Linked2Safety

FP7-288328

A Next-Generation, Secure Linked Data Medical Information Space for Semantically-Interconnecting Electronic Health Records and Clinical Trials Systems Advancing Patients Safety in Clinical Research

---

Deliverable D4.1

Linked Medical Data Space (LMDS) Design

Editor(s):	Ratnesh Sahay (NUIG), Yasar Khan (NUIG), Ronan Fox (NUIG), Dimitrios Ntalaperas (UBITECH), Panagiotis Hasapis (INTRASOFT), Eleni Kamateri (CERTH), Eleni Panopoulou (CERTH)
Responsible Partner:	NUIG
Status-Version:	Final – v1.0
Date:	21/12/2012
EC Distribution:	Public

Project Number:	FP7-288328
Project Title:	Linked2Safety

Title of Deliverable:	D4.1 – Linked Medical Data Space Design
Date of Delivery to the EC:	

Workpackage responsible for the Deliverable:	WP4 – Linked2Safety Linked Medical Data Space
Contributor(s):	NUIG, CERTH, UBITECH, INTRASOFT
Reviewer(s):	SIVECO
Approved by:	All Partners

Abstract:	The objective of this document is to present a detailed design specification for the Linked Medial Data Space (LMDS). The design specification is based on the D1.2 Linked2Safety Reference Architecture and requirements analysis conducted in the D1.1.
Keyword List:	Healthcare/EHR standards, Clinical Repositories, Linked Data, Data-Cubes, Semantic Web Technologies, Ontology, Security and Access Policy.

Document Description

Document Revision History

Version	*Date*	*Modifications Introduced*	
		Modification Reason	*Modified by*
v0.1	07/08/2012	Initial planning and high level sketch	NUIG
v0.2	05/09/2012	TOC definition	NUIG
V0.3	06/09/2012	Detailed framework definition	NUIG
V.04	31/10/2012	TOC definition and task allocation distributed to the deliverable partners	NUIG
V.05	18/11/2012	Initial sketch of the policy model	NUIG, CERTH
v.06	10/12/2012	Sections 7 and 8 received by the deliverable partners	CERTH, UBITECH, INTRASOFT
v.07	11/12/2012	All sections merged	NUIG
v.08	14/12/2012	Draft version circulated for internal review	NUIG
v.09	17/12/2012	Internal review feedback received	All partners
v.1.0	21/12/2102	Final version submitted	NUIG

Contents

List of Figures

List of Tables

Definitions, Acronyms and Abbreviations

Table 1: Definitions, Acronyms and Abbreviations

Acronym	Title
ADE	Adverse Drug Event
ADL	Archetype Definition Language
AE	Adverse Event
AOM	Archetype Object Model
API	Application Programming Interface
AQL	Archetype-Based Query Language
ATAM	Architecture Tradeoff Analysis Method
CDA	Clinical Document Architecture
CPR	Computerised Patient Record
DSM-IV	Diagnostic and Statistical Manual of Mental Disorders, Fourth Edition
EDC	Electronic Data Capture
EHR	Electronic Health Record
EMR	Electronic Medical Record
EPR	Electronic Patient Record
HCLS	Healthcare and Life Sciences
HIMSS	Healthcare Information and Management Systems Society
HL7	Health Level Seven
ICEHR	Integrated Care HER

LD	Linkage Disequilibrium
MAF	Minor Allele Frequency
NFR	Non-Functional Requirements
OLAP	Online Analytical Processing
OWL	Web Ontology Language
PHR	Personal Health Record
RAIS	Requirements, Architecture, Interoperability Issues and Solutions
RDF	Resource Description Framework
RDFS	RDF Schema
RR	Reporting Ratio
SNOMED	Systematised Nomenclature of Medicine
SNP	Single Nucleotide Polymorphism
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
US	Usage Scenarios
XML	Extensible Markup Language

1. Executive Summary

The present document is Deliverable 4.1 "D4.1 – Linked Medical Data Space (LMDS) Design" (henceforth referred to as D4.1) of the Linked2Safety project. The Linked Medical Data Space (LMDS) is a semantically-interlinked platform enabling secure and policy driven data access mechanism for exploiting clinical data-cubes that originate from disparate and diverse clinical sources.

The goal of LMDS is to enable seamless sharing and linking of clinical data-cubes on top of a secure framework ensuring access to clinical resources by the authorised stakeholders. The LMDS has a crucial role of sharing consistent and meaningful data/knowledge for decision making in medical and clinical research domains. In the Linked2Safety architecture, the LMDS resides between existing heterogeneous clinical resources and the intelligent decision making statistical data mining space. Further, the LMDS design specification will be implemented and evaluated as part of the D4.2.1/D4.2.2 Linked Medical Data Space.

The purpose of LMDS is to allow the clinical research community a homogenised access to the anonymised clinical data/knowledge needed to perform complex statistical data mining operations. The published data-cubes within the LMDS adhere to the Linked Data principles. Therefore, the LMDS data is available as structured data using dereferencable URIs in a machine-interpretable format. The tools developed and used within the LMDS comply with standard technologies such RDF, OWL, and SPARQL.

The five main sub-spaces of the LMDS are: (i) the **storage framework** ensures that the clinical data (i.e., data-cubes) can be uploaded, accessed, and explored for analysis and hypothesis testing; (ii) the **linking component** semantically-interlinks data-cubes originating from various clinical partners, and also links them with the Linked Data cloud (LOD); (iii) the **querying component** federates user's queries over the distributed data-cubes and combines results; (iv) the **policy model** allows describing restrictions as per the user's roles and applies user-based restrictions in accessing the clinical data-cubes; and finally (v) the **security and authentication component** employs a user-authentication mechanism in accessing the clinical data-cubes.

The LMDS design specification presented in this deliverable involves both the technical and legal partners of the consortium which further ensures the safety and security aspects of the sensitive clinical resources. Finally, the maintenance and evaluation of the LMDS will be carried throughout the Linked2safety project.

2. Introduction

This section presents the work pursued within the Task 4.1, named "Linked Medical Data Space (LMDS) Design". The LMDS interlinks anonymised medical (i.e., EHR observations) and Life Sciences data originating from various clinical partners, and also links them with the Linked Data Cloud (LOD). Access to this anonymised data is governed by adaptable access policies represented by the Linked2Safety Access Policy Model. The Section 2.1 describes the scope and the key objectives which have guided this work. The Section 2.2 highlights the motivation behind developing this semantically-interlinked platform. Finally, Section 2.3 presents the overall work performed in Task 4.1 and describes the structure of this document.

2.1. Document Scope

The present document is Deliverable 4.1 "D4.1 – Linked Medical Data Space (LMDS) Design" of the Linked2Safety project. The main objective of this document is to define, design, and confirm the security and safety aspects of the LMDS. The work presented in this deliverable is based on the Reference Architecture definition area of *"The Linked Medical Data Space"* discussed in the deliverable D1.2 Linked2Safety Reference Architecture.

The deliverable presents the different sub-spaces—storage, linking, querying, policy, and security—that build the LMDS. Moreover, it covers the design specification driven by an integration scenario. The design specification and the sub-spaces provided in this deliverable will be further referenced within the context of D4.2.1/D4.2.2 Linked Medical Data Space.

2.2. Motivation

The main motivation to build the LMDS is to resolve heterogeneities among resources originating from clinical research and EHRs data spaces. To preserve the anonymisation and security of patients (and related stakeholders) these heterogeneous resources are tailored as data-cubes within the LMDS. Further, the LMDS will use the Semantic EHR Model (developed as part of the Task 1.4) to provide shared and consistent ontological reference point for the heterogeneous data-cubes originating from clinical partners.

The above motivation to resolve heterogeneous clinical resources by a semantically-interlinked platform derives from the functional and non-functional requirements presented in the Requirement Analysis deliverable D1.1. Specifically Table 13 (of the deliverable D1.1 Requirements Analysis) describes the functional requirements collected from the usage scenarios and the semi-structured interviews; and Table 14 (of the deliverable D1.1 Requirements Analysis) non-functional requirements collected from the usage scenarios and the semi-structured interviews. The 5 out of 12 top priority functional requirements require a semantically-interlinked platform in order to be satisfied, and the 4 out of 12 top priority non-functional requirements require a semantically-interlinked platform.

