well as identification via Universal Description, Discovery and Integration (UDDI) and digital signatures. The major concepts of XrML are license, grant, principal, right, resource and condition and can be extended to provide additional requirements of specific projects. As mentioned before, the high detail regarding the used security approach confines the language's applications. The model of the language is based on the ID of the user which means that usage without registration or/and identification is not possible [1, 6].

2.4. ODRL

The *Open Digital Rights Language* (ODRL) is built on XML-Schema - however no instrument for management is delivered. ODRL provides DRM-related semantics for open and trusted surroundings, though it does not directly provide backing for security mechanisms. While focusing on providing a digital version of rights management, at the same time it also offers an extendable set of services related to the digital Web milieu.

The distinct vocabulary delivers possible expressions regarding terms and conditions related to digital as well as physical objects. ORDL gives its users the opportunity to express who possess the right, the kind of allowed using scenarios as well as offers and contracts related to the items. The included basis model of the ODRL description includes models for permission, restriction, requirements, conditions, rights holders, context, offer, contract, revocation and security. However, it does not include capacities and necessities regarding content protection, physical or digital distribution, as well as payment related issues [1, 7].

However, the objectives of this language definition deviate much from the objectives of the other candidates and therefore it will not be considered in further comparisons.

2.5. XACL

XML Access Control Language (XACL) was one of the first projects to establish a language for access policies based on XML. Its developer, IBM, built it upon four basic control access concepts:

- object: resource to control access
- subject: entity which requests the access
- action: applied on the object (read, write, create, delete, and possible extensions)
- condition: must be satisfied for the subject to execute action over the object

The purpose of the included structure is to governor the access to XML documents. The difficulty of adapting the used structure to other settings is caused by the language being based on DTDs and thus limited in terms of expressiveness. XACL delivers a provisional authorization model which can be used to specify provisional actions associated with an act like read, write, create or delete. In contrast to other access control and authorization systems, XACL does not answer request with simple "yes" or "no" statements but provides information which actions have to be performed in order to achieve access. The same is valid the other way round. These actions are called "provisional actions". Examples therefore can be found in the areas of auditing, encryption or XSL transformations besides standard actions like write, create or delete.

The XACL architecture can be split in two units. The first one is dedicated to the evaluation of the access decision while the second one focuses on the execution of the request itself and is responsible for the execution of the provisional actions mentioned before. It can be stated that the provisional model features more detail in expressiveness and flexibility, but it is also limited in that way that it is not suitable for distributed environments due to its centralized policy model [1, 11].

2.6. SAML

Security Assertion Markup Language (SAML) is a suggestion of OASIS including an assertion language as well as a protocol to describe, demand and send verification and permission data between security domains. The major aim of this proposal is to endorse interoperability for diverse security systems in form of a XML-based framework regarding e-business. A key aspect of SAML is the use in SSO environments, enabling the user to login in one domain and use resources in other domains as well, without the need for re-authenticating. SAML supports users by providing three different kinds of assertion statements as there are "authentication", "authorization decision" and "attribute".

An expression in SAML consists of a set of assertions related to a certain subject. A subject can be described as an entity be it computer or human which claims an identity registered in a security domain. For instance, a person can be identified via her email address within a DNS domain. The assertions are represented in XML including a nested structure. This means that a single assertion may hold several inner statements regarding authentication, authorization and attributed.

In SAML it is possible to request assertions from SAML authorities via a protocol which is based on XML requests and response messages. It is also possible to bind these messages to different underlying transport protocols. One example would be SOAP over HTTP.

Overall, this project is, in the authors' opinion, rather an extension to a given XML framework like XACML than a standalone construct suitable for our purposes [1, 12].

2.7. XACML

OASIS eXtensible Access Control Markup Language (XACML) originates from (as does SAML) an initiative of OASIS which is dedicated to formulate access policies in a XML-based structure. XACML targets detailed control of authorized actions via the introduction of activity classes, characterization of the requesting user, providing the used protocol as well as content introspection. In addition to that, it features a policy authorization model in order to guide designers along the process of developing their own model. The syntax of the specification consists out of a tuple including {subject, object, action}. The subject element is used to represent users, groups and roles in general. The object element is dedicated to provide the detail

factor inside the XML documents while the action element is separated in four types as there are read, write, create and delete. Furthermore, the concept of provisional authorization is adopted as well.

One point of discussion is that the design is assuming that every document to be protected is a XML document, which results in problems of the environment is a non-XML environment. In addition to that, the assignment of privileges can only be achieved by using the role as subject, not the user herself directly. The XACML schema can be described in three models as there are the data-flow model, the policy language model and the administrative model.

The core schema can be extended to implement new features and the Policy decision point (PDP) in the policy language model can be adapted to represent a totally different evaluation approach and its related decision making process. Due to the architecture used every PDP input and output has to be SAML-compliant which in return enables the redistribution without any additional security measures in terms of completion (specification of SAML) [1, 13].

2.8. SecPAL

The main aim of the *Security Policy Assertion Language* (SecPAL) project is set on developing a distributed solution regarding the expression of authorization policies. At the same time, it is dedicated to do further research in the field of language design and semantics as well as algorithms and analysis techniques. The project itself is partnership amongst the advanced technology incubation group of Microsoft's Chief Research and Strategy Officer and Microsoft Research Cambridge.

The language focuses on the following five areas:

- Expressiveness – via flexible delegation of authority by the use of the primitive “can say” in combination with support for domain-specific constraints as well as a separation between queries and assertions while also allowing negations in queries without permitting them inside of assertions.
- Clear, readable syntax – SecPal features a concrete syntax based on simple statements close to natural language and in addition provides a XML schema for exchange purposes.
- Succinct, unambiguous semantics – due to ambiguity issues in other languages, SecPal defines three deduction rules for specifying the meaning of assertions directly, independently of any other logic.
- Effective decision procedures – “the language’s query evaluation is decidable and tractable by translation into Datalog with constraints.”
- Extensibility – it is possible to extend the language in a modular or orthogonal way by for example parameterized verbs, additional environment functions and the extension of language constraints by the user herself. Furthermore, it features a PKI-based, SOPA-encoded infrastructure combined with a policy-editing tool and the possibility of authorization queries with C#.

Unfortunately, the usage in Endorse is not possible due to the restrict limitations included in the license agreement [8, 9].

2.9. SPL

The *Semantic Policy Language* (SPL) is a XML-based language dedicated to specify the access control policies via a combination of the semantic properties of resources to be accessed, the external authorization entity as well as the context. In order to reduce the complexity introduced by the definition process of access control policies, SPL uses concepts known from programming as there are modularity, parameterization and abstraction.

The modular structure of the policies’ definition can be described as followed:

1. Separation of the specification into the sections access control criteria, the allocation of policies according to their resources and semantic information as well.
2. Abstraction of the access control related components

3. Reusability of access control components

It is also possible to embrace contextual considerations in a transparent way by the help of semantic information, while at the same time support the task of semantic validation. Despite methods in other languages SPL uses a separate specification called Policy Application Specification (PAS) in order to store references to the target object. The “relating”- action is performed dynamically when a request is received. Policies and PAS allow parametrization empowering SPL to produce general and flexible policies. Furthermore, new policies can be created by importing components of other policies without generating ambiguity. Hereby, it is possible to state the abstract meaning of elements included in the policies. In addition to that, the schema for SPL is described as a set of XML schema templates, helping to create specifications and heavily support automated syntactic validation [1, 14].

This chapter elicited possible PRDL candidates. All of them are based on XML which represents the standard format for data representation and exchange. In the upcoming part the languages will be discussed in detail and they will be compared regarding their features in language characteristics, system integration aspects, approach regarding access control and security aspects. As basis for the discussion the agreed goals from D2.4 [10] will be used. After this analysis it should be possible to identify the suitable candidate to use for the further progress in the project.

3. Evaluation and PRDL Goals

After the introduction of XML-based framework candidates, the following section will summarize and discuss features of the candidates regarding language characteristics, system integration aspects, approach regarding access control and security aspects. Combining the intended goals from D2.4 [10] and [1] it can be stated that a suitable solution SHOULD include several of the following properties:

- Simplicity
- Flexibility
- Expressiveness
- Modularity
- Scalability
- Interoperability
- Extensibility
- Lack of ambiguity
- Open access control scheme
- Integration with external authorization schemes
- Access based on contents
- Integrated solution
- Provisional authorization
- Temporary authorization
- Distributed execution of policies
- Mechanisms for secure distribution of contents

The upcoming tabular description was taken from a survey [1]:

LANGUAGE					
	X-Author	FASTER	XACL	XACML	SPL
Policy Specification Method	Credential-based.	Based on RBAC and hierarchies.	RBAC.	Based on identity and credentials.	Based on attribute certificates
Syntax	DTD	XML Schema	DTD	XML Schema	XML Schema
Complexity Level	Low	Medium	Low	High	Low
Expressiveness	Medium	Medium	Low	High	High
Ambiguity	Possible because of positive and negative authorizations	Possible because of positive and negative authorizations	Possible because of positive and negative authorizations	Possible. Requires the PMP to resolve conflicts	NO
Modular Language	NO	NO	NO	YES	YES. Modular policies and with parameters. They can also be composed without ambiguity.
Semantic Validation	NO	NO	NO	NO	YES. Automatic detection of inconsistencies and errors based on the semantic information about the context, the resource to be accessed and the authorization entities.
Content-based Access	NO. Based on Structure	NO. Based on Structure	NO.	Dependent on Implementation.	YES. At the semantic level

	(DTDs).	(XML Schemas).			(metadata).
Scalability	Low. Based on subscriptions. Credentials registered locally.	Medium. Based on certificates but it requires subscription (for the identification).	Low. Centralized.	Dependent on the Implementation.	High. Fully distributed scheme and certificate based. No subscription is required.
Interoperability Level	Low. Federated Sources	Low	Null	Medium-High (Not sufficiently specified) Based on SAML assertions.	High. Integration of Privilege Management Infrastructure based on metadata about the Source Of Authorizations.
Policies can be modified in a dynamic and transparent way	NO	YES (centralized)	YES (centralized)	NO	YES

APPROACH					
	X-Author	FASTER	XACL	XACML	SPL
Generality	Specific Purpose	Specific Purpose	Specific Purpose	Specific Purpose	General Purpose
Access Control Scheme	Identification. Language based on DTDs to express credentials and its types.	Identification. Language based on XML-Schema for the expression of identity and attribute certificates.	Identification. Language based on DTDs to express RBAC elements (iduser, group, role).	Identification. Language based on XML-Schema for the expression of identity and conditions about the attribute certificates, the resource and the environment.	Attributes. Language based on XML-Schema for the specification of conditions related to the attribute certificates, the resource and the context. Complemented with other semantic components of the language (PAS, SRR, SOADs)

SYSTEM					
	X-Author	FASTER	XACL	XACML	SPL
Dependency of the Language	YES	YES	YES	Dependent of the implementation	NO It could use any language that becomes the standard.
Application Scope	XML documents (valid respect to a DTD or simply web formed)	XML documents	XML documents	Resources identifiable through a URI (anyURI)	Not restricted: Software Objects (distributed objects, Web services, applets, servlets...) Data Objects (without format restriction: multimedia objects, forms, XML, ...)
Integration with external authorization mechanisms	NO	Possible. Not defined.	NO	Possible. Uses SAML.	Complete. X.509 Standard and semantic information.

SECURITY					
	X-Author	FASTER	XACL	XACML	SPL
Secure Distribution	YES Passive Containers. Problems with key Management.	NO	NO	NO	YES Active Containers.
Distributed Mechanism for Policy Execution	NO Centralized Execution	NO Centralized Execution	NO Centralized Execution	NO Centralized Execution	YES Based on Active Containers
Provisional Authorization	NO	NO	YES	YES	YES
Temporary Authorization	NO	NO	NO	NO	YES

After consideration of all listed aspects of the former described candidates it can be stated that XACML and SPL are the best fitting proposals. In direct evaluation, XACML has some drawbacks which, however, could be solved or circumvented. One of the major aspects here is in the security area and distribution. This could be resolved by the use of XML databases for secure storage and distribution as they were described in [10]. Another aspect is the validation of semantics which could be addressed by the editor with the help of drop-down fields providing correct syntax and ensure correct semantic.

In summary the XACML solution prevails due to the demands to the PRDL formulated in [10] regarding the prioritisation of the stakeholders and their goals. There it is stated that standardization has a very high priority together with interoperability. Due to the fact that XACML is an international accepted industrial standard, the demand could be matched best with XACML as PRDL candidate.

4. PRDL Meta-Model and Glossary

The following section explains the preliminary meta-models of the PRDL language and the meta-model on how access will be handled in the ENDORSE system. As a first step, the scenarios given in D2.1 and D.2.3 were analysed and serve as a basis for the meta-models. Although the deliverable mainly deals with preliminary PRDL specification and the according meta-model, the meta access model has been created to take a look on how request for data will be made within the ENDORSE system. Additionally it should serve as a first step towards the ENDORSE component architecture. As the definitions for the different parts of the meta-models are given in Section 4.3, the following sections focus on the interrelation and functional binding of the constructs.

4.1. PRDL Meta Access Model

In a nutshell, the meta access model, shown in Figures 1 (UML) and 2 (more descriptive), describe how a request to the ENDORSE system has to be composed in order to enable the rules developed with PRDL evaluate the request. As this is mainly an architecture issue, this model should provide help and is a summary of what occurred during the PRDL development. The model is still in an preliminary state and under development as it has to be adapted to the ENDORSE architecture that will be drafted in D3.1. Nevertheless the model will be used for showing the interaction between the PRDL rules and a request within the preliminary implementation of the rule engine in D.4.3. In the following the elements of the model are described especially emphasizing on element interactions.

Request

The root element of the meta access model is the *Request*. A request contains the three elements which are the *RequestPurpose*, the requested *Data* and the reference to the *DataController* who executes the request. It also includes the *DataSubject* implicitly as it is part of the *Data* element. The rule engine will have to handle this request and either grant or deny the access by evaluating the request against the stored PRDL rules. (e.g. A system administrator requests access to the telephone numbers of all customers for direct marketing.) The Request element may be enhanced with elements that implement the duration of validity of the request etc.

RequestPurpose

The introduction of a second type of purpose, the *RequestPurpose*, is needed to distinguish form the *Purpose* for that *Consent* has been given by a *DataSubject*. A *RequestPurpose* can also be legitimated by legal obligations and others. A *DataController* always has to justify a request for data by adding a purpose. This purpose can be completely opposite to the one that the data subject has a agreed on freely. The legitimation for a request purpose may be a given consent from the data subject but there can be various others including legal obligations etc.. A *Request* can include more than one *RequestPurposes* but has to have at least one. (e.g direct marketing, promotion, customer relationship management, etc.)

Data

The second element included into a request to the ENDORSE system is the *Data* element. This element represents the data elements which the *DataController* wants to use for a specific purpose. It is composed out of a *Consent*, *Constraint, Purpose*, *ProcessingGround* and a list of *DataObjects*. As the access or use of data may be granted by the *DataSubject* by for example given explicit consent the Data has a *Consent* object assigned. When there are explicit reasons for denying specific actions with data these are represented by the *Constraint* element. Data is additionally a container for a subset of *DataObjects* that describe the data in greater detail. A Data element must have at least one *DataObject* and a *Consent* and can have one or more *Constraints*. (e.g contact information of customer, medical information of customer, etc.)

DataObject

Includes all the objects that define the personal data within the *Data* field. The data object is the atomic representation of a part of personal data and It represents the smallest part of information within a *Data* element. (e.g. telephone number, first name, last name, etc.). The *Data* element can consist of numerous of *DataObject* elements.

Constraint

In case of special circumstances, that prevent certain access or actions on the *Data* element, a *Constraint* element is defined. In order to get a granted access to the *Data* the request has to meet the preconditions of the constraints to get access (e.g. The *DataController* has to have an approval of the *DataProcessor* in charge to process data).

Consent

The *Consent* element in every *Data* element grants that the *DataSubject* has given his consent. It includes one or a number of purposes that has been granted by the DataSubject. The distinction between this *Purpose* and the *RequestPurpose* is important to mention here to avoid ambiguity. (e.g. The user has given consent to the storage of his personal data for services provision.) As purpose is not the only element that can legitimate the processing of data the ProcessingGround element is introduced.

Purpose

Purpose is the reason why data is collected and processed. For example the purpose for maintaining customer records is 'performing contractual obligations', such as shipping goods to the correct address, 'marketing', etc. Some purposes have DS's consent, many do not. For example as part of a Consent, the Purpose can represent the action to be performed on Data that the DataSubject has granted. A Consent element can include more Purpose elements. (e.g. for marketing purpose, for third party selling, etc.).

ProcessingGround

An organization can process data for many reasons, which can be for example legal obligations or operative reasons. The PRDL rule has to specify the precise reason for what data access is granted. The consent element is in charge of specifying the consent given by the user but the explicit reason for data processing can be divers. To model these various reasons the ProcessingGround element was introduced into the PRDL meta rule model.

DataRole

There are three actor elements in the meta access model. These are the *DataSubject*, *DataProcessor* and *DataController*. They are all related to data and therefore the *DataRole* super element was created to interlink these elements.

DataSubject

The *DataSubject* has one or many *DataElements* assigned to it. The *DataSubject* does not have another operative role within the meta access model but has to be a part of it to accomplish completeness. (e.g User John Doe, etc.)

DataController

As a central actor, the *DataController* creates a request for Data. Additionally every *DataController* has a *DataProcessor* assigned which is responsible for the executed actions on the Data. If a for example a constraint requires additional authentication the *DataProcessor* is the responsible element. (e.g Jane Doe from the Marketing department)

DataProcessor

The *DataProcessor* element is responsible for a *DataController* that executes actions on *Data*. If there is an incident and a *DataController* does not know how to behave the *DataProcessor* assists. This element is also included to assure completeness and to model a company that processes data better.

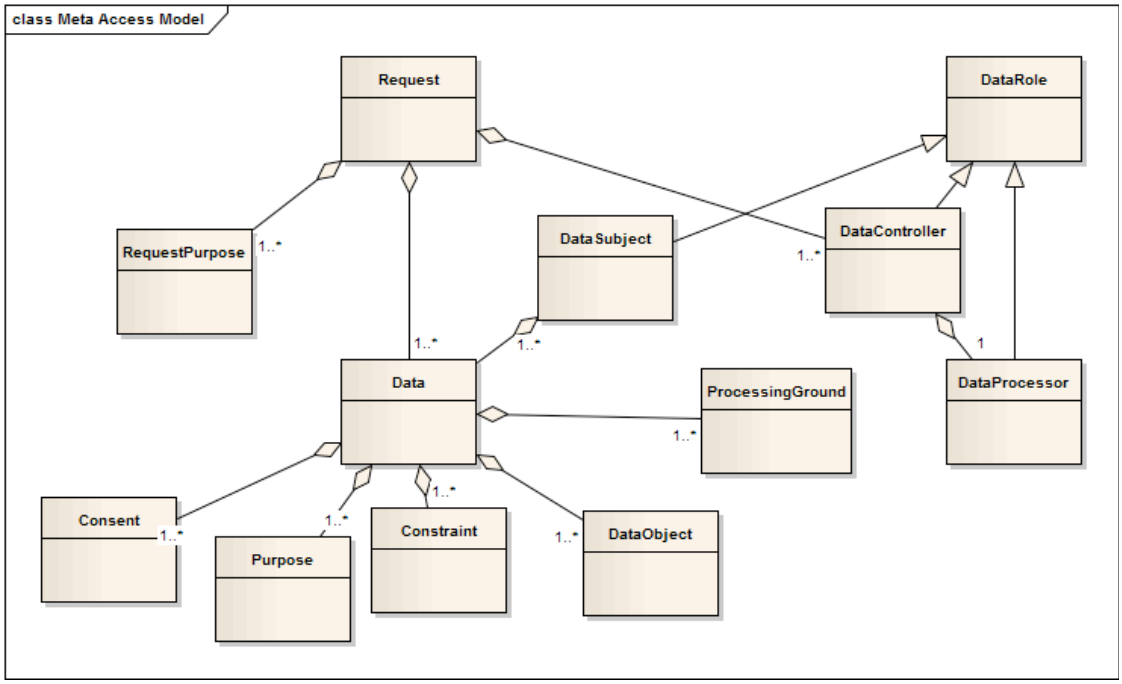

Figure 1: Meta Access Model UML (http://ict-endorse.eu/wiki/index.php/PRDL)

4.2. PRDL Meta Rule Model

The meta rule model reflects the current status of the PRDL development and is illustrated in Figures 4 (UML) and 3 (more descriptive). It exhibits all the relevant elements needed for representing a rule that can handle requests for personal data in the context of a data processing company. This model was created after an analysis of the scenario deliverables (D2.1 and D2.3) and the language requirements specification D2.4. As most of the elements are described in section 4.3, this section focusses on the ones that are not described in great detail the Glossary. These elements are mainly needed for administrating the rules and therefore are not included in the Glossary. During the development process it was decided to create a rule language based on syntax templates (explained in Section 5) and further a generic XML language that is able to represent these rules and additionally is parseable into an existing rule language (e.g. Drools[1]). The preliminary meta rule model has been the basis for all those developments.

[1]http://www.jboss.org/drools

Examples for rules and their according XML and Drools representation are given in Section 5. It has to be noted that the meta rule model is preliminary and adoptions will occur to handle requirements that appear in the future. In order to assure that the model is completely documented the missing elements are described in the following:

PRDLPolicySet

The *PRDLPolicySet* is the root element in the PRDL model and is responsible for providing a container to the PRDL policies. The policy set can be a set of company specific policies or can be a set of policies representing e.g. national implementations of the European Data Protection Directive. Examples for rules with the complete PRDL XML rule structure will be given in Section 5. A PRDLPolicySet can include one ore many PRDLPolicies. The basic principle behind the policy set is taken from XACML [13].

PRDLPolicy

The *PRDLPolicy* element includes the actual PRDL rules and represents the second container within the PRDL language. A policy can include rules that are related to specific data or specific procedures within a company. This element enables the user to structure the rules fine-grained. A *PRDLPolicy* can contain one or more *PRDLRules*.

PRDLRule

The PRDLRule element is the root element of the rule model. All the rule relevant elements are contained within the PRDLRule which are the StaticRuleAttributes and the DynamicRuleAttributes. These elements will be explained in greater detail in the related sections. The rule element does not have any other elements contained as the attribute container encapsulates them. A PRDLRule is composed of StaticRule attributes and DynamicRule attributes.

StaticRuleAttribute

The StaticRuleAttribute is the parent element for all the rule elements that are not part of a dynamic process. These elements are the DataController, DataObject which belongs to a DataSubject, Modality, Purpose, Instrument, Location and Exception. Most of the elements are described in the Section 4.3 glossary. The DataController is described in Section 4.1 within the meta access model. The elements which require a short introduction are the Instrument, Location and the Exception.

The Instrument element represents the possibility of specifying which instrument is used by the DataController to perform an action on the data. When the data access is restricted to the location of the DataController the corresponding element is used for implementation. Finally the Exception element defines if there are specific exceptions concerning the processing of data such as special authentication needed or the prohibition of processing for certain people. A set of data processing related actions was also included as examples for the Action element. These are defined in the Glossary. More information on the design of a PRDL rule will be given in Section 5.

DynamicRuleAttribute

A rule consists not only of static but also of *DynamicRule* attributes. It serves as a parent class for all attributes that can be related to a dynamic condition or a process. The elements serve as a parent for the *Action*, *Condition*, *State*, *Effect* and *CronAttribute* elements. First the Condition element is used to model conditions that influence the rule in a dynamic way. These conditions can include one or more *ConditionStatements*. For example, the daytime can be a condition that influences the access to distinct data. Second the State attribute is introduced to gain the ability of a rule being a process which was discussed very

much throughout the consortium and maybe will influence the future development of PRDL. In the current state this element is not used as the rule does not implement a process.

Third, the *Effect* element introduces an ability of PRDL to change the effect that a rule actually has on a system. It can be used to influence other processes or data within the system besides clarifying whether performing actions on data is allowed or not. Fourth, the *CronAttribute* had to be embedded to meet a special requirement that originates from data protection legislation. The rule has to be able to model the fact that data, for example, has to be deleted after 10 years. As a Cron job is a event scheduler under Unix systems, the *CronAttribute* has similar functionalities triggering a rule to check such time based actions. The DynamicRuleAttributes are in an preliminary state at the time of writing and will be specified in greater detail in the D3.4 final specification of PRDL.