2.3. Document Structure

The remainder of D4.1 is divided into six sections:

Section 3 presents an integration scenario describing the resources, terminologies, and repositories used by the Linked2Safety clinical partners.

Section 4 presents the storage framework describing arrangement of the storage components within the LMDS and technology used to store data-cubes in a context-aware fashion.

Section 5 presents the linking component describing the publication principles for data-cubes and establishing links between internal (i.e., data-cubes from the clinical partners) and external links to the Linked Data Cloud.

Section 6 presents the querying component describing federation of a query over distributed clinical resources. The querying component incorporates a mechanism to prepare an indexing meta-data in order to locate resources that may answer a query.

Section 7 presents the access policy model describing a semantically-enabled policy model that allows user-restricted access to clinical resources.

Section 8 presents the security component describing a user-authentication mechanism on top of the access policy model for accessing sensitive resources.

3. Integration Scenario

Electronic Health Record (EHR) and Life Sciences domains contain an increasing wealth of information. This information has a significant role in the advancement of medical research and personalized healthcare. The issue is effective utilisation of this data, in terms of cost, effort time and success, by different stakeholders in the healthcare and medical research domain, such as clinical researchers, pharmaceutical companies etc.. They require this data for various purposes, such as identification of adverse events, identification of suitable patients to participate in small or large scale clinical trials etc. The main hurdle in the effective utilisation of the data is the lack of interoperability and disconnection between different pieces of data, due to heterogeneity and the use of local standards in the data. Also the data from different domains needs to be semantically interlinked, such as interlinking EHR and Life Sciences databases. To address the interoperability of EHR and Life Sciences data an extensible, scalable architecture is required that will facilitate the semantic interlinking between spatially distributed clinical care information sources, electronic patients' health records and clinical trials systems for gathering and sharing adequate knowledge to support decision making in medical and clinical research.

In the case of Linked2Safety project, there are three clinical partners, namely, the University Hospital of Lausanne (CHUV), the Cyprus Institute of Neurology and Genetics (CING), and the ZEINCRO Hellas S.A.. Each data provider created its own locally defined clinical terminologies and related data set. For instance, CHUV uses terminologies covering demographic characteristics, psychiatric disorder, migraine, and cardiovascular risk factors. CHUV's psychiatric disorder related terminologies partially correspond with the DSM-IV classification. CING uses terminologies along three dimensions: the first set of terminologies correspond to the breast cancer domain, the second to the diabetic domain and the third to the neurogenetics domain. Those variables include family history and genetic data as well as patient characteristics and medical information. ZEINCRO's terminologies cover the respiratory domain. Further, terminologies are categorised along: adverse event (AE), demographic, medical history and concomitant medication. The data maintained by the three clinical partners are isolated from each other. Consequently, the applicability of clinical variables/terminologies are limited to a local context.

In order to achieve the anonymisation of clinical data, we employ the data-cube approach where a selected set of phenotype and genotype describe the structure of a data-cube. A data-cube is an array of data having 0 or more dimensions. An example data-cube is shown in Figure 1.

	Sex	Male			Female		
Drug	BMI > 25 Dyslipidemia	0	1	9	0	1	9
Insulin	0	24	35	4	23	12	1
	1	16	32	0	20	18	2
	9	3	8	0	1	3	0
Digrin	0	12	24	3	15	25	0
	1	19	17	1	22	18	4
	9	4	2	0	1	2	1

Figure 1 : sample data-cube

Figure 1 shows a sample data-cube describing the total number of patients exhibiting either of the two adverse events, broken down by sex and the drug administered. The sample data-cube uses four dimensions: Drug, Sex, Adverse Event 1 (BMI > 25) and Adverse Event 2 (Dyslipidemia). Each observation represents the total number of patients exhibiting a particular adverse event. One of the main obstacle in analysing such data-cubes is the use and interpretation of each dimensions or measures. For example, Figure 2 shows Dataset 1 and Dataset 2 which represent gender attribute of patients as "Sex" while the same attribute is represented as "Gender". Also Dataset 1 and Dataset 3 are using "Drug" and Dataset 2 is using "Dose" for the same concept.

Dataset 1

Sex	Drug	BMI > 25	Dyslipidaemia	Headache	Rash	Positives
male	insulin	0	0	0	0	99
male	insulin	1	1	1	0	9
male	insulin	0	9	1	9	62

Dataset 2

Sex	Dose	SNP	Breast Cancer	Counts
male	insulin	3	1	14
male	insulin	1	3	10
male	insulin	2	0	11

Dataset 3

Gender	Drug	Diabetes	Smoker	Counts
male	insulin	0	0	99
male	insulin	0	0	9
male	insulin	0	0	62

Figure 2: Heterogeneous Data-cubes

Additionally, a clinical researcher can try to combine more than one data-cube for advanced analysis and hypothesis testing. For example, if a clinical researcher has access to only Dataset 1 (Figure 2), it would be impossible to do an accurate analysis of adverse events caused by a specific drug. This is due to the fact that the information regarding the adverse events of insulin drug is not complete defined in Dataset 1. Information about adverse events, such as SNP, Breast Cancer and Diabetes are missing in Datasets 1 but can be obtained from Dataset 2 and Dataset 3 (Figure 2) by integrating all the dataset together in a seamless way.

Key Requirements: considering the heterogeneity and secure access of clinical data-cubes, the following are the key requirements to build a semantically-interlinked platform: (i) standard representation and context-aware storage of data-cubes. It is important to notice from the above discussion that data-cubes are local in nature (i.e., they are tied with local clinical sites). Therefore, it is crucial to exploit their origin while performing any kind of analysis and/or hypothesis testing; (ii) linking data-cubes originating from various clinical partners, and also links them with the Linked Data cloud (LOD); (iii) federation of clinical queries over distributed data-cubes and combine results; (iv) policy based access to user-restricted clinical data-cubes; and finally (v) supporting user-authentication mechanism in accessing the clinical data-cubes.

4.Storage Framework

Figure **3** shows the LMDS components described in D1.2. The five components within the red rectangle: RDF Dump, SPARQL endpoint, Lookup Index, Linked Data API, and RDF Store, comprise the storage framework. The purpose of the storage framework is to ensure that the data (i.e., data-cubes) in the LMDS can be uploaded, explored, reused and integrated for different hypothesis testing and analysis performed by different clinical users.

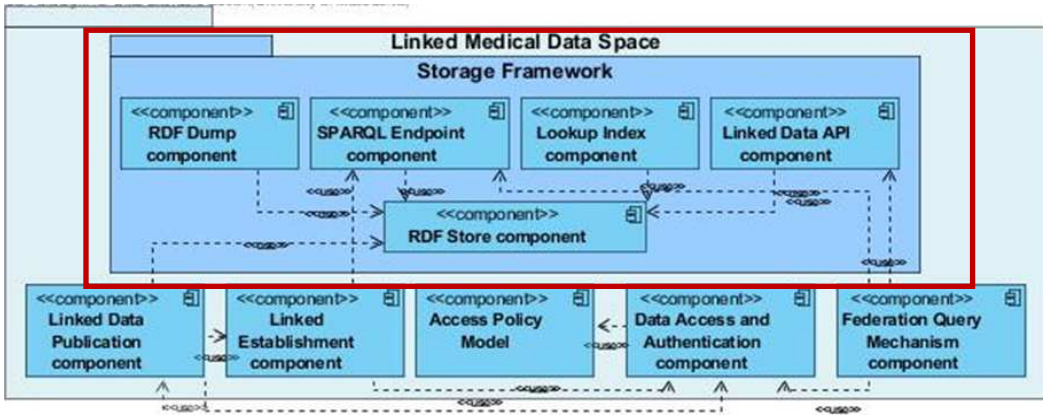

Figure 3: Linked Medical Data Space Architecture: Storage Framework

The main task of the storage framework is to ensure stable access to clinical data-cubes from different access points. This includes making data available in standardised machine readable format such that it can be included in federated queries (as a SPARQL endpoint); through APIs that implement domain specific language that facilitate adoption by software developers; and through a series of querying, analysis and visualisation tools that aid nontechnical domain experts in exploring the content of the LMDS.