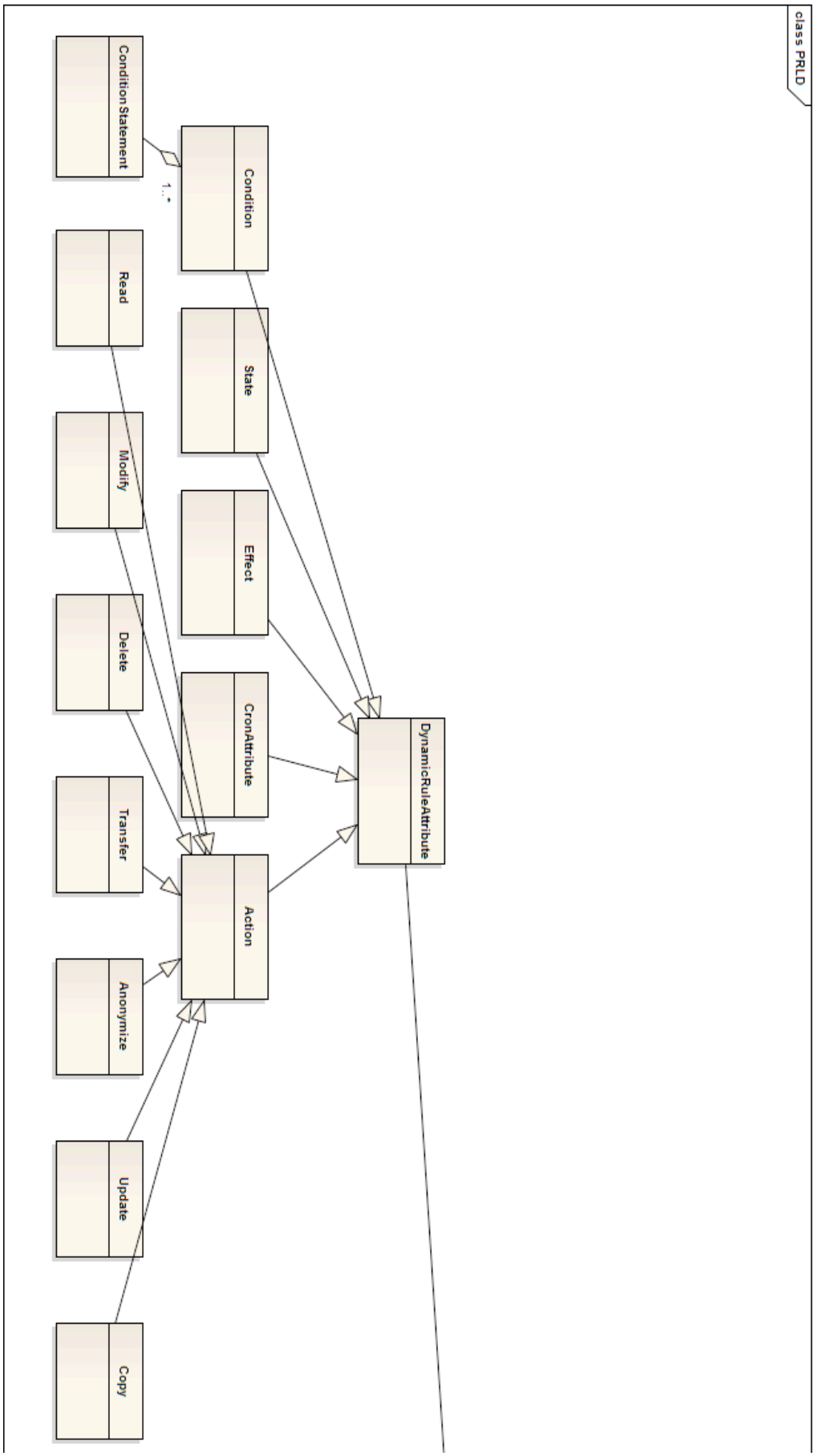

Figure 2: Meta Rule Model UML (part 1) **(http://ict-endorse.eu/wiki/index.php/PRDL)**

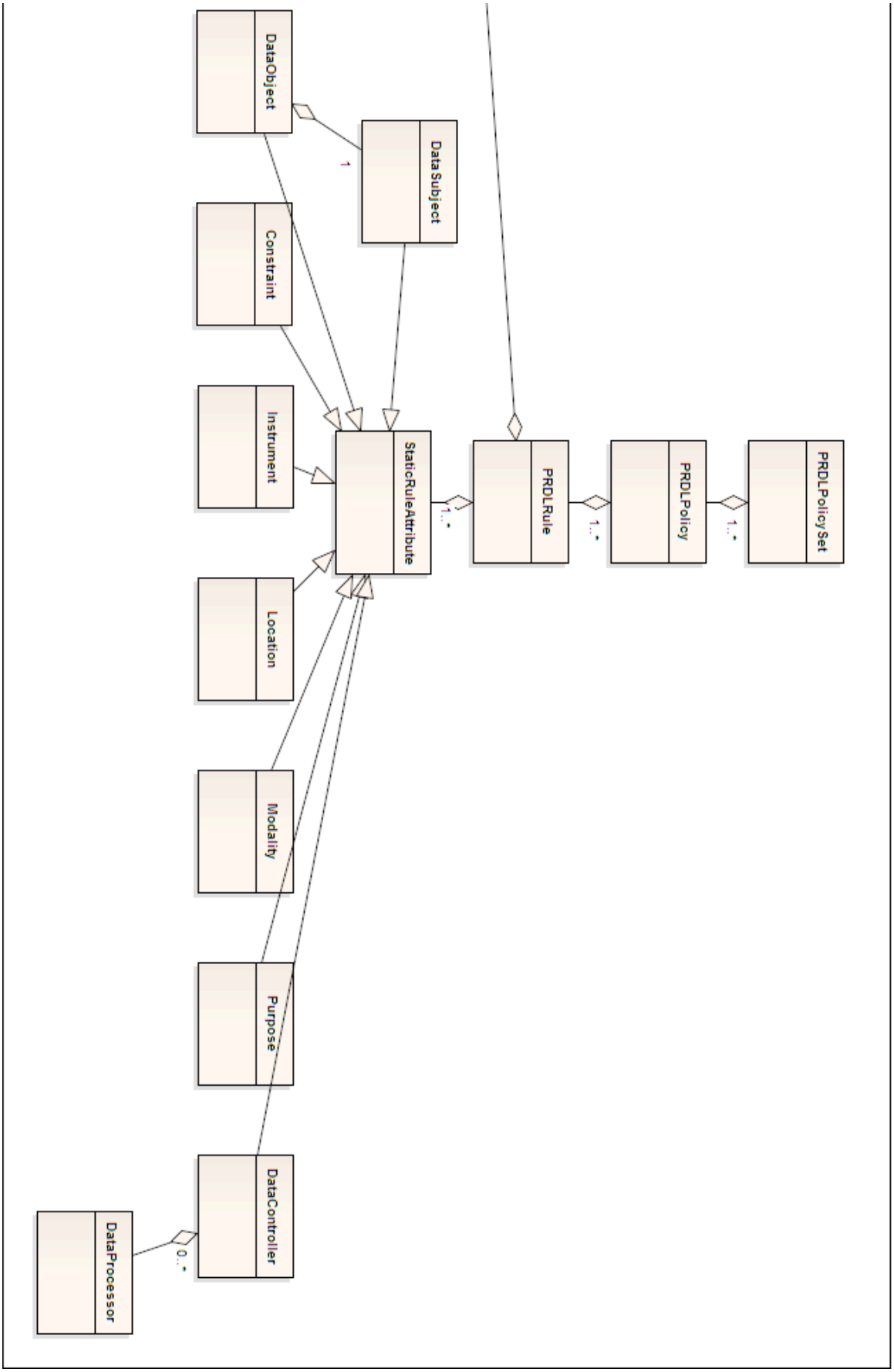

Figure 2: Meta Rule Model UML (part 2) **(http://ict-endorse.eu/wiki/index.php/PRDL)**

4.3. Glossary

The ENDORSE Glossary covers the central concepts that are relevant for the Meta-Model of the Privacy Rule Definition Language (PRDL) and the Access Model which were developed in a joint effort by various consortium partners from different domains. The Glossary is available under http://ict-endorse.eu/wiki/index.php/Glossary.

This ENDORSE Glossary is presented in tabular form; the entries are clustered alphabetically for the sake of easier navigation. Concepts that are present in the ENDORSE Meta-Model have the entry "YES" in the last column. Entries of "YES" with parentheses indicate presence in the Access Meta-Model only. Relevant references for the glossary are [15 – 19]. All the entries in the glossary marked with a "NO" in the "included in the Meta-Model" column are not directly used within one of the models but are important to understand expressions throughout the ENDORSE project.

4.3.1. A

Expression	*Definition*	*Reference*	*Short Description*	*Example*	*included in Meta Rule Model*
Action	An operation on a resource	XACML specification (CRuecker, SUAS)	Every operation that can be conducted with data is defined as an action. These operations can differ in a broad range from read, write operations to transfer operations up to selling operations.	Read, modify, delete, transfer, copy, anonymize	YES
Act	The act performed by the subject (subject = e.g. a data controller)	Breaux, T. D. (2009). Legal Requirements Acquisition for the Specification of Legally Compliant Information Systems. Phd thesis, p. 24. (SOlislaegers, TILT)	**This concept concerns the formulation of runtime requirements.** An act is performed by a subject on an object.	Examples: collection, recording, organization, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, blocking, erasure or destruction (see Article 2(b) Data Protection	YES (implicit in the Action element)

				Directive)	
Actor (Subject)	The actor who performs an action. (3)	Breaux, T. D. (2009). Legal Requirements Acquisition for the Specification of Legally Compliant Information Systems. Phd thesis, p. 24. (SOlislaegers, TILT)	**This concept concerns the formulation of runtime requirements.** Answer to the question of 'Who performs the action'? (Breaux, p. 25) . Data protection relevant legal actors are data controller, data processor, data recipient. (PBueso, UniZar)	Will mostly, in our context, be the data controller. **SO DON'T CONFUSE WITH DATA SUBJECT!**	NO
Anonymize (Dissociation) (is an) Action	An action describing the anonymization of data by a data processor	CRuecker, SUAS; PBueso, UniZar	The legal definition could also include the term dissociate, which is the legal expression for “anonymize” under Spanish law. Anyway, “anonymize” is a verb denoting a certain type of action.	Substitute John Doe with XX.	YES

4.3.2. B

No entries under “B” at the time of writing.

4.3.3. C

Expression	*Definition*	*Reference*	*Short Description*	*Example*	*included in Meta Rule Model*
Consent (part of) Data Object	'The data subject's consent' shall mean any freely given specific and informed indication of his wishes by which the data subject signifies his	Article 2(h) DPD, PBueso, UniZar	Data owners give consent for specific actions and data objects. This consent is the basis for every data processing. The company has to gather consent from	Consent for newsletter sending	(YES)

	agreement to personal data relating to him being processed.		every data owner for every distinct action they want to conduct on the data. There exist several forms of consent: *explicit* (in turn, written or not written), *implicit* (derived from positive actions), *tacit* (derived from negative actions or omissions).		
Constraint	Something that serves as a restrictive condition to avoid actions form happening. (CRuecker, SUAS)			The user wants to view data but the due to his insufficient rights it is prohibited. (CRuecker, SUAS)	(YES)
Controller (data controller, legal definition)	'Controller' shall mean the natural or legal person, public authority, agency or any other body which alone or jointly with others determines the purposes and means of the processing of personal data.	Article 2(d) DPD.	Where the purposes and means of processing are determined by national or Community laws or regulations, the controller or the specific criteria for his nomination may be designated by national or Community laws.		(YES)
Condition	Is a Boolean expression that refines the applicability established by target.	XACML specificati on (Cruecker, SUAS)	Evaluation of a *Condition* can also result in an error (Indeterminate) or discovery that the *Condition* doesn't apply to the request (NotApplicable). A Condition can be quite complex, built from an arbitrary nesting of non-boolean functions and attributes.	Condition is either true or false	YES

Condition (Legal Definition)	An event that occurs before, during or after executing a rule. There are different types of conditions (temporal, subjective...).	Breaux (2009), p. 25 (SOlislaegers, TILT)	**This concept concerns the formulation of runtime requirements** Answer to the question: 'When is the action performed', i.e. under which condition.		(YES)
CronAttribute	This attribute enables users to schedule rule execution to run periodically at certain times or dates.	CRuecker, SUAS	Defines timed trigger of rules to be executed.	Delete data of person 10 years after passing away.	YES
ConditionStatement	A condition can be composed out of several condition statements.	CRuecker, SUAS	For the purpose of aggregation	Age > 14y && gender == f	YES
Copy	Is an action describing the copying of data	PBueso, UniZar	Any form of duplication of any data.	Paper printout of birth date	YES

4.3.4. D

Expression	*Definition*	*Reference*	*Short Description*	*Example*	*included in Meta Rule Model*
Data Object	Factual information used as a basis for reasoning.	Merriam Webster, (CRuecker, SUAS)	The data object field is the atomic representation of a part of data object. It represents the smallest part in a data element. Note: In the access model the consent is included in the DataObject.	First Name, Last Name	(YES)