In the storage framework (Figure **3**), the RDF store component is the main repository for clinical data-cubes. It provides storage of clinical data-cubes in a standard format i.e., in RDF format. In the LMDS, there will be multiple linked RDF stores originating from various clinical partners. The storage framework provides four access mechanisms to these linked RDF stores: (i) the RDF Dump component provides a mechanism for downloading partial or complete sets of data-cubes; (ii) the Lookup Index component is an intelligent indexing mechanism storing meta-information (e.g., labels, annotations) that may improve linking of clinical RDF data-cubes; (iii) the SPARQL Endpoint component provides

services for querying the linked RDF stores; (iv) the Linked Data API component provides a "Restful Interface" allowing access to data-cubes via SPARQL endpoints. Further, the Linked Data APIs made available through this space are serialised into template SPARQL queries which either are executed at a single SPARQL endpoint or from a variety of domain specific SPARQL endpoints, through a federated SPARQL query, depending on the requirements of the user.

In this section we specially discus two key points of the storage framework (i) storage pre-processing, i.e., building of RDF clinical data-cubes; and (ii) the RDF store. We mentioned above that context-aware storage of data-cubes is essential to record origin of the cubes, i.e., which cube came from which sources. Such context information would be helpful during combining and/or separating data-cubes in order to answer a federated query.

4.1. Building Data-Cubes

This is a pre-processing step before the actual storage of RDF data-cubes. This pre-processing involves transforming heterogeneous (i.e., different storage formats) clinical data into RDF data-cubes. The transformation of clinical data into RDF helps to store data-cubes in standard graph format, however, representing each data-cube as a statistical contingency table still requires a structural definition of each data-cubes. The RDF Data-cube Vocabulary [1] provides a set of vocabularies to represent and publish a multi-dimensional statistical data in a Web-friendly format to enable it to be linked and combined with related information and data sets. Statistical data sets are comprised of observations (values) which are organised along a group of dimensions. RDF data-cubes build using the RDF data-cube vocabulary can further published following the linked data principles. The following section on linking mechanism describes the linked data principles and their use for linking different data-cubes. The RDF data-cube vocabulary organizes data-cubes according to a set of *dimensions*, *attributes,* and *measures*, collectively called components. The *dimension* component identifies the observation, e.g. a disease which the observation covers. A *measure* specifies a phenomenon being observed, e.g. patients exhibiting a particular adverse event. An *attribute* allows the qualification and interpretation of the observed values, e.g. specifying units for the *measure.*

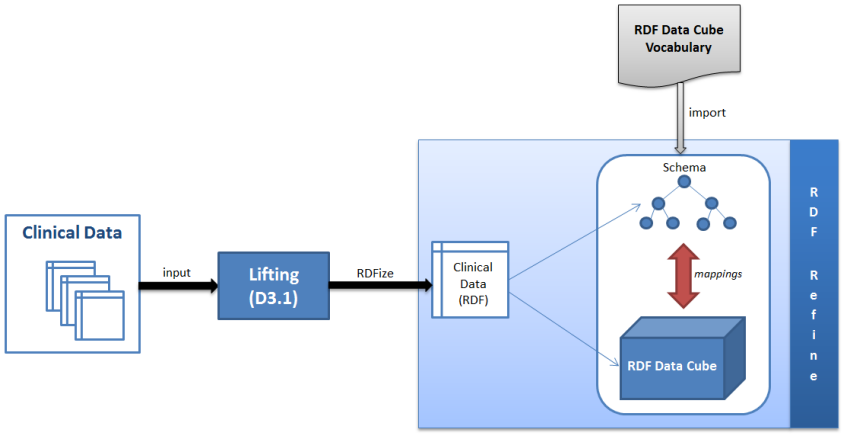

Figure 4: RDF Clinical Data-cube Creation

Figure 4 shows the workflow employed for lifting a syntactic clinical data to the semantic format (i.e., RDF) and building RDF data-cubes. The deliverable 3.1 Interoperable EHR Data Space Design performs both the lifting and building of data-cubes. RDF data-cubes are build using the RDF Refine[1], which is a Google Refine extension for exporting data in RDF format. RDF Refine takes the data sets in various formats as input, such as Excel, CSV, XML, RDF etc. Based to the structure of the data (i.e., schema) in the data set, a corresponding schema (i.e., data-cube structure) is created complying the RDF data-cube vocabulary. In the schema creation process, dimensions, measures and attributes, defined in the RDF data-cube vocabulary, are identified in the data set. The next step then is to store RDF data-cubes in a context-aware repository.

4.2. Context-aware RDF Store

Since data-cubes originate from various clinical partners, it is important to preserve and exploit the identity of a data-cube. Identity here simply means a mechanism to specify the origin of a data-cube. Furthermore, each data-cube can be annotated with additional information (e.g., accessibility, location, provider) about the use and restrictions applied to a data-cube. Figure 5 shows the LMDS storage arrangement for different sets of data-cubes coming from three clinical partners. For instance, data-cubes generated at CHUV are stored in RDF Data Store 1 (Figure 5). RDF data-cubes generated from CING and Zeincro data sets are stored in RDF Data Store 2 and RDF Data Store N, respectively. In the same fashion, new data sets provided by other medical data providers can be integrated in the Linked Medical Data Space.

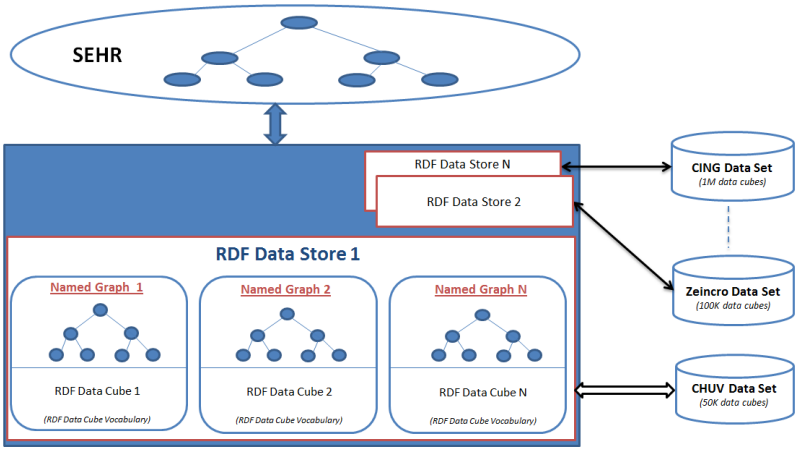

Figure 5: Storage Scenario

Context-awareness is achieved using the RDF Named Graph [2]. The Named Graphs data model is a simple variation of the RDF data model. The basic idea of the model is to introduce a graph naming mechanism, which allows RDF triples to include meta-information (i.e., contextual information) about RDF graphs. A named graph is an entity which consists of an RDF graph and a name in the form of a URI reference, i.e., the URI can be used as an identity of a graph which could be a location, organisation, user, etc. Two named graphs

[1] refine.deri.ie

which have different names but share the same RDF graph are seen as two separate entities. Two named graphs which have different RDF graphs have to be named with different URI references. The RDF data model represents information as a single node-and-edge labelled graph. Within the Named Graphs data model, information is represented as a set of named graphs. A named graph is an RDF resource and can be described in the usual open way using RDF statements. RDF statements about a named graph may occur in the named graph itself or in other graphs. Information which is stated about a named graph is understood to refer to each statement within the graph. For instance, the statement that somebody is the creator of a named graph implies that he is the creator of each statement within the graph. This interpretation provides a simple, but flexible alternative to RDF reification, as it enables meta-information to be stated about graphs containing only a single statement as well as about graphs containing multiple statements. As described above, graphs are uniquely identified by being named with a URI reference, it is possible for different information providers to make statements about a graph.