Data (includes) Data Objects	An individual data entity that can be matched to a to an identified or identifiable natural person.	DSG 2000, DPD (CRuecker, SUAS)	Every piece of data that has an unambiguous relation to a data subject is a data object. Data can be composed out of many data objects. (DSG 2000, DPD)	Name, Tel. Nr, Address	YES
Data Subject (Legal Definition)	Identified or identifiable natural person	Article 2 (a) Data Protection Directive (SOlislaegers, TILT)	An identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity.	Anyone who uses the internet/online services and can be identified	(YES)
Data Subject (is a) Data Role	An individual real or virtual legal entity that has properties assigned to it. - The data subject is the person to whom personal data (a given piece or item of information in whatever form) refer.	DSG 2000 (CRuecker, SUAS), DPD - Article 2(a) DPD, (RLeenes, TILT; PBueso, UniZar)	The data subject is the object of interest as it has data related and this data companies wants to process. In the legal definition this data relation is expressed. Data Subject can be Data Owner and/or Data Processor.	Human User, Computer User, the customer, an individual, EurA	(YES)
Data Owner (is a) Data Role	A natural or legal person who holds data as a personal property. There could be	DPD (RLeenes, TILT)	A data owner (person) provides personal data to a data processor	An Individual, EurA, etc. (RLeenes, TILT)	(YES)

	different categories of data processors such as “data sub-processors”, i.e. companies hired by the data processor or working for him (under supervision of the data controller).		(company) in order to enable interaction with it.		
Data Processor (is a) Data Role	A natural or legal person, public authority, agency or any other body, which processes personal data on behalf of the controller.	&4 DSG 2000, (CRuecker, SUAS), Article 2(e) DPD (RLeenes, TILT)	The data processor has the intention of processing data from the data for a purpose that has to be well defined.	Company, State, Legal Authority, EurA	(YES)
Delete (is an) Action	An action describing the erasing of data.	CRuecker, SUAS; PBueso, UniZar	Make data inexistent	Any deletion of data (but not of rules)	YES
DynamicRule Attribute	Rule attributes that enable change during process execution.	CRuecker, SUAS; Soluta; CN	Rules can dynamically effect other rules or processes	On successful execution of rule #1, modify rule #2	YES

4.3.5. E

Expression	*Definition*	*Reference*	*Short Description*	*Example*	*included in Meta Rule Model*
Exclusion	To prevent or restrict the usage of data. For instance, “data D concerning data subject S cannot be transferred to third parties” points to an exclusion (everyone's prevented from transferring D).	Breaux, 2009 (CRuecker, SUAS)	An act that an actor has no express permission to perform or that an actor is not expressly required or prohibited from performing.		NO

Event	Something that happens which is relevant to the data handling: occurrence (CRuecker, SUAS)	PBueso, UniZar	An “event” as something which is generally relevant to the data handling; for instance, a data subject signs a form authorizing onward transfer (this fits into the definition “something what happens which is relevant...”).	It occurs that a user wants to view the profile of a relative.	NO
Effect	Effect (of a rule): “Permit” or “Deny” (Technical Term). For legal people it is rather a deontic modality which is accorded to a given action	XACML specification (Cruecker, SUAS; PBueso, UniZar)	Every operation that can be conducted with data is defined as an action. These operations can differ in a broad range from read, write operations to transfer operations up to selling operations. (in the broadest of terms, we lawyers speak of three deontic or regulative modalities: permission, prohibition and obligation).	Access is granted	YES
Exception (legal definition)	An event that does not occur before, during or after executing a rule	Breaux (2009), p. 25. (SOlislaegers, TILT)	**This concept concerns the formulation of runtime requirements** Answer to the question 'When is the action not performed?'	User is part of SysAdmin group but is not allowed to access.	YES
Exception (legal construct, defeasible reasoning)	An action or event that defeats an otherwise valid action or event.	(RLeenes, TILT)	Exceptions are to be read in conjunction with rules (to which they form the exception). The are similar to negative conditions in rules and could be formulated as such.	R1: The health data of DS may no be consulted without his consent. The DS health data may be consulted if	

				the DS is unconsciou s. (Overrides R1).	

4.3.6. F

Expression	*Definition*	*Reference*	*Short Description*	*Example*	*included in Meta Rule Model*
Fact	An act or state of being that is conditionally true.	Breaux, 2009 (CRuecker, SUAS)	A fact additionally is a piece of information presented as having objective reality. (Merriam Webster)		NO

4.3.7. G

No entries under “G” at the time of writing.

4.3.8. H

No entries under “H” at the time of writing.

4.3.9. I

Expression	*Definition*	*Reference*	*Short Description*	*Example*	*included in Meta-Model*
Incident	“Incident” as any event which affects or could affect the security and integrity of personal data .	PBueso, UniZar	Note: this is more or less the legal definition of “incident” under Spanish Law		NO
Instrument	The Instrument element models the different tools which could be used by a data controller to execute actions on the data.	CRuecker, SUAS	Instruments are not supposed to describe the used tools for data maintenance.	e.g. Internal system frontend, direct database access etc.	YES

4.3.10. J

No entries under “J” at the time of writing.

4.3.11. K

No entries under “K” at the time of writing.

4.3.12. L

Expression	*Definition*	*Reference*	*Short Description*	*Example*	*included in Meta-Model*
Location	This element is used to defines the current location of the data controller specifying where the data operations are triggered from.	CRuecker, SUAS	This elements helps specify whether the data controller operates from inside or outside of his company and whether the data controller acts from another country.	Company network,national network, from a foreign country via VPN	*YES*

4.3.13. M

Expression	*Definition*	*Reference*	*Short Description*	*Example*	*included in Meta Rule Model*
MAY			Well defined in RFC 2119 (accessible from http://tools.ietf.org/html/rfc2119)	The nickname MAY be at least 6 characters long. (CRuecker, SUAS)	NO
MUST (NOT)			Well defined in RFC 2119	The nickname MUST be at least 6 characters long. (CRuecker, SUAS)	NO
Modality	The modality of the action (e.g., may, must, etc.)	Breaux, 2009, RFC 2119	The classification of propositions on the basis of whether they assert or deny the possibility, impossibility, contingency, or necessity of	May, Must, Should	YES

			their content. (M. Webster)		
Modify (is an) Action	An action describing the altering or updating of data by a data processor.	CRuecker, SUAS; PBueso, UniZar	The action can be more specific. “Rectify” (in case of mistakes) and “update” (in case of new personal data). Furthermore, the question arises of whether “blocking”, “cancellation”, “deletion” have to be considered forms of modification as well.	Changed status to married.	YES

4.3.14. N

No entries under “N” at the time of writing.

4.3.15. O

Expression	*Definition*	*Reference*	*Short Description*	*Example*	*included in Meta Rule Model*
Object	The object on which the action is performed	Breaux (2009), p. 24, 25 (SOlislaegers, TILT)	**Concept used in the context of formulating runtime requirements** Answer to the question: 'Upon what is the action performed?'	See p. 29 Breaux: notice. Other example: BSN number or any other type of data.	NO

4.3.16. P

Expression	*Definition*	*Reference*	*Short Description*	*Example*	*included in Meta Rule Model*
Personal Data			(legal) any information relating to an identified or identifiable natural person (`data subject')	name, hair colour, insurance status	NO
Property			Some attribute that has data (CRuecker, SUAS)	The password has a minimum length of six characters. (CRuecker, SUAS)	NO
Processing			Any operation or set of		NO

of personal data (legal definition)			operations which is performed upon personal data, whether or not by automatic means, such as collection, recording, organization, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, blocking, erasure or destruction. Article 2(b) DPD.		
Processor (is a) Data Role (legal definition)	see also *Data Processor*		A natural or legal person, public authority, agency or any other body which processes personal data on behalf of the controller. Article 2(e) DPD.		NO
Purpose	The purpose is a reason why something is done or used (Merriam Webster)	DSG 2000, (CRuecker, SUAS). Cf. Article 6 Data Protection Directive (SOlislaegers, TILT).	Something set up as an object or The purpose of a data processing has to be there for every processing action and has to be precise. Usage of general purpose statements such as general marking are prohibited by law (DSG2000) to be attained: Intention (1)	For goods delivery, for a newsletter sending, for interacting with the customer	YES
Purpose (legal definition)	An act describing why an action is performed	Breaux (209), p. 24, 25. (SOlislaegers, TILT)	**This concept concerns the formulation of runtime requirements.** Answer to the question 'Why is the action performed'	ibid	YES
Policy	A Policy represents a single access control policy, expressed through a set of Rules.	XACML specificato n, (CRuecker, SUAS)	The policy is a container that includes a target, a set of rules as well as obligations. (XACML)	Rule Container	YES
PolicySet (contains) Policy	The policy set is the overall container that encapsulates all the policies	XACML specificatio n (CRuecker, SUAS)	A PolicySet is a container that can hold other Policies or PolicySets, as well as references to policies found in remote locations.	Policy Container	YES

	within a system.		(XACML)		
Permission	An act that an actor is permitted to perform.	Breaux, 2009 (CRuecker, SUAS)	Permission describes the state of a data processor having the consent of the data owner to perform distinct actions on personal data.	Access is permitted to administrators	YES
Processing Ground	An organization can process data for many reasons, which can be for example legal obligations or operative reasons. The Processing Ground element specifies the precise reason for what data access is granted.	CRuecker, SUAS, RLeenes, TILT	As consent is only one of the explicit permissions for the processing of data, they have to be taken into account. This is accomplished with the ProcessingGround.	Legal obligations, operative obligations, etc.	YES

4.3.17. Q

No entries under “Q” at the time of writing.

4.3.18. R

Expression	*Definition*	*Reference*	*Short Description*	*Example*	*included in Meta Rule Model*
Recipient (legal definition)			A natural or legal person, public authority, agency or any other body to whom data are disclosed, whether a third party or not; however, authorities which may receive data in the framework of a particular inquiry shall not be regarded as recipients. Article 2(g) DPD		NO
Requester		CN	An individual, real or virtual legal entity,	Human, Organisation,	NO

			who requests the ENDORSE Process Engine to get access on the data. (Rizwan)	etc. (Rizwan)	
Rule	A representation of a single access control statement.	XACML specification (CRuecker, SUAS)	It contains a *condition*, which is a Boolean function. If the Condition evaluates to true, then the Rule's Effect (a value of Permit or Deny that is associated with successful evaluation of the Rule) is returned. (XACML)	EurA MAY use data for purpose	YES
Restriction	A limitation on the use of of personal data. (Merriam Webster) "Data D concerning data subject S can be transferred only under authorization of X" is a restriction.	Ardagna, 2008 (CRuecker, SUAS; PBueso, UniZar)	A privacy statement specifies restrictions that have to be satisfied before or after access to personal data is granted. If just one condition is not satisfied, the access should not be granted. (Ardagna, 2008)		NO
Refrainment (is a) Restriction	An act that an actor is prohibited from performing	Breaux, 2009 (CRuecker, SUAS)	See *Restriction*	Paul is not allowed to access data.	YES
Retention (part of) Data Object	Retention in the DP world refers to the obligation to retain (store) data for a certain amount of time (after which it MUST be deleted).	XACML specification (CRuecker, SUAS)	With the retention attribute the data object has knowledge of explicit time that the data processor is allowed to perform actions or which actions are allow in general.		NO
Read (is an) Action	An action describing the viewing of data by a data processor.	CRuecker, SUAS; PBueso, UniZar	There has to be a further specification of actions that are read operations such as "retrieve" and	Data processor reads data of Giulia.	YES

			"access".		
Request	Aggregation of Data, Request Purpose and Data Controller in the Access Model	CRuecker, SUAS	The data processor creates a request to perform an action on data.	SysAdmin requests DataObject(s) for specific Purpose(s)	YES

4.3.19. S

Expression	*Definition*	*Reference*	*Short Description*	*Example*	*included in Meta Rule Model*
SHOULD (NOT)			Well defined in RFC 2119 (s. http://tools.ietf.org/html/rfc2119)	The nickname SHOULD be at least 6 characters long. (CRuecker, SUAS)	NO
State	The current condition of a being or object.	Breaux, 2009, (CRuecker, SUAS)	Particular condition that some entity is in at a specific time	Online or offline.	YES
StaticRule Attribute	Process-indepentent persistent rule attributes super class	CRuecker, SUAS	The StaticRuleAttribute element serves as a parent for all static elements in the PRDL rule.	Normal access rule elements without process interferenc e	YES

4.3.20. T

Expression	*Definition*	*Reference*	*Short Description*	*Example*	*included in Meta Rule Model*
Third party (legal definition)			Any natural or legal person, public authority, agency or any other body other than the data subject, the controller, the processor and the persons who, under the direct		NO

			authority of the controller or the processor, are authorized to process the data. Article 2(f) DPD.		
Target	Is an encapsulation of a subject, resource and an action.	XCAML specification (CRuecker, SUAS)	Defines simple conditions on the subject, resource, action, and environment that partly determine whether the policy, policy set, or rule applies to a request.		NO
Target (legal definition)	To where/with whom an action is performed by the subject	Breaux (2009), p. 24,25 (SOlislaegers, TILT)	**This concept concerns the formulation of runtime requirements.** Answer to the question: 'with whom is the transaction performed?	Think of a transfer of data with a third party, the third party = the target. Could also be, e.g. a type of database where data is sent to.	YES
Transfer (is an) Action	An action describing the moving of data by a data controller. (Technical term). As legal term "disclosure" is suggested	CRuecker, SUA, PBueso, UniZar	The data processor may transfer data as well, but only following specific controller's instructions. Secondly, data transfer may take place within the data controller's sphere of control; i.e. without involving a “data recipient” in legal sense. Maybe we could use “disclosure” instead of “transfer”, but this is just a suggestion. Suppose for instance that I send a backup device to the company's headquarters: would it be a “data transfer”? Note that special legal requirements and conditions may apply to transfers of sensitive data (encryption). Thirdly, data transfers might be, so to say, of very different scope: inside/outside national/EU/EEE borders, to countries with/without an adequate level of protection, to companies registered at Safe Harbor.	Data is transferred to a third party company or to another department.	YES

4.3.21. U

Expression	*Definition*	*Reference*	*Short Description*	*Example*	*included in Meta Rule Model*
User	User (as a short name for Information System User).	PBueso, UniZar	Information System User (is a legal category (at least under Spanish Data Protection Law)) must be carefully distinguished from the "user" of electronic communication services according to Directive 2002/58/EC.	Different types of users might be defined (ranging e.g. from administrators to end-users), but this depends on companies' internal policies.	NO
Update	The action of bringing data to the most recent state	CRuecker, SUAS	It is considered that an update can be more than bringing data to a recent state so the update can be seen as base action. Further sub-actions could be rectify, add, remove etc.	The data controller requests the permission for the rectification of a customers data.	YES

4.3.22. V

No entries under "V" at the time of writing.