In the LMDS, if CHUV provides 50K data-cubes, then the corresponding RDF Data Store 1 (for CHUV) will have 50K named graphs stored in it, i.e., one named graph for each data-cube. All these RDF data-cubes, i.e., RDF named graphs are aligned to the SEHR model (developed in D1.4).

```
# Example Named Graph representing RDF Data Cube

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix l2s-ct: <http://linked2safety.hcls.deri.org/clinical-trials/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix qb: <http://purl.org/linked-data/cube#> .
@prefix sdmx-code: <http://purl.org/linked-data/sdmx/2009/code#> .
@prefix l2s-drug: <http://linked2safety.hcls.deri.org/codelist/drug/> .
@prefix sdmx-dimension: <http://purl.org/linked-data/sdmx/2009/dimension#> .


l2s-ct: datacube-graph-2 {

        <http://linked2safety.hcls.deri.org/clinical-trials/dataset2/0/3/1/1/3/3> a qb:Observation ;
            qb:dataSet <l2s-ct:dataset2> ;
            sdmx-dimension:sex <sdmx-code:sex-M> ;
            l2s-prop:drug l2s-drug:insulin ;
            l2s-ct:snp "3"^^xsd:int ;
            l2s-ct:breast-cancer "1"^^xsd:int ;
            l2s-ct:counts "14"^^xsd:int .

        <http://linked2safety.hcls.deri.org/clinical-trials/dataset2/1/1/3/2/2/1> a qb:Observation ;
            qb:dataSet <l2s-ct:dataset2> ;
            sdmx-dimension:sex <sdmx-code:sex-M> ;
            l2s-prop:drug l2s-drug:insulin ;
            l2s-ct:snp "1"^^xsd:int ;
            l2s-ct:breast-cancer "3"^^xsd:int ;
            l2s-ct:counts "10"^^xsd:int .

        <http://linked2safety.hcls.deri.org/clinical-trials/dataset2/2/3/1/1/3/3> a qb:Observation ;
            qb:dataSet <l2s-ct:dataset2> ;
            sdmx-dimension:sex <sdmx-code:sex-M> ;
            l2s-prop:drug l2s-drug:insulin ;
            l2s-ct:snp "2"^^xsd:int ;
            l2s-ct:breast-cancer "0"^^xsd:int ;
            l2s-ct:counts "11"^^xsd:int .

}
```

Figure 6: Example Named Graph

Figure 6 represents an example named graph which represents a RDF data-cube. Each named graph is identified a URI, such as in our example the URI (or identity) for the named graph is the *l2s-ct:datacube-graph-2* name, where *l2s-ct* is a prefix and the prefix can be expanded to its complete form from the prefix section in Figure 6. This named graph represents a set of triples which in turn represent a certain RDF data-cube. In the same manner all the other RDF data-cubes can be represented using the named graphs mechanism.

5. Linking Component

Once RDF data-cubes are stored in a context-aware fashion, then next challenge is to semantically link them based on any similarity or additional information they can provide. Linking disparate biomedical data is a challenge due to inconsistency in naming and heterogeneities in data models and formats. The need for data integration has placed these domains in the forefront of technologies such as ontologies and, more recently, the Linked Data technologies. Data sources with content from these domains constitute a large portion of the 'Linked Data cloud',

and their number is growing quickly. Linked Data is a technology that can be applied for addressing these challenges as it facilitates linking diverse data sources.

Linked Data is about publishing structured data in RDF using URIs and it requires the identification of entities with URI references that can be dereferenced over the HTP protocol into RDF data that describes the identified entity. In addition Linked Data include the creation of typed links between URI references, so that one can discover more data. More specifically, the four Linked Data principles as described by Berners-Lee [3] are the following:

a) All items should be identified using URIs;

b) All URIs should be dereferenceable, that is, using HTTP URIs allows looking up the item identified through the URI;

c) When looking up a URI it should lead to more data.

d) Links to other URIs should be included in order to enable the discovery of more data.

To address the aforementioned challenges in the context of Linked2Safety project, the aim of the LMDS Linking Framework is twofold. First is to publish clinical data-cubes originating from different clinical partners complying with the Linked Data principles and second is to establish and maintain semantic links between data-cubes and then link these data-cubes with other relevant datasets found on the Linked Data Cloud, to provide further enrichment to data-cubes.

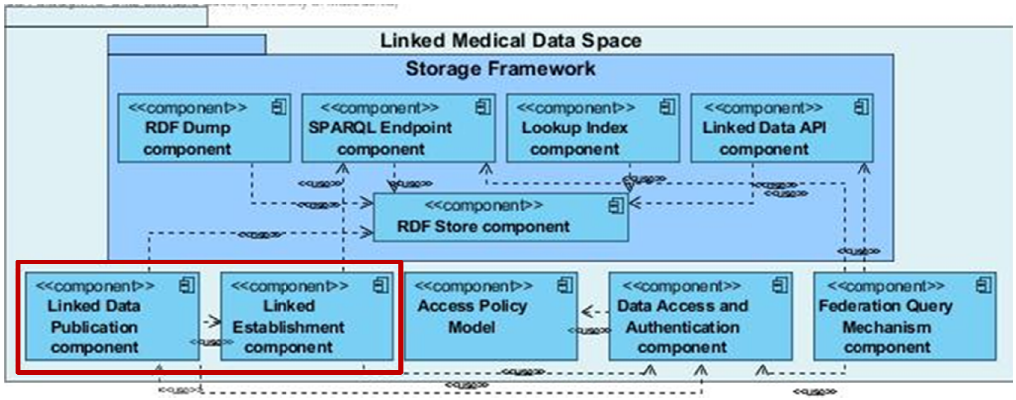

Figure 7: Linked Medical Data Space Architecture: Linking components

Figure 7**:** Linked Medical Data Space Architecture: Linking shows the dependencies of linking components (i.e., Linked Data Publication and Linked Establishment) within the LMDS. The linking components result in the creation of a single linked data space that can be queried with semantically rich and complex queries, which will significantly enhance the process of clinical trials research, such as patient matching with clinical trials and tracking adverse events of drugs. Figure 8 shows a sequence of steps that involves publication and link establishment between internal and external clinical resources.

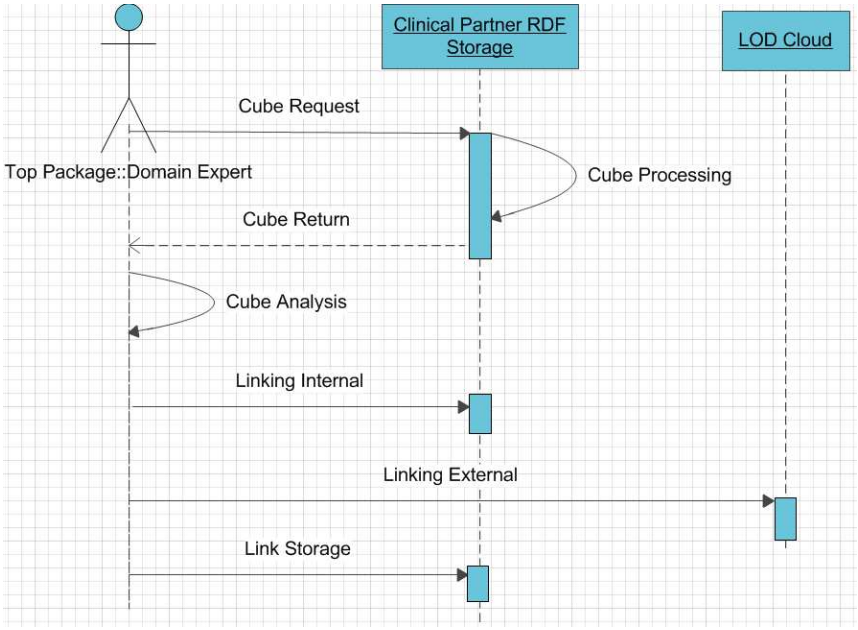

Figure 8: Sequence Diagram for Linking clinical data-cubes

Figure 9 provides a snippet of RDF data-cube. Each entity in the data-cube is identified by a unique HTTP derefrenceable Uniform Resource Identifier (URI). On lookup, these URI's leads one to more data about the entity to which the URI refers, in the form of RDF statements. For example, '<http://linked2safety.hcls.deri.org/data/clinical-trials/M/insulin/0/0/0/0>' is a URI referring to a certain observation entity in the context of a clinical trial. This URI leads one to more data about the observation, such as 'this is an observation about male patients', 'number of patients in this observation are 72', 'patients were taking insulin as a drug', and 'patient body mass index (BMI) is not greater than 25' and so on.

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix qb: <http://purl.org/linked-data/cube#> .
@prefix sdmx-code: <http://purl.org/linked-data/sdmx/2009/code#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix sdmx: <http://purl.org/linked-data/sdmx#> .
@prefix sdmx-dimension: <http://purl.org/linked-data/sdmx/2009/dimension#> .