4.3.23. W

No entries under "W" at the time of writing.

4.3.24. X

No entries under "X" at the time of writing.

4.3.25. Y

No entries under "Y" at the time of writing.

4.3.26. Z

No entries under "Z" at the time of writing.

5. Syntax of PRDL

In general, syntax deals with how sentences are constructed and how users of human languages or technical languages apply a great variety of possible arrangements of the elements in sentences. One of the most obvious yet important ways in which languages differ is the order of the main elements in a sentence [20]. In PRDL we have to define a syntax that is able to deal with rule statements concerning privacy and data protection. These rules are derived from national and European data protection law as well as company policies and obligations. As companies are under legal compulsion to process and handle data in a way that is compliant to the law, this is the most important basis for the PRDL.

Although PRLD is not designed to express law itself, it should be possible to check it against rule sets derived from different national law sets. The laws of society are based on human communication as they are negotiated by humans living together [21]. This implies that law has never a stable state but is adapted, changed and negotiated all the time. Resulting from that fact, the development of PRDL leads to challenges when trying to define rules out of legal texts. Breaux [17] defines three fundamental challenges faced by engineers who extract legal requirements from legal texts. These are ambiguity, traceability and accountability. The PRDL development faces similar challenges as the language has to deal with the ambiguity of law, the loss of the original law text where the rule was abstracted which makes traceability complicated, and finally accountability of the rule which should answer the question whether the rule act in the compliant way it should. Creating statements in PRLD which are not compliant to the law will be possible but a support system within the ENDORSE framework should guide the rule editor to avoid non compliant rules. In order to meet the requirements that were developed and negotiated within the ENDORSE consortium, the following syntactical assembly is presented.

The syntax of PRDL consists of syntax templates provided to help the user to create PRDL statements and make the rule creation easier. These templates have been created after an evaluation of the needed constructs described in D2.4. Additionally the templates will present the basis for the editor development described in D4.1. Also some scenarios from D2.3 are based on these PRDL templates. As this deliverable defines the preliminary stage of PRDL also the syntax templates are about to be changed and adopted towards a final specification which is due in April, 2012. Further examples for rules that can be implemented with PRDL are given in appendix Section 8.2. To assure the compatibility to a real world use case the privacy statement of Docticare was analysed and adequate rules were defined. These are given in appendix section 8.3. Additionally a rule set including a fact description for the first year review scenario was created as a starting point in order to demonstrate the interrelation between the language, the rule engine and the system. These use case will be implemented with a demonstrator and will be presented in the D.4.3 deliverable.

The following chapter shows in addition to the pseudo natural and XML representation of the PRDL rules also a possible implementation in the Drools [22] rule language. In section 5.2, an additional rule set is given that will be implemented in the D.4.3. It is intended to give further examples for the usage of the rule templates. Due to the fact that the preliminary rule engine specification is due in October, 2011 Drools was chosen as a first candidate to serve as the engine within the ENDORSE framework. Advantages of Drools are that the rule engine is completely generic in respect to the used data model for the rules. The final decision for a rule engine will be taken after the publish of the D4.3. The next section explains each template and gives an introduction to the specific cases that can be implemented with the rule construct.

5.1. Syntax Templates for PRDL

The syntax templates explained in this Section have been created during the development process of scenario generation and the listing of example rules to these scenarios. Privacy officers, namely the personal responsible for maintaining companies policies concerning privacy, use this templates to formulate PRDL statements which represent their company ones. The templates include the most important language constructs defined and described in D.2.4. As an extension every template has a set of rules from the scenarios given in D.2.3 and a representation in XML from the meta model described in Section 4. It has to be noted that the given Drools examples have not yet been implemented and are only draft in valid Drools syntax.

5.1.1. XML base structure

Before starting with the PRDL rule templates and according rule examples in XML and Drools the XML base structure is given in the following. In the subsequent examples the XML header will be dropped to assure a better readability. The complete XSD schema for the PRDL is given in the appendix in section 8.1:

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

<PRDLPolicySet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://evesim.org/~cruecker/prdl.xsd">
        <PRDLPolicy>
                <PRDLRule>
                        <DataController>
                                <DataProcessor>
                                        <name/>
                                </DataProcessor>
                        </DataController>
                        <Modality/>
                        <Action/>
                        <DataObject>
                                <DataSubject/>
                        </DataObject>
                        <Purpose/>
                </PRDLRule>
        </PRDLPolicy>
</PRDLPolicySet>
```

5.2. PRDL Rule Templates

This section shows an overview of the PRDL templates and the used elements. The templates are described in greater detail in the following subsections and examples are provided. As PRDL is developed in an ongoing process the templates may be changed and extended in the future. [] are used to mark elements within the rule, {} indicate the choice of on of the elements, () are used to mark placeholders for elements only needed to create human readable rules. They are implicitly part of the rule.

Template for a data object specification rule (obsolete) **[DataObject] {MUST, MAY} (INCLUDE) [Property].** Rule Elements: [DataObject][Modality][Property]
Template for normal privacy data access rule **[DataController] {MUST, MAY} {VIEW, ADD, DELETE, MODIFY, STORE, ACTION} [Data Object] (FOR) [Purpose]** Rule Elements: [DataController][Modality][Action][DataObject][Purpose]
Template for a normal access rule with a constraint **[DataController] {MUST, MAY} {VIEW ENTER, DELETE, MODIFY, ACTION} [Data Object] (FOR) [Purpose] (IF) [Constraints]** Rule Elements: [DataController][Modality][Action][DataObject][Purpose][Constraint]

Template for data access rules with time constraints

[DataController] {MUST, MAY} {VIEW, ADD, DELETE, MODIFY, STORE} [Data Object] (FOR) [Purpose] (WHEN) [TimeConstraint]

Rule Elements: [DataController][Modality][Action][DataObject][Purpose][TimeConstraint]

5.2.1. Template for a data object specification rule

During the requirement discovery phase, PRDL was envisaged to check whether a data object meets certain properties. For example, does the password entered by a user meet certain security requirements. Although being dropped later on this thoughts resulted in the first rule syntax template presented in the following. The requirement was marked as not important later on because this is a activity the front-end of a system has to be able to handle. For completeness the template has not been dropped but will not be pursued any further.

The template is used to define general specification of data objects:

I. **[DataObject] {MUST, MAY} (INCLUDE) [Property].**

   Rule Elements: **[DataObject][Modality][Property]**

5.2.1.1. Examples for the data object specification rule

This rules were composed out of D2.1 and deal with EuropA UseCases and giving examples for the current rule template. The given rules are covered by the described rule template.

- Rule 1: The password MUST INCLUDE 6 or more characters.
- Rule 2: The password MUST INCLUDE numbers and letters.

For two rules there are XML and according Drools examples given. These were created with an older version of the meta model and are not supported any longer:

- Rule 1: The password MUST INCLUDE 6 or more characters.

```
<PRDLRule>
        <StaticRuleAttributes>
                <DataObject>
                        <type>password</type>
                        <owner>
                                <name>EurA Customer John Doe</name>
                        </owner>
                </DataObject>
                <Modality>
                        <must/>
                </Modality>
                <Property>
                        <type>
                        <key>length</key>
                        <addition>greater or equal</addition>
                        <value>6</value>
                        </type>
                </Property>
        </StaticRuleAttributes>
</PRDLRule>
```

- Rule 2: The password MUST INCLUDE numbers and letters.

```
<PRDLRule>
        <StaticRuleAttributes>
                <DataObject>
                        <type>password</type>
                        <owner>
```

```
                        <name>John Doe</name>
                    </owner>
                </DataObject>
                <Modality>
                    <must/>
                </Modality>
                <Property>
                    <type>
                    <key>composition</key>
                    <value>numbers && letters</value>
                    </type>
                </Property>
            </StaticRuleAttributes>
        </PRDLRule>
```

Drools examples for the given rules.

- Rule 1: The password MUST INCLUDE 6 or more characters.

```
rule "Check Password Length"
when
    $p : Login (password.length() >= 6)
    $a : Access (username = "John Doe")
then
    $a.grantAccess();
end
```

- Rule 2: The password MUST INCLUDE numbers and letters.

```
rule "Check Password"
when
    $p : Login (password.containsLetters() == true
    && password.containsNumbers() == true)
    $a : Access (username = "John Doe")
then
    $a.grantAccess();
end
```

5.2.2. Template for normal privacy data access rule

This rule template is supposed to handle all the regular requests for privacy data within a company system that does not have any kind of constraint included. The design of the rule originates from the discussion process within the consortium and the requirements deliverables.

The template given is used to construct a simple access rule:

II. **[DataController] {MUST, MAY} {VIEW, ADD, DELETE, MODIFY, STORE, ACTION} [Data Object] (FOR) [Purpose]**

Rule Elements: [DataController][Modality][Action][DataObject][Purpose]

5.2.2.1. Examples for the privacy data access rule

These examples are taken from the D2.3 and cover some of the possible rules that could be applied within the Docticare System of Europ Assistance. The rules mainly cover the usage of different data objects for different purposes. Most of the rules taken from the D2.3 cover the different usages of data for distinct purposes and therefore are implementable with this rule template.

1. Rule 3: EurA MAY read name, address, mobile number, FOR customer identification
2. Rule 4: EurA MAY read blood group FOR third party research.
3. Rule 5: EurA MAY read customers email FOR future purpose.

For three given rules there are XML and according Drools examples shown in the following.

- Rule 3: EurA MAY process name, address, mobile number, FOR customer identification

```
<PRDLRule>
        <DataController>
                <name>EurA employee</name>
                <DataProcessor>
                        <name>EurA</name>
                </DataProcessor>
        </DataController>
        <Modality>
                <type>may</type>
        </Modality>
        <Action>
                <type>process</type>
        </Action>
        <DataObject>
                <type>name, address, mobile number</type>
                <DataSubject>
                        <type>EurA customer</type>
                </DataSubject>
        </DataObject>
        <Purpose>
                <type>customer identification</type>
        </Purpose>
</PRDLRule>
```

- Rule 4: EurA MAY transfer blood group FOR third party research.

```
<PRDLRule>
        <DataController>
                <name>EurA employee</name>
                <DataProcessor>
                        <name>EurA</name>
                </DataProcessor>
        </DataController>
        <Modality>
                <type>may</type>
        </Modality>
        <Action>
                <type>transfer</type>
        </Action>
        <DataObject>
                <type>blood group</type>
                <DataSubject>
                        <type>EurA customer</type>
                </DataSubject>
        </DataObject>
        <Purpose>
                <type>third party research</type>
        </Purpose>
</PRDLRule>
```

- Rule 5: EurA MAY store customers email FOR future purpose.

```
<PRDLRule>
        <DataController>
                <name>EurA employee</name>
                <DataProcessor>
                        <name>EurA</name>
                </DataProcessor>
        </DataController>
        <Modality>
                <type>may</type>
        </Modality>
        <Action>
                <type>store</type>
        </Action>
        <DataObject>
                <type>email adress</type>
                <DataSubject>
                        <type>EurA customer</type>
                </DataSubject>
        </DataObject>
        <Purpose>
                <type>future purposes</type>
        </Purpose>
</PRDLRule>
```

Drools examples for the given rules.

- Rule 3: EurA MAY process name, address, mobile number, FOR customer identification

```
rule "Access to name, address and mobile number"
when
        $r : Request (dataprocessor.getAuth == "EurA"
        && modality.get() == "may"
        && requesteddata.isEqual("name,address,mobilenr")
        && purpose.getPurpose() == "customer identification")
then
        $r.grantAccess();
end
```

- Rule 4: EurA MAY transfer blood group FOR third party research.

```
rule "Transfer of blood group"
when
        $r : Request (dataprocessor.getAuth == "EurA"
        && modality.get() == "may"
        && requesteddata.isEqual("bloodgroup")
        && purpose.getPurpose() == "third party research")
then
        $r.grantAccess();
end
```

- Rule 5: EurA MAY store customers email FOR future purpose.