# --- Own namespaces ------------------------

@prefix l2s: <http://linked2safety.hcls.deri.org/> .
@prefix l2s-data: <http://linked2safety.hcls.deri.org/data/> .
@prefix l2s-dsd: <http://linked2safety.hcls.deri.org/dsd/> .
@prefix l2s-prop: <http://linked2safety.hcls.deri.org/property/> .
@prefix l2s-drug: <http://linked2safety.hcls.deri.org/codelist/drug/> .

l2s-data:clinical-trials a qb:DataSet ;
    qb:structure l2s-dsd:positives-by-adverse-event ;
    rdfs:label "Clinical Trials Dataset"@en ;
    rdfs:comment "Clinical Trials Dataset indicating adverse events seen in a sample population upon administration of two drugs"@en .

<http://linked2safety.hcls.deri.org/data/clinical-trials/M/insulin/0/0/0/0> a qb:Observation ;
    qb:dataSet l2s-data:clinical-trials ;
    l2s-prop:drug l2s-drug:insulin ;
    sdmx-dimension:sex <sdmx-code:sex-M> ;
    l2s-prop:bmi25 "0"^^xsd:int ;
    l2s-prop:dyslipidemia "0"^^xsd:int ;
    l2s-prop:headache "0"^^xsd:int ;
    l2s-prop:rash "0"^^xsd:int ;
    l2s-prop:positives "72"^^xsd:int .