```
rule "Storage of email"
```

```
when
        $r : Request (dataprocessor.getAuth == "EurA"
        && modality.get() == "may"
        && requesteddata.isEqual("email")
        && purpose.getPurpose() == "future purpose")
then
        $r.grantAccess();
end
```

5.2.3. Template for a normal access rule with a constraint

This rule template is very similar the normal access template but introduces the constraint attribute to enable rule only to fire when a constraint is fulfilled. The examples shown are also taken from the D2.3 and therefore directly apply able in the Docticare system.

III. [DataController] {MUST, MAY} {VIEW, ENTER, DELETE, MODIFY, ACTION} [Data Object] (FOR) [Purpose] (IF) [Constraints]

Rule Design: [DataController][Modality][Action][DataObject][Purpose][Constraint]

5.2.3.1. Examples for normal access rules with a constraint

- Rule 6: A subject MAY read sensitive data of customer FOR a medical treatment IF data subject is a doctor under contract with EurA.
- Rule 7: EurA MUST delete personal data related to contract XX FOR legal compliance IF no accident regarding the contact itself occurred.
- Rule 8: EurA MUST delete personal data related to contract XX FOR legal compliance IF no claim from the target customer occurred.

For three given rules there are XML and according Drools examples illustrated in the following.

- Rule 6: A subject MAY read sensitive data of customer FOR a medical treatment IF data subject is a doctor under contract with EurA.

```
<PRDLRule>
        <DataController>
                <type>any</type>
                <DataProcessor>
                        <name>any</name>
                </DataProcessor>
        </DataController>
        <Modality>
                <type>may</type>
        </Modality>
        <Action>
                <type>read</type>
        </Action>
        <DataObject>
                <type>sensitive data</type>
                <DataSubject>
                        <type>any</type>
                </DataSubject>
        </DataObject>
        <Purpose>
                <type>medical treatment</type>
```

```
            </Purpose>
            <Constraint>
                <type>data processor under contract of EurA</type>
            </Constraint>
        </PRDLRule>
```

- Rule 7: EurA MUST delete personal data related to contract XX FOR legal compliance IF no accident regarding the contact itself occurred.

```
        <PRDLRule>
            <DataController>
                <type>EurA employee</type>
                <DataProcessor>
                    <name>EurA></name>
                </DataProcessor>
            </DataController>
            <Modality>
                <type>must</type>
            </Modality>
            <Action>
                <type>delete</type>
            </Action>
            <DataObject>
                <type>sensitive data</type>
                <DataSubject>
                    <type>contract xx owner</type>
                </DataSubject>
            </DataObject>
            <Purpose>
                <type>legal compliance</type>
            </Purpose>
            <Constraint>
                <type>no incident with contract XX</type>
            </Constraint>
        </PRDLRule>
```

- Rule 8: EurA MUST delete personal data related to contract XX FOR legal compliance IF no claim from the target customer occurred.

```
        <PRDLRule>
            <DataController>
                <type>EurA employee</type>
                <DataProcessor>
                    <name>EurA></name>
                </DataProcessor>
            </DataController>
            <Modality>
                <type>must</type>
            </Modality>
            <Action>
                <type>delete</type>
            </Action>
            <DataObject>
                <type>sensitive data</type>
                <DataSubject>
                    <type>contract xx owner</type>
                </DataSubject>
            </DataObject>
            <Purpose>
                <type>legal compliance</type>
            </Purpose>
            <Constraint>
                <type>no claim from customer</type>
            </Constraint>
        </PRDLRule>
```

Drools examples for the given rules:

- Rule 6: A subject MAY read sensitive data of customer FOR a medical treatment IF data subject is a doctor under contract with EurA.

```
rule "Doctor from Docticare"
when
        $r : Request (dataprocessor.getAuth == "Any"
        && modality.get() == "may"
        && requesteddata.isEqual("sensitiveData")
        && purpose.getPurpose() == "medical treatment"
        && constraintInfo.get()== "ContractedDoctor")
then
        $r.grantAccess();
end
```

- Rule 7: EurA MUST delete personal data related to contract XX FOR legal compliance IF no accident regarding the contact itself occurred.

```
rule "Delete data from contract"
when
        $r : Request (dataprocessor.getAuth == "EurA"
        && modality.get() == "must"
        && requesteddata.isEqual("data from contract XX")
        && purpose.getPurpose() == "legal compliance"
        && constraintInfo.get()== "no incidents")
then
        $r.grantAccess();
end
```

- Rule 8: EurA MUST delete personal data related to contract XX FOR legal compliance IF no claim from the target customer occurred.

```
rule "Delete data from contract if no claim"
when
        $r : Request (dataprocessor.getAuth == "EurA"
        && modality.get() == "must"
        && requesteddata.isEqual("data from        contract XX")
        && purpose.getPurpose() == "legal compliance"
        &&     constraintInfo.get()== "no claims")
then
        $r.grantAccess();
end
```

5.2.4. Template for data access rules with time constraints

Especially in the insurance business the deletion of data after a certain period is a key factor to be compliant to data protection law. These rules can be implemented by using the template that includes time constraints.

IV. [DataController] {MUST, MAY} {VIEW, ADD, DELETE, MODIFY, STORE} [Data Object] (FOR) [Purpose] (WHEN) [TimeConstraint]

Rule Elements: [DataController][Modality][Action][DataObject][Purpose][TimeConstraint]

5.2.4.1. Examples for normal access rules with a time constraint

- Rule 9: Max's MAY access his data through Docticare WHEN expiration date of the contract + 60 days is not reached.
- Rule 10: EurA MUST delete Max's data WHEN 10 years after the last expiration of his contract passed.

For three given rules there are XML and according Drools examples illustrated in the following.

- Rule 9: Max's MAY access his data through Docticare WHEN expiration date of the contract + 60 days is not reached.

```
<PRDLRule>
        <DataController>
                <type>customer Max</type>
                <DataProcessor>
                        <name>EurA></name>
                </DataProcessor>
        </DataController>
        <Modality>
                <type>may</type>
        </Modality>
        <Action>
                <type>read</type>
        </Action>
        <DataObject>
                <type>personal data</type>
                <DataSubject>
                        <name>Max</name>
                </DataSubject>
        </DataObject>
        <Purpose>
                <type>accuracy</type>
        </Purpose>
        <Constraint>
                <type>contract expiration data + 60 not reached</type>
        </Constraint>
</PRDLRule>
```

- Rule 10: EurA MUST delete Max's data WHEN 10 years after the last expiration of his contract passed.

```
<PRDLRule>
        <DataController>
                <type>EurA employee</type>
                <DataProcessor>
                        <name>EurA></name>
                </DataProcessor>
        </DataController>
        <Modality>
                <type>must</type>
        </Modality>
        <Action>
                <type>delete</type>
        </Action>
        <DataObject>
                <type>personal data</type>
                <DataSubject>
```

```
                        <name>Max</name>
                    </DataSubject>
                </DataObject>
                <Purpose>
                    <type>accuracy</type>
                </Purpose>
                <Constraint>
                    <type>contract expiration data + 10 years</type>
                </Constraint>
                <CronAttribute>
                    <type>check every 24h</type>
                </CronAttribute>
            </PRDLRule>
```

Drools examples for the given rules:

- Rule 9: Max's MAY access his data through Docticare WHEN expiration date of the contract + 60 days is not reached.

  rule "Customer Max access"

  when

  $r : Request (dataprocessor.getAuth == "Max"

  && modality.get() == "may"

  && requesteddata.isEqual("personal data")

  && constraintInfo.get()== "expiration data of contract + 60 not reached")

  then

  $r.grantAccess();

  end

- Rule 10: EurA MUST delete Max's data WHEN 10 years after the last expiration of his contract passed.

  rule "Deletion after 10 years"

  when

  $r : Request (dataprocessor.getAuth == "EurA"

  && action.get() == "delete"

  && modality.get() == "must"

  && requesteddata.isEqual("personal data")

  && constraintInfo.get()== "expiration data of contract + 10 years reached")

  then

  $r.grantDelete();

  end

5.3. Rule Set taken from Deliverable 2.3 related to EurA

These rules represent the basic rule set for the first implementing efforts towards PRDL and the demonstrator.

- Europ Assistance MAY use name, address mobile number FOR customer identification.
- Europ Assistance MAY use blood group FOR third party research.

- Europ Assistance MAY use customer's email FOR future purpose.
- Europ Assistance MAY use name, email address FOR online competition, marketing purpose.
- Europ Assistance MAY use personal data FOR promotion of Docticare services.
- SysAdminGroup MAY use personal data FOR internal service processing EXCEPT user Paul.
- User MAY use non-personal data FOR internal processing.
- DocCharly MAY access Stefania's data FOR medical treatment IF correct token is used.
- Subject MAY read sensitive data of customer IF subject is a doctor under contract with EurA.
- Subject MAY read sensitive data of customer FOR one hour IF consent of the customer is given.
- EurA MAY retain personal data IF user has not given consent to the processing for marketing purposes and not ask for their deletion.
- EurA MAY retain personal data IF user has not given consent to the processing for marketing purposes and not ask for the deletion of the Docticare account.
- EurA MUST delete personal data IF customer asks for their deletion or the deletion of the Docticare account and has not given consent to processing for marketing purposes.
- Customer MAY access her data IF contract has not expired plus 60 days.
- EurA MUST delete sensitive data of a customer IF the last contract has been expired for 10 years.

5.3.1. Rule set for the first year review scenario

These are some facts that describe the setting of the scenario for the first year review. Most of the rules taken from the Docticare Data processing sheet fit into this scenario and also rules created from other scenarios in the D2.3 are taken.

5.3.1.1. Scenario Facts:

- Laura wants to launch an online competition to promote Docticare.
- Laura works in the marketing department of EuropAssistance.
- Laura wants to use name, surname, address, email address, telephone number of the customer for the organization of the online competition.
- Laura wants to use the medical data for online competition (know that this is unlikely but as an example).
- Laura wants to include a third party person to help her with organizing the competition.
- Laura wants to include a colleague of the marketing department to help her organizing the competition.
- Laura wants to transfer data of customers that took part in the competition to a third party.
- Customer has to give consent that data can be used for marketing purposes.
- Customer has to give consent that data can be used for competitions.
- John and Martin are customers of EuropAssistance.
- John and Martin have given consent to the processing of their data and to the usage of data for marketing purposes.
- Angelina is a customer of EuropAssistance but hasn't give consent to the processing and usage of her data.
- Angelina has not yet made a request for the deletion of the data or the deletion of her account.

- Laura wants to include all the customers in the competition also Angelina. (ENDORSE should prevent this from happening)

5.3.1.2. Rules involved taken from Docticare Processing Description:

- Europ Assistance MAY process Name, surname, address, email address, telephone number, FOR commercial information and promotion of the Europ Assistance Service IF explicit consent is given for the specific purpose.
- Employees and partners within EuropAssistance MAY process Personal data FOR providing key services related to the compition or to the marketing activity.
- Europ Assistance MAY disseminate the data of the winner of the competition through the website.
- Europ Assistance MAY NOT disseminate the data of non winners.

5.3.1.3. Other Rules involved:

- Europ Assistance MAY use Name, surname, address, email address, telephone number FOR online competition, marketing purpose.
- Marketing of Europ Assistance MAY use name, email address FOR online competition, marketing purpose.
- Laura (member of marketing department of Europ Assistance) MAY use name, email address FOR online competition, marketing purpose.
- Other member of marketing department of Europ Assistance MAY process name, email FOR online competition.

5.3.1.4. General applicable rules within Docticare

The following rules are taken form the Privacy Statement of Docticare and therefore they have to be involved in the scenario (full version of the statement in Section 8.4):

- Europ Assistance MAY process Personal not sensitive data FOR the management of the Docticare Services IF the data subject register himself on the portal or IF the data subject has subscribed a Docticare contract.
- Europ Assistance MAY process Personal data (including sensitive ones) FOR assuring compliance to legal requirements, regulations and/or order of government bodies IF the law or regulation prescribe the process or IF there is an order of government bodies.
- Europ Assistance MAY process Name, surname, address, email address, telephone number, FOR commercial information and promotion of the Europ Assistance Service IF personal consent is given.
- Europ Assistance MAY process Personal data FOR customer satisfaction surveys IF personal consent is given.
- EuropAssistance MAY communicate Personal data (including sensitive ones) to determinate subjects FOR providing key services of Docticare.
- Europ Assistance MAY communicate Personal data (including sensitive ones) to all government bodies IF formally requested for them.
- Employees and partners within EuropAssistance MAY process Personal data FOR providing key services of Docticare.
- Europ Assistance MAY NOT disseminate personal data.
- Data Subject MAY request personal data FOR updating, rectifying or objecting to the processing IF legitimate.

5.4. Next steps in the PRDL development

The next steps that will be taken to further develop and specialize PRDL towards the scenarios and real world examples for personal data processing system, are to implement a complete test scenario. Within the scenario the rules will be used to model a real world privacy statement (e.g from Docticare) and to test this setup within a testing environment. There will be important on the expressiveness of the language, maybe illustrate insufficiencies or other problems. The ongoing effort will lead into a final PRDL specification in April, 2012.

6. Conclusion

The deliverable at hand presents the ongoing work on PRDL and shows how the ideas and inputs of the preliminary requirements deliverable are formed into a language. Essential in this process was the bidirectional communication with the ENDORSE project consortium in terms of trial scenario development, further language inputs and reality checks concerning existing languages. Finally the deliverable presents the initial steps towards a framework which ensures the fair and lawful usage of personal data in organizations.

Bridging the gap from the requirements deliverable the first Section deals with candidate languages and their impact in the computer law society as well as requirements and goals these languages focused on. One thing that all languages had in common was the usage of XML based schema as a base for the language implementation. Although a lot of work was conducted in the area of legal normalization and computerization non of the candidate languages complied with the requirements that PRDL had. Therefore the decision to create an own XML based language as a first step was taken [6,7,8,9].

As the development of a new language always involves the creation of an editing environment an XML based language and XML editing environments can never be seen as user-friendly. Initially PRDL was envisaged as a language that is useable even for non technical endusers outside of the information technology business. To meet these requirement additional syntax templates which are parseable into PRDL XML were created. These can be included into the first editor prototype enable the real world testing of the language.

The template development is not finished and there will certainly some elements or even new templates added to enable a complete coverage of occurring use cases. To emphasize the valid definition and interpretation of the language elements that occur in the templates a Glossary was created. All the relevant elements were added and the ENDORSE consortium added the definitions that are considered as the right ones in the context of the project. The glossary describes all the relevant terms also from different science perspectives completely.

Many examples for rules written in the described syntax templates are given throughout the deliverable and should assure that the foci of the language are clear to the reader. To cover the main goal of PRDL which is to describe rules which can be executed with a rule engine, a further step was unavoidable. As time and men-power are not sufficient at the moment to create a proprietary rule engine an existing one has to be used. Therefore it was decided to transfer the XML represented rules into the Drools rule language as a first step towards a working prove of concept. As Drools is fully generic in terms of used meta model is was ideal for a first step towards a fully functional rule language. The next steps will be to show that all the relevant constructs needed for implementing privacy policies in PRDL can be covered by Drools or maybe another rule engine. This analysis and tryouts will be presented in the Deliverable 4.3 which is due in October, 2011.

Concluding this deliverable presents the way that PRDL has evolved from a language defined by the elements and constructs needed to represent privacy policies in a computerized way into a XML based language which can be written using syntax templates and is parseable into an existing rule language. The future work will include the development of a full rule live cycle starting with the creation of rules through a legal person over to the execution within the rule engine and all consequences triggered from the rules.

7. References

[1] Yagüe, M.I. – *Survey on XML-Based Policy Languages for Open Environments,* Journal of Information Assurance and Security 1 (2006) 11-20.

[2] Elisa Bertino, Silvana Castano, and Elena Ferrari. 2001. On specifying security policies for web documents with an XML-based language. In *Proceedings of the sixth ACM symposium on Access control models and technologies* (SACMAT '01). ACM, New York, NY, USA, 57-65.

[3] Bertino, E.; Castano, S.; Ferrari, E.; , "Securing XML documents with Author-X," *Internet Computing, IEEE* , vol.5, no.3, pp.21-31, May/Jun 2001.

[4] Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati. 2002. A fine-grained access control system for XML documents. *ACM Trans. Inf. Syst. Secur.* 5, 2 (May 2002), 169-202.

[5] Damiani, E.; Samarati, P.; De Capitani di Vimercati, S.; Paraboschi, S.; , "Controlling access to XML documents," *Internet Computing, IEEE* , vol.5, no.6, pp.18-28, Nov/Dec 2001.

[6] XrML 2.0 Technical Overview V1.0 - An online version is available: http://www.xrml.org/reference/XrMLTechnicalOverviewV1.pdf, last accessed on 28/07/2011.

[7] Open Digital Rights Language (ODRL) V1.1 - An online version is available: http://odrl.net/1.1/ODRL-11.pd) (June 10, 2011)

[8] Moritz Y. Becker, Cedric Fournet, and Andrew D. Gordon, SecPAL: Design and Semantics of a Decentralized Authorization Language, in *Journal of Computer Security (JCS)*, vol. 18, no. 4, pp. 597--643, IOS Press, 2010.

[9] Moritz Y. Becker, Microsoft Research, Secpal, http://research.microsoft.com/en-us/projects/secpal/, last accessed on 28/07/2011.

[10] ENDORSE Deliverable D2.4 Language Requirements Specification

[11] Michiharu Kudo and Satoshi Hada. 2000. XML document security based on provisional authorization. In *Proceedings of the 7th ACM conference on Computer and communications security* (CCS '00), Pierangela Samarati (Ed.). ACM, New York, NY, USA, 87-96.

[12] OASIS Security Services (SAML) TC, http://www.oasis-open.org/committees/security/, last accessed on 28/07/2011.

[13] eXtensible Access Control Markup Language, OASIS Standard, Tim Moses, 2005, Available at : http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf, last accessed on 28/07/2011.

[14] Mariemma I. Yagüe, Antonio Maña, Javier Lopez - A metadata-based access control *model for web services* , Internet Research, Vol. 15, No. 1. (2005), pp. 99-117

[15] Merriam-Webster Dictionary and Thesaurus, http://www.merriam-webster.com/, last accessed on 28/07/2011.

[16] The Platform for Privacy Preferences 1.1 Specification, W3C Working Group Note, 13 November 2006. http://www.w3.org/TR/P3P11, last accessed on 28/07/2011.

[17] Legal Requirements Acquisition for the Specification of Legally Compliant Information Systems, Breaux, 2009: http://www.cs.cmu.edu/~breaux/intothewild/tdbreaux-thesis.pdf, last accessed on 28/07/2011.

[18] A privacy-aware access control system, Ardagna, 2009: http://spdp.dti.unimi.it/papers/ACDS-JCS2008.pdf, last accessed on 28/07/2011.

[19] Austrian federal act concerning Protection of Personal Data, DSG, 2000: http://www.ris.bka.gv.at/RisInfo/LawList.pdf, last accessed on 28/07/2011.

[20] An Introduction to Syntax, Robert D. van Valin, Jr, State University of New York, Buffalo, Cambridge University Press, 2001.

[21] Language and Law – new applications of formal linguistics, Grewendorf G., Rathert M. in Formal Linguistics and Law, Berlin, New York (Mouton de Gruyter) 2009

[22] Drools - The Business Logic integration Platform, http://www.jboss.org/drools, last accessed on 28/07/2011.

8. APPENDIX

8.1. PRDL XSD Schema

This section shows the XSD schema provided for the PRDL language. As PRDL is under active development the XSD model may be changed to face future challenges. The most up to date version of the schema can be retrieved from http://dbe.fh-salzburg.ac.at/~cruecker/prdl.xsd.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
	<xs:element name="Action" type="Action"/>
	<xs:complexType name="Action">
		<xs:complexContent>
			<xs:extension base="DynamicRuleAttribute">
				<xs:sequence/>
			</xs:extension>
		</xs:complexContent>
	</xs:complexType>
	<xs:element name="Anonymize" type="Anonymize"/>
	<xs:complexType name="Anonymize">
		<xs:complexContent>
			<xs:extension base="Action">
				<xs:sequence/>
			</xs:extension>
		</xs:complexContent>
	</xs:complexType>
	<xs:element name="Condition" type="Condition"/>
	<xs:complexType name="Condition">
		<xs:complexContent>
			<xs:extension base="DynamicRuleAttribute">
				<xs:sequence>
					<xs:element name="ConditionStatement"
type="ConditionStatement"/>
				</xs:sequence>
			</xs:extension>
		</xs:complexContent>
	</xs:complexType>
	<xs:element name="ConditionStatement" type="ConditionStatement"/>
	<xs:complexType name="ConditionStatement">
		<xs:sequence/>
	</xs:complexType>
	<xs:element name="Constraint" type="Constraint"/>
	<xs:complexType name="Constraint">
		<xs:complexContent>
			<xs:extension base="StaticRuleAttribute">
				<xs:sequence/>
			</xs:extension>
		</xs:complexContent>
	</xs:complexType>
	<xs:element name="Copy" type="Copy"/>
	<xs:complexType name="Copy">
		<xs:complexContent>
			<xs:extension base="Action">
				<xs:sequence/>
			</xs:extension>
		</xs:complexContent>
	</xs:complexType>
	<xs:element name="CronAttribute" type="CronAttribute"/>
	<xs:complexType name="CronAttribute">
```

```
        <xs:complexContent>
            <xs:extension base="DynamicRuleAttribute">
                <xs:sequence/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="DataController" type="DataController"/>
    <xs:complexType name="DataController">
        <xs:complexContent>
            <xs:extension base="StaticRuleAttribute">
                <xs:sequence>
                    <xs:element name="DataProcessor" type="DataProcessor"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="DataObject" type="DataObject"/>
    <xs:complexType name="DataObject">
        <xs:complexContent>
            <xs:extension base="StaticRuleAttribute">
                <xs:sequence>
                    <xs:element name="DataSubject" type="DataSubject"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="DataProcessor" type="DataProcessor"/>
    <xs:complexType name="DataProcessor">
        <xs:sequence>
            <xs:element name="name" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="DataSubject" type="DataSubject"/>
    <xs:complexType name="DataSubject">
        <xs:complexContent>
            <xs:extension base="StaticRuleAttribute">
                <xs:sequence/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="Delete" type="Delete"/>
    <xs:complexType name="Delete">
        <xs:complexContent>
            <xs:extension base="Action">
                <xs:sequence/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="DynamicRuleAttribute" type="DynamicRuleAttribute"/>
    <xs:complexType name="DynamicRuleAttribute">
        <xs:sequence>
            <xs:element name="id" type="xs:int" minOccurs="0"/>
            <xs:element name="desc" type="xs:string" minOccurs="0"/>
            <xs:element name="type" type="xs:string" minOccurs="0"/>
            <xs:element name="name" type="xs:string" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="Effect" type="Effect"/>
    <xs:complexType name="Effect">
        <xs:complexContent>
            <xs:extension base="DynamicRuleAttribute">
                <xs:sequence/>
            </xs:extension>
```

```
                </xs:complexContent>
        </xs:complexType>
        <xs:element name="Instrument" type="Instrument"/>
        <xs:complexType name="Instrument">
                <xs:complexContent>
                        <xs:extension base="StaticRuleAttribute">
                                <xs:sequence/>
                        </xs:extension>
                </xs:complexContent>
        </xs:complexType>
        <xs:element name="Location" type="Location"/>
        <xs:complexType name="Location">
                <xs:complexContent>
                        <xs:extension base="StaticRuleAttribute">
                                <xs:sequence/>
                        </xs:extension>
                </xs:complexContent>
        </xs:complexType>
        <xs:element name="Modality" type="Modality"/>
        <xs:complexType name="Modality">
                <xs:complexContent>
                        <xs:extension base="StaticRuleAttribute">
                                <xs:sequence/>
                        </xs:extension>
                </xs:complexContent>
        </xs:complexType>
        <xs:element name="Modify" type="Modify"/>
        <xs:complexType name="Modify">
                <xs:complexContent>
                        <xs:extension base="Action">
                                <xs:sequence/>
                        </xs:extension>
                </xs:complexContent>
        </xs:complexType>
        <xs:element name="PRDLPolicy" type="PRDLPolicy"/>
        <xs:complexType name="PRDLPolicy">
                <xs:sequence>
                        <xs:element name="PRDLRule" type="PRDLRule"/>
                </xs:sequence>
        </xs:complexType>
        <xs:element name="PRDLPolicySet" type="PRDLPolicySet"/>
        <xs:complexType name="PRDLPolicySet">
                <xs:sequence>
                        <xs:element name="PRDLPolicy" type="PRDLPolicy"/>
                </xs:sequence>
        </xs:complexType>
        <xs:element name="PRDLRule" type="PRDLRule"/>
        <xs:complexType name="PRDLRule">
                <xs:sequence>
                        <xs:element name="DataController" type="DataController" minOccurs="1"
maxOccurs="unbounded"/>
                        <xs:element name="DataSubject" type="DataSubject" minOccurs="0"
maxOccurs="unbounded"/>
                        <xs:element name="Modality" type="Modality" minOccurs="1"
maxOccurs="unbounded"/>
                        <xs:element name="Action" type="Action" minOccurs="1"
maxOccurs="unbounded"/>
                        <xs:element name="DataObject" type="DataObject" minOccurs="1"
maxOccurs="unbounded"/>
                        <xs:element name="Purpose" type="Purpose" minOccurs="1"
maxOccurs="unbounded"/>
                        <xs:element name="Constraint" type="Constraint" minOccurs="0"
maxOccurs="unbounded"/>
```

```
                        <xs:element name="Instrument" type="Instrument" minOccurs="0"
maxOccurs="unbounded"/>
                        <xs:element name="Location" type="Location" minOccurs="0"
maxOccurs="unbounded"/>
                        <xs:element name="Condition" type="Condition" minOccurs="0"
maxOccurs="unbounded"/>
                        <xs:element name="State" type="State" minOccurs="0" maxOccurs="unbounded"/>
                        <xs:element name="Effect" type="Effect" minOccurs="0" maxOccurs="unbounded"/>
                        <xs:element name="CronAttribute" type="CronAttribute" minOccurs="0"
maxOccurs="unbounded"/>
                </xs:sequence>
        </xs:complexType>
        <xs:element name="Purpose" type="Purpose"/>
        <xs:complexType name="Purpose">
                <xs:complexContent>
                        <xs:extension base="StaticRuleAttribute">
                                <xs:sequence/>
                        </xs:extension>
                </xs:complexContent>
        </xs:complexType>
        <xs:element name="Read" type="Read"/>
        <xs:complexType name="Read">
                <xs:complexContent>
                        <xs:extension base="Action">
                                <xs:sequence/>
                        </xs:extension>
                </xs:complexContent>
        </xs:complexType>
        <xs:element name="State" type="State"/>
        <xs:complexType name="State">
                <xs:complexContent>
                        <xs:extension base="DynamicRuleAttribute">
                                <xs:sequence/>
                        </xs:extension>
                </xs:complexContent>
        </xs:complexType>
        <xs:element name="StaticRuleAttribute" type="StaticRuleAttribute"/>
        <xs:complexType name="StaticRuleAttribute">
                <xs:sequence>
                        <xs:element name="id" type="xs:int" minOccurs="0"/>
                        <xs:element name="desc" type="xs:string" minOccurs="0"/>
                        <xs:element name="type" type="xs:string" minOccurs="0"/>
                        <xs:element name="name" type="xs:string" minOccurs="0"/>
                </xs:sequence>
        </xs:complexType>
        <xs:element name="Transfer" type="Transfer"/>
        <xs:complexType name="Transfer">
                <xs:complexContent>
                        <xs:extension base="Action">
                                <xs:sequence/>
                        </xs:extension>
                </xs:complexContent>
        </xs:complexType>
        <xs:element name="Update" type="Update"/>
        <xs:complexType name="Update">
                <xs:complexContent>
                        <xs:extension base="Action">
                                <xs:sequence/>
                        </xs:extension>
                </xs:complexContent>
        </xs:complexType>
</xs:schema>
```