<http://linked2safety.hcls.deri.org/data/clinical-trials/M/insulin/0/0/0/1> a qb:Observation ;
    qb:dataSet l2s-data:clinical-trials ;
    l2s-prop:drug l2s-drug:insulin ;
    sdmx-dimension:sex <sdmx-code:sex-M> ;
    l2s-prop:bmi25 "0"^^xsd:int ;
    l2s-prop:dyslipidemia "0"^^xsd:int ;
    l2s-prop:headache "0"^^xsd:int ;
    l2s-prop:rash "1"^^xsd:int ;
    l2s-prop:positives "38"^^xsd:int .
```

Figure 9: Partial View of RDF Data-cube

Another important aspect of the Linking Framework is to interlink all the data-cubes generated by clinical partners, such as providing typed links between resources in the data-cubes and also to establish links between the datasets in the LMDS and the external medical datasets found on the Linked Data Cloud. Only triplication of data is not enough to get full benefits of RDF based technologies. RDF data-cubes without linking would be like isolated islands of data where each island has only part of the data necessary for answering the query of a user investigating the data. The relevant data for the user for further analysis can be spread across these islands of data and not just on a single island of data. So to get the full benefits of these islands of data, one needs to link these pieces of information and make an integrated single piece of data which will then have complete answers to users' queries.

To illustrate this, let's take an example of how to link pieces of data. For instance, a clinical researcher is analysing an adverse event of drug, say Insulin, on patients having certain types of habits and diseases. The data about the drug and its adverse events on patients is stored in the form of RDF data-cubes across different sites. For instance, at site 1 the concept of drug is represented with the URI "<http:linked2safety.hcls.deri.org/drug>" and at site 2 the same drug concept is represented using the URI "<http:linked2safety.hcls.deri.org/dose>" which means that at site 1 insulin can be referred to as "<http:linked2safety.hcls.deri.org/drug#insulin>" while at site 2 it can be referred to as "<http:linked2safety.hcls.deri.org/dose#insulin>". Now to investigate an adverse event with the insulin as a drug, the clinical researcher needs to search the data-cubes to find relevant cubes for his/her case which has

the information about insulin and its adverse events. So if the clinical researcher queries the data space for insulin using the URI "<http:linked2safety.hcls.deri.org/drug#insulin>", he/she will miss site 2, which has information about the insulin drug and its adverse event. This missing information can be crucial to the conclusion regarding the hypothesis of the clinical researcher. To tackle this issue the semantic links between data in the data-cubes should be identified and established. More specific to our example the two concepts or instances should be linked as same concepts or instances using the OWL construct, "*owl:sameAs*". In the context of the above example the link to be identified and established would be as; "<http:linked2safety.hcls.deri.org/dose#insulin> *owl:sameAs* <http:linked2safety.hcls.deri.org/drug#insulin>". This example is illustrated graphically in Figure 10.

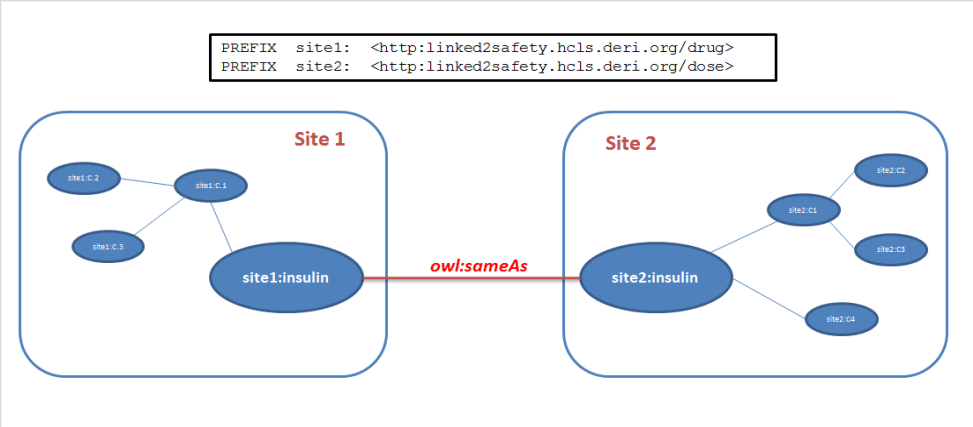

Figure 10: Linking – Example 1

Figure 11 below also illustrates an example of linking with the OWL construct, i.e., *owl:seeAlso*. In this example *owl:seeAlso* link is established among l2s:Dyslipidemia at site 1 and dbpedia-owl:dyslipidemia at DBpedia site in the Linked Open Data Cloud. Consequently, more information about the l2s:Dyslipidemia disease defined at site 1 can be found with the concept or instance dbpedia-owl:dylipidemia defined in a dataset named DBpedia in the Linked Open Data cloud. In this manner the LMDS can be linked with external data sources to provide further context and completeness to the data and hence in turn will make the data in LMDS richer in terms of the semantics of the data.

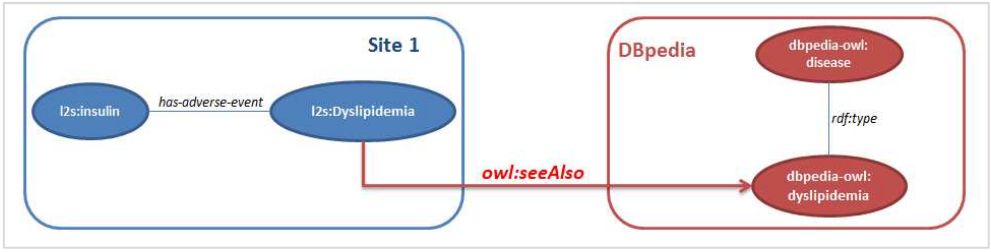

Figure 11: Linking – Example 2

After linking all the RDF data-cubes generated from different medical providers, new information, which can be in the form of implicit links between concepts as well as data, can be inferred (derived) from the existing links and relationships that exists between concepts and data. This inferred knowledge can add some crucial aspects to the existing knowledge which can be of great importance to the

clinical researchers. An example of such data inference within existing data source is depicted in Figure 12.

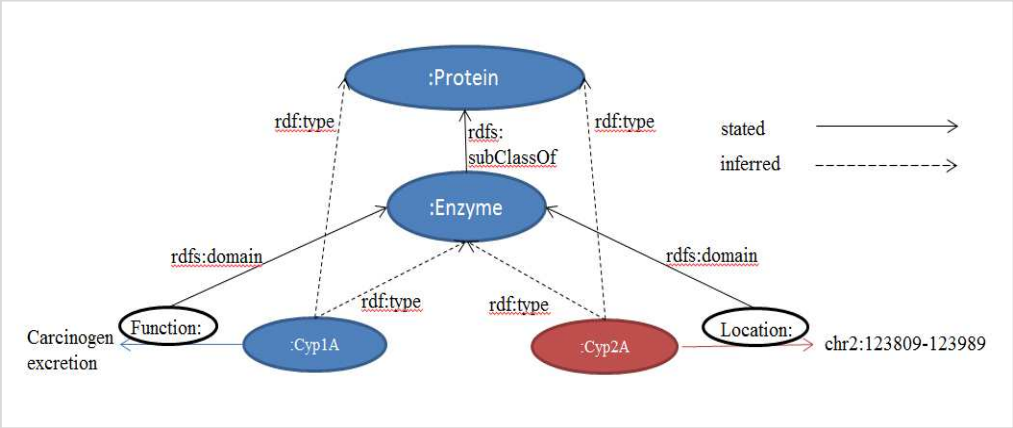

Figure 12: Inferencing – Example

The solid arrow lines in the figure represents existing explicit relationships which exists between the concepts, such as :Enzyme is a sub class of :Protein and Location property has :Enzyme as domain class and so on. The dotted lines with arrows represent the properties between concepts which are inferred on the basis of existing relationships which exists among the concepts. For example, :Cyp2A is the domain of a property Location which has also :Enzyme as a domain, so it can be inferred that :Cyp2A is a sub type (rdf:type) of :Enzyme. Also :Enzyme is a sub class of :Protein, so it can also be inferred that all concepts of type :Enzyme are also of type :Protein, as shown by the dotted lines in Figure 12. In the same manner, implicit knowledge that exists in the knowledge bases can be made explicit using the inference mechanism.

6. Federation Querying Component

Clinical researchers need a single point of access to query distributed data sources without having to construct queries for each data source.

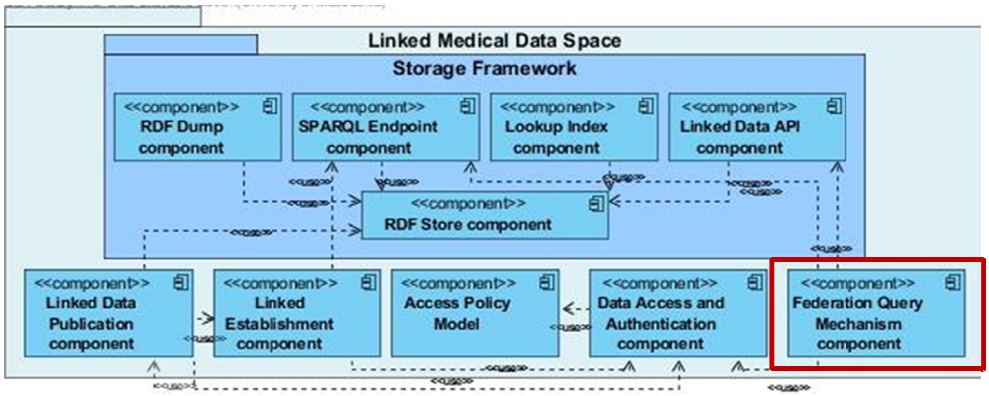

Figure 13: Linked Medical Data Space Architecture: Querying Framework

Figure 13 shows the dependencies of the Federation Query Mechanism component within the LMDS and Figure 14 shows the sequence of steps in federating a query and combining the results set.

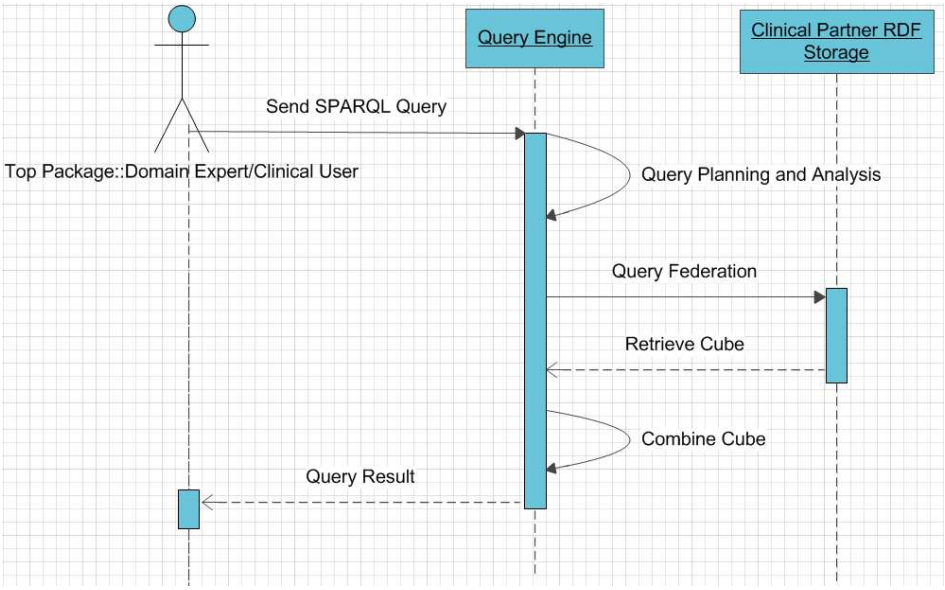

Figure 14: Sequence diagram for the Query Federation

To realise the federation of queries across distributed data sources, domain and technology experts have advocated the SPARQL query federation. A number of HCLS datasets are available in RDF format along with SPARQL endpoints to query the RDF data. A SPARQL query federation mechanism can be established to query and integrate data from all these distributed data sources to improve the research carried out by clinical researchers. SPARQL query federation is a mechanism to merge data from distributed RDF data sources.

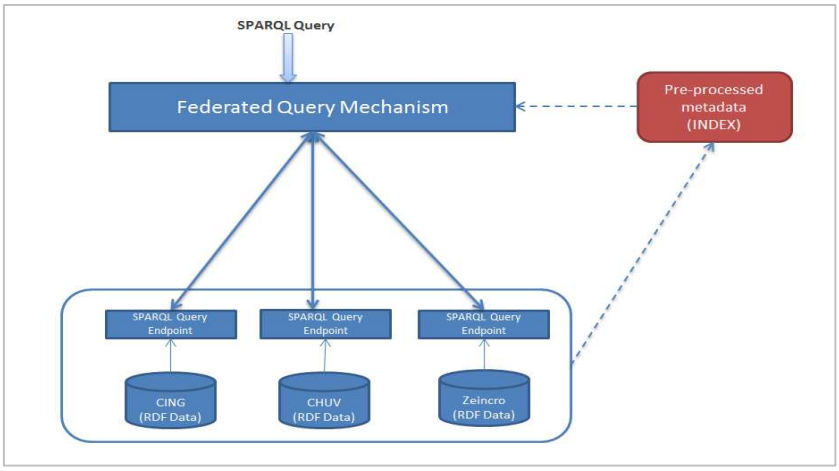

Figure 15: Architecture of Federated Querying Mechanism

Figure 15 represents the basic architecture of the LMDS federated querying mechanism. At each site, there is a standard SPARQL endpoint which can be used to query the underlying RDF data at that specific site. The data at each SPARQL end point is pre-processed to create index which will be used by the federated querying mechanism in query processing tasks, such as relevant source selection and site dependent sub query generation etc. This index can be updated with the passage of time, as data is updated at the SPARQL endpoints. The federated

querying mechanism, using the index, is responsible for sending the SPARQL query or sub query at the relevant site and then aggregating the results fetched from each site based on the query sent.

The Federation Query Mechanism in the LMDS is responsible for querying heterogeneous and distributed data sources (i.e., data-cubes). It provides services to allow querying heterogeneous and distributed data sources (i.e., data-cubes) as if they are in a central repository. In this way, users can seamlessly access complex and heterogeneous data sources. The primary responsibility of this component is the federation of queries and combining queries results from multiple data sources into a meaningful content. The knowledge (i.e. Semantic EHR Model) and data (i.e. data-cubes) stored in the LMDS only contains the data approved by the clinical partners under the legal and ethical context of the project. Therefore, the federated mechanism for queries adheres to the mentioned legal and ethical rules. Clinical queries can be executed using Linked Data APIs or directly using SPARQL queries. The Federation Query Mechanism interacts with the Linked Data API and/or with the SPARQL end point components.

For instance, a clinical researcher wants to query HCLS dataset which contains information about clinical observations of both male and female patients exhibiting adverse events, such as Dyslipidaemia, and BMI, and have been administered drugs, such as Insulin and Digrin. However, the data is distributed across two different sites rather than residing on a single centralized site. Figure 16 and Figure 17 show two distributed datasets stored at different sites. At the first site, i.e. Figure 16, only observations about male patients are stored while at second site, i.e. Figure 17, only observations about female patients are stored. Now if the clinical researcher wants to query the dataset to retrieve observations about both male and female patient subjects where patients have been administered Insulin as a drug and exhibit BMI > 25 and Dyslipidaemia as adverse events (i.e. both adverse events have value = 1), he/she will have to write separate queries for each site and then after getting the results from each site, he/she will have to aggregate the results to make sense of it or if he/she queries only one of the sites, then h/she will get incomplete results for the query. Query federation will provide a single point access to the distributed data stores and get aggregated results. The clinical researcher will write a single query to retrieve the desired results, the federated query mechanism will decompose the query to sub queries and run each sub query on the subsequent site and then aggregate all the retrieved results from different sites and present it to the clinical researcher. Getting such complete results with minimum effort will have a great impact on the research carried out in the clinical trial research domain.

```
@prefix qb: <http://purl.org/linked-data/cube#> .
@prefix sdmx-code: <http://purl.org/linked-data/sdmx/2009/code#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix sdmx-dimension: <http://purl.org/linked-data/sdmx/2009/dimension#> .
@prefix l2s-data: <http://linked2safety.hcls.deri.org/data/> .
@prefix l2s-dsd: <http://linked2safety.hcls.deri.org/dsd/> .
@prefix l2s-prop: <http://linked2safety.hcls.deri.org/property/> .
@prefix l2s-drug: <http://linked2safety.hcls.deri.org/codelist/drug/> .

l2s-data:clinical-trials a qb:DataSet ;
    qb:structure l2s-dsd:positives-by-adverse-event ;
    rdfs:label "Clinical Trials Dataset"@en ;
    rdfs:comment "Clinical Trials Dataset indicating adverse events seen in a sample population

<http://linked2safety.hcls.deri.org/data/clinical-trials/M/insulin/0/0/0/0> a qb:Observation ;
    qb:dataSet l2s-data:clinical-trials ;
    l2s-prop:drug l2s-drug:insulin ;
    sdmx-dimension:sex <sdmx-code:sex-M> ;
    l2s-prop:bmi25 "0"^^xsd:int ;
    l2s-prop:dyslipidemia "0"^^xsd:int ;
    l2s-prop:headache "0"^^xsd:int ;
    l2s-prop:rash "0"^^xsd:int ;
    l2s-prop:positives "72"^^xsd:int .

<http://linked2safety.hcls.deri.org/data/clinical-trials/M/insulin/0/0/0/1> a qb:Observation ;
    qb:dataSet l2s-data:clinical-trials ;
    l2s-prop:drug l2s-drug:insulin ;
    sdmx-dimension:sex <sdmx-code:sex-M> ;
    l2s-prop:bmi25 "0"^^xsd:int ;
    l2s-prop:dyslipidemia "0"^^xsd:int ;
    l2s-prop:headache "0"^^xsd:int ;
    l2s-prop:rash "1"^^xsd:int ;
    l2s-prop:positives "38"^^xsd:int .
```

Figure 16: Distributed Dataset 1

```
@prefix qb: <http://purl.org/linked-data/cube#> .
@prefix sdmx-code: <http://purl.org/linked-data/sdmx/2009/code#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix sdmx-dimension: <http://purl.org/linked-data/sdmx/2009/dimension#> .
@prefix l2s-data: <http://linked2safety.hcls.deri.org/data/> .
@prefix l2s-dsd: <http://linked2safety.hcls.deri.org/dsd/> .
@prefix l2s-prop: <http://linked2safety.hcls.deri.org/property/> .
@prefix l2s-drug: <http://linked2safety.hcls.deri.org/codelist/drug/> .

l2s-data:clinical-trials a qb:DataSet ;
    qb:structure l2s-dsd:positives-by-adverse-event ;
    rdfs:label "Clinical Trials Dataset"@en ;
    rdfs:comment "Clinical Trials Dataset indicating adverse events seen in a sample population

<http://linked2safety.hcls.deri.org/data/clinical-trials/F/insulin/0/0/0/0> a qb:Observation ;
    qb:dataSet l2s-data:clinical-trials ;
    l2s-prop:drug l2s-drug:insulin ;
    sdmx-dimension:sex <sdmx-code:sex-F> ;
    l2s-prop:bmi25 "0"^^xsd:int ;
    l2s-prop:dyslipidemia "0"^^xsd:int ;
    l2s-prop:headache "0"^^xsd:int ;
    l2s-prop:rash "0"^^xsd:int ;
    l2s-prop:positives "95"^^xsd:int .

<http://linked2safety.hcls.deri.org/data/clinical-trials/F/insulin/0/0/0/1> a qb:Observation ;
    qb:dataSet l2s-data:clinical-trials ;
    l2s-prop:drug l2s-drug:insulin ;
    sdmx-dimension:sex <sdmx-code:sex-F> ;
    l2s-prop:bmi25 "0"^^xsd:int ;
    l2s-prop:dyslipidemia "0"^^xsd:int ;
    l2s-prop:headache "0"^^xsd:int ;
    l2s-prop:rash "1"^^xsd:int ;
    l2s-prop:positives "29"^^xsd:int .
```

Figure 17: Distributed Dataset 2

Figure 18 represents a SPARQL query that will get the desired results for the clinical researcher. The query will be federated to each site according to the availability of data at each site and how much relevant the data at each site is to

the user query. The aggregated results are shown in Figure 19 where results contain both observations about male and female patients.

```
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX l2s-data:<http://linked2safety.org/data/>
PREFIX l2s-prop:<http://linked2safety.org/properties/>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX sdmx:<http://purl.org/linked-data/sdmx#>
PREFIX l2s:<http://linked2safety.org/>
PREFIX qb:<http://purl.org/linked-data/cube#>
PREFIX sdmx-code:<http://purl.org/linked-data/sdmx/2009/code#>
PREFIX sdmx-dimension:<http://purl.org/linked-data/sdmx/2009/dimension#>

SELECT ?instance ?positives ?gender FROM l2s-data:clinical-trials
WHERE {
?instance a qb:Observation .
?instance l2s-prop:drug ?drug .
?drug rdfs:label "Insulin" .
?instance l2s-prop:bmi25 "1"^^xsd:int .
?instance l2s-prop:dyslipidemia "1"^^xsd:int .
?instance l2s-prop:positives ?positives .
?instance sdmx-dimension:sex ?gender.
}
```

Figure 18: SPARQL Query 1

Another real time scenario is where a clinical researcher wants to know about the drugs which have dyslipidaemia and BMI > 25 as adverse events on patients. Figure 20 shows the SPARQL query which will answer the clinical researcher's query. In this scenario the data is distributed in such a fashion that observation about patients who have been administered Insulin as drug with its adverse events is stored at one site while observations about patients who have been administered Dirgin as drug with its adverse events is stored on another site. A snapshot of the data at both sites is shown in Figure 21.

Query Result

instance	positives	gender
http://linked2safety.org/data/clinical-trials/M/insulin/1/1/9//0	55	sdmx-code:sex-M
http://linked2safety.org/data/clinical-trials/F/insulin/1/1/9//9	69	sdmx-code:sex-F
http://linked2safety.org/data/clinical-trials/F/insulin/1/1/1//1	2	sdmx-code:sex-F
http://linked2safety.org/data/clinical-trials/F/insulin/1/1/9//1	7	sdmx-code:sex-F
http://linked2safety.org/data/clinical-trials/M/insulin/1/1/0//1	40	sdmx-code:sex-M
http://linked2safety.org/data/clinical-trials/F/insulin/1/1/0//1	1	sdmx-code:sex-F
http://linked2safety.org/data/clinical-trials/M/insulin/1/1/1//0	10	sdmx-code:sex-M
http://linked2safety.org/data/clinical-trials/M/insulin/1/1/9//1	83	sdmx-code:sex-M
http://linked2safety.org/data/clinical-trials/F/insulin/1/1/1//9	14	sdmx-code:sex-F
http://linked2safety.org/data/clinical-trials/M/insulin/1/1/0//9	69	sdmx-code:sex-M
http://linked2safety.org/data/clinical-trials/F/insulin/1/1/0//9	72	sdmx-code:sex-F
http://linked2safety.org/data/clinical-trials/M/insulin/1/1/0//0	64	sdmx-code:sex-M
http://linked2safety.org/data/clinical-trials/M/insulin/1/1/1//9	52	sdmx-code:sex-M
http://linked2safety.org/data/clinical-trials/M/insulin/1/1/9//9	7	sdmx-code:sex-M
http://linked2safety.org/data/clinical-trials/M/insulin/1/1/1//1	10	sdmx-code:sex-M
http://linked2safety.org/data/clinical-trials/F/insulin/1/1/0//0	14	sdmx-code:sex-F
http://linked2safety.org/data/clinical-trials/F/insulin/1/1/1//0	54	sdmx-code:sex-F
http://linked2safety.org/data/clinical-trials/F/insulin/1/1/9//0	99	sdmx-code:sex-F

Figure 19: SPARQL Query 1 Results

The SPARQL query in this scenario can be answered by both sites, so the query is sent as whole to the both sites by the query federation mechanism. Dataset 1 (site 1) provides data about Insulin and its adverse events, such as dyslipidaemia and BMI>25 exhibited by patients and Dataset 2 (site 2) provides data about Dirgin and the same adverse events exhibited by the patients. Both sites provide part of the answer to the clinical researcher's query. The query federation mechanism merges data fetched from both sites and presents it to the clinical researcher, giving the impression such that a single centralized data source is queried. The SPARQL query results are shown in Figure 22.