8.2. Complete rule analysis of D2.1

This Section includes all the rules that were created out of the first trial scenario deliverable 2.1. They are the basis for the presented rule templates and the analysis of what terms are needed to express data protection rules.

- Register to the Docticare Website

  Rules within the registration process:

  1. The user MUST state his/her name TO register to Docticare.
     a. The name MUST BE his/her real one.
  2. The user MUST state his/her surname TO register to Docticare.
     a. The surname MUST BE his/her real one.
  3. The user MUST state his/her e-mail address TO register to Docticare.
     a. The email address MUST BE the users real one.
     b. The email address MUST BE valid.
     c. The email address MUST BE validated by the user.
     d. The email address MUST BE unique within the system.
  4. The user MUST state his/her password TO register to Docticare.
     a. The password MUST BE 6 and more characters long.
     b. The password MUST INLCUDE numbers and letters.
     c. The password MUST BE secure in terms of DP law.
  5. The user MUST state his/her nickname TO register to Docticare.
     a. The nickname MUST BE unique within the system.
     b. The nickname MUST BE not the users name.
     c. The nickname MUST BE 3 and more characters long.
  6. The user has to state one privacy level
     a. The privacy level MUST INCLUDE an accurate description of the company way to handle the data.
  7. The user may state other privacy fields.
  8. The user may state his/her gender.
     a. The gender MUST BE the users real one.

- Login to the Docticare Website

  The user has to login onto the website of Docticare to use the services.

  1. Rules within the Login Process
     - The user MUST state his/her valid username or valid email address.
     - The user MUST state his/her own password.
     - The user MAY view his/her services for checking new offers IF already logged in.
  2. Update personal data on the Docticare Website

User can edit and update the personal data stored on Docticare website.

- The user MUST state his/her valid username or valid email address FOR logging into the system.
- The user MUST state his/her valid password FOR logging into the system.
- The user MAY change his/her personal data FOR being up to date IF personal data has changed.
- The user MAY add additional information FOR building up his/her profile IF a new medical record was created.

3. Recover the password of your login at the Docticare Website

The user can recover the password for the login process.

- The user MUST state his / her valid email address FOR recovering his/ her password IF the password got lost.
- The user MUST change the temporal password received FOR re-establishing security within the system.
- IF a password reset OCCURS the user MUST change the temporal password on his/her first login.

4. Compile the Medical Passport

The user can compile a medical passport to show the relevant information to his/her doctor.

- The user MAY edit / retrieve medical passport data FOR creating a medical passport IF logged in.
- The user MAY add medical passport data FOR enhancing the medical passport IF logged in.
- The user MAY retrieve his/her medical passport FOR presenting it to a doctor IF logged in.
- The user MAY retrieve a token FOR granting access to his /her medical passport to a doctor.
- The doctor MAY VIEW the users medical passport FOR making a diagnosis IF the token is valid.
- The token MUST HAVE an expiration period of two months.

5. Retrieve passport information

The doctor can also view the passport information when provided a token form the patient.

- The doctor MAY VIEW the medical passport FOR making a diagnosis IF the patient grants access via token.
- IF a valid token login OCCURS the doctor MAY VIEW the medical passport FOR making a diagnosis IF the patient has granted the access.

6. Update Medical File on Docticare

The user can edit or retrieve medical file data that is an extension of the medical passport include also documents.

- The user MAY update his/her medical file within the Docticare system FOR maintaining data actuality.
- The user MAY delete his/her medical file within the Docticare system FOR preventing wrong usage of the data.
- The user MAY upload additional files FOR enhancing his/her medical file.

7. Retrieve Medical File Data

The user can retrieve the medical file data that is an extension of the medical passport.

- The user MAY retrieve his / her medical file FOR showing it to a doctor IF logged in.
- The user MAY retrieve his / her medical file FOR collecting all relevant data IF logged in.

8. Access Data and Add Notes

The data controller can access the data to add or update notes.

- The data controller MAY view the data FOR adding a not IF sufficient credentials.
- The data controller MAY view persons data file FOR editing a note IF sufficient authenticated.

9. Add an Appointment

The user can add and edit appointments in his / her personal agenda

- The user MAY view his/her data FOR adding an appointment IF logged in.
- The user MAY view his/her personal data FOR editing an appointment IF logged in.

10. Editing an appointment from operators' side.

Operator can enter into the agenda upon request of User DS and, if requested by the User DS fix the appointment.

- The data operator MAY view person's data FOR changing an appointment IF sufficient authenticated.
- IF user request for altering an appointment OCCUR the data operator MAY view the users personal data FOR altering the appointment IF sufficient authenticated.

11. Asking for a second opinion to a doctor

The user can ask a doctor for a second opinion about a medical issue. She has to fill in a form for that.

- The user MAY provide his data FOR asking for a second medical opinion IF logged in.
- The data subject MUST state additional information FOR asking for a second medical opinion.
- The user MUST enter the phone number of a second doctor.
- The user MUST enter the description of the request.

12. Answering to User DS

The user request for second opinion is sent the user operator. The email is filtered by the User OP and sent to the User Doctor. He directly responds to the user DS via email.

- IF a user request for a second medical opinion OCCURS the user operator MAY inform the user doctor FOR answering the request IF sufficient authenticated.
- IF a user operator request OCCURS the user data doctor MAY create the second opinion FOR answering the request IF sufficient authenticated.

13. Ask for booking a medical exam

The user can ask the user OP for the booking of a medical exam.

- The user MUST state additional information FOR booking an exam IF logged in.
- The user MUST state the preferred city in which to undergo the exam FOR booking an exam IF logged in.
- The user MUST state the preferred medical structure in which to undergo the exam FOR booking the exam IF logged in.
- The user MUST state the type of exam FOR booking an exam IF logged in.
- The user MUST state the description of the exam FOR booking an exam IF logged in.

14. Booking a medical exam

The user operator receives the request and books the exam.

- IF a request for an exam OCCURS the user operator books the exam FOR answering the request IF sufficient authenticated.
- The user operator MUST mark the requested exam as booked FOR informing the user DS IF sufficient authenticated.
- IF the a not bookable exam OCCURS the user OP MUST inform the user DS FOR negotiating a new exam IF sufficient authenticated.

15. Chat with User DC

The user DS wants to book a chat session with a doctor. The user operator sets the timeslot for the chat. The user DS has the opportunity to store the chat afterwards.

- The user DS MAY book a chat session FOR consulting a doctor IF logged in.
- The user operator MAY view the users personal data FOR booking a chat session IF sufficient authenticated.

16. Asking for a personalized check up to a doctor

The user can fill in a form to ask for a personalized check up.

- The user MUST state his / her weight FOR getting a personalized check-up IF logged in.
- The user MUST state his / her height FOR getting a personalized check-up IF logged in.
- The user MUST state his / her medical history FOR getting a personalized check-up IF logged in.
- The user MUST state his/ her present medical situation FOR getting a personalized check-up IF logged in.
- The user MAY state additional information FOR getting a personalized check-up IF logged in.

17. Answering to User DS (internal)

- IF a personalized check-up request OCCURS EurA MAY use the email address FOR contacting the user IF user explicitly agreed.

8.3. Privacy statement of Docticare including rules

The Docticare platform will be the playground for the first ENDORSE framework prototype. Therefore the personal data statement of Docticare is the first choice to derive relevant rules from. These rules are shown in the section with the original text to argue the origin. The scenarios that will be implement in the first prototype will include these rules and additional scenario rules as well.

Docticare Privacy Statement

Pursuant to Legislative Decree 30 June 2003, n. 196 - Privacy Code regarding data protection, you are advised that:

1. your personal data, (including sensitive ones) will be processed by Europ Assistance Service S.p.A. in hardcopy, electronic and/or automatic medium, for purposes involving:

   a) the management of the Docticare services

   Europ Assistance MAY process Personal not sensitive data FOR the management of the Docticare Services IF the data subject register himself on the portal or IF the data subject has subscribed a Docticare contract.

   or

Europ Assistance MAY process personal sensitive data FOR the management of the Docticare Services IF personal consent is given.

b) complying with legal requirements, regulations or Community legislation and/or orders from government bodies;

Europ Assistance MAY process Personal data (including sensitive ones) FOR assuring compliance to legal requirements, regulations and/or order of government bodies IF the law or regulation prescribe the process or IF there is an order of government bodies.

c) with regard to the only personal data as: name, surname, address, email address, telephone number, commercial information and promotion of the services of the Europ Assistance Service.

Europ Assistance MAY process Name, surname, address, email address, telephone number, FOR commercial information and promotion of the Europ Assistance Service IF personal consent is given.

d) customer satisfaction surveys

Europ Assistance MAY process Personal data FOR customer satisfaction surveys IF personal consent is given.

2. data processing will be undertaken that is:
   a) necessary for the management of services offered by the website www.docticare.it
   b) mandatory based on law, regulation or Community legislation and/or provisions of public bodies
   c) optional for purposes of conducting activities involving commercial information and promotion of services and service customer satisfaction surveys (1.c, 1.d e 1.e);

3. the data may be communicated to the following subjects as autonomous data controllers:
   a) a. specific subjects charged by Europ Assistance Service S.p.A. with providing key services necessary for execution of the Docticare services, such as — by way of example — subjects charged with managing files and processing data, credit institutions, experts, medical examiners;
   b) EuropAssistance MAY communicate Personal data (including sensitive ones) to determinate subjects FOR providing key services of Docticare)
   c) b. judicial authorities, as well as all government bodies to which disclosure may be given if formally requested;
   d) Europ Assistance MAY communicate Personal data (including sensitive ones) to all government bodies IF formally requested for them.

4. In addition your data may become known by employees and partners acting as data processor or as persons in charge of the processing.

   Employees and partners within EuropAssistance MAY process Personal data FOR providing key services of Docticare.

5. The data are not subject to dissemination.

   Europ Assistance MAY NOT disseminate personal data

6. The Data Controller is Europ Assistance Service S.p.A. You may request the list of the data processors, obtain information regarding your personal data and also update, rectify, and object to the processing of your data on legitimate grounds by writing to:

   Data Subject MAY request personal data FOR updating, rectifying or objecting to the processing IF legitimate.

Europ Assistance Service S.p.A. - Piazza Trento, 8 - 20135 Milan – Italy

Ufficio Protezione Dati.

UfficioProtezioneDati@europassistance.it