```
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX l2s-data:<http://linked2safety.org/data/>
PREFIX l2s-prop:<http://linked2safety.org/properties/>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX sdmx:<http://purl.org/linked-data/sdmx#>
PREFIX l2s:<http://linked2safety.org/>
PREFIX qb:<http://purl.org/linked-data/cube#>
PREFIX sdmx-code:<http://purl.org/linked-data/sdmx/2009/code#>
PREFIX sdmx-dimension:<http://purl.org/linked-data/sdmx/2009/dimension#>

SELECT ?drug ?dyslipidemia ?bmi25 ?positives FROM l2s-data:clinical-
trials
WHERE {
?instance a qb:Observation .
?instance l2s-prop:drug ?drug .
?instance l2s-prop:bmi25 ?bmi25 .
?instance l2s-prop:dyslipidemia ?dyslipidemia .
?instance l2s-prop:positives ?positives .
}
```

Figure 20: SPARQL Query 2

Dataset 1

Sex	Drug	BMI > 25	Dyslipidaemia	Positives
male	insulin	0	0	99
male	insulin	1	1	9
male	insulin	0	9	62

Dataset 2

Sex	Drug	BMI > 25	Dyslipidaemia	Positives
male	Digrin	0	0	14
male	Digrin	1	1	10
male	Digrin	0	9	11

Figure 21: Distributed Datasets

Query Result

drug	dyslipidemia	bmi25	positives
http://linked2safety.org/drug/digrin	0	1	51
http://linked2safety.org/drug/digrin	0	1	86
http://linked2safety.org/drug/digrin	9	0	34
http://linked2safety.org/drug/insulin	9	1	91
http://linked2safety.org/drug/digrin	0	1	42
http://linked2safety.org/drug/insulin	0	1	82
http://linked2safety.org/drug/insulin	9	9	25
http://linked2safety.org/drug/insulin	9	1	0
http://linked2safety.org/drug/insulin	1	1	55
http://linked2safety.org/drug/digrin	1	9	48
http://linked2safety.org/drug/insulin	0	1	90
http://linked2safety.org/drug/insulin	0	0	73

Figure 22: SPARQL Query 2 Results

It is important to mention that in the context of Linked2Safety project there will be a limited number of known and reliable SPARQL endpoints. The clinical partners controls the publication of data-cubes and dedicated resources will be allocated for the storage maintenance and availability of SPARQL endpoints.

7.Access Policy Model

The Access Policy Model of the Linked Medical Data Space named Linked Medical Data Space (LMDS) Security Model deals with the access control of the RDF data-cubes that are requested though federated queries from the distributed RDF sources located in the public rooms of the data providers. The policies that are applied on the RDF data-cubes allow or deny expert users access to them based on the data providers' preferences.

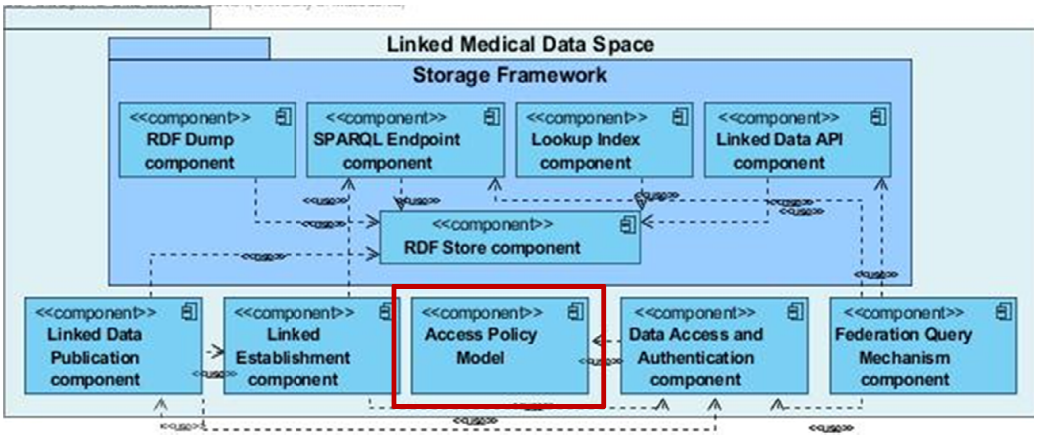

Figure 23: Linked Medical Data Space: Access Policy Model

Figure 23 shows the relation and dependencies of the Access Policy model with the other LMDS components. The LMDS Security Model acts as a common reference model allowing the consistent description and/or use of access policies within the Linked Medical Data Space. The semantic description of access policies will be further enhanced by a refinement of the role model associated with each access policy (this analysis will be reported in the following deliverable 4.2a,b).

Data providers use the LMDS Security Model to represent and assign access policies to the RDF data-cubes. Among other the access policies define the requirements that should be satisfied by the expert users in order to access the RDF data-cubes. As expert users are authenticated, access to data is permitted or prohibited to him/her based on their profile attributes/characteristics (e.g. role, purpose, working area, etc.).

The access policies are represented using a custom policy language (semantic rule language) and the system uses a reasoning engine in order to check the consistency of access policies and enforce them upon the requested data (i.e. RDF data-cubes) to create a temporary view on the stored data that contain only those elements that have been authorised to be retrieved by the specific expert user. Therefore, when evaluating a request query (SPARQL query), only the authorised RDF data-cubes are retrieved.

The design of an efficient Access Policy Model such as LMDS Security Model that will be applied to RDF data-cubes coming from distributed data providers should fulfil a number of requirements. In particular, it should consider and address the security requirements coming from the data providers (outlined in Deliverable D1.1 Requirements Analysis), while at the same time it should conform to the legislative, legal and ethical consideration (described in Deliverable D2.1 Legal and Ethical Requirements and Deliverable D2.2a Linked2Safety Data Privacy

Framework and Consent Forms). Furthermore, an efficient Access Policy Model should take into account the nature of data that will be controlled investigating similar approaches undertaken so far.

The rest of this section is structured as follows: Section 7.1 presents the requirements coming from the data providers and legislative considerations that will be taken into account for the design of the LMDS Security Model. Section 7.2.1 offers an overview of the significant literature published in research areas related to basic concepts covered by the LMDS Security Model. Section 7.2.2 outlines the LMDS Security Model and its basic concepts. Moreover, section 7.2.3 provides examples of possible access policies.

7.1. Requirements from Data Providers and Legislative Considerations

The goal of the LMDS Security Model is to provide confidentiality and security over anonymised, non-identifiable RDF data-cubes ensuring that only authorised expert users can access them. Data confidentiality is quite crucial in Linked2Safety as RDF data-cubes derive from privacy-sensitive/personal medical information and data providers may be not willing to share their content (i.e. RDF data-cubes' content) with every expert user even though cubes are provided in an anonymised, non-identifiable form. Thus, the LMDS Security Model should be able to adequately enforce data providers' preferences and protect their RDF data-cubes from unauthorised disclosure.

This section presents the requirements for the creation of the LMDS Security Model. The design of the LMDS Security Model should take into account some general security requirements that should be satisfied by any access policy model including:

- The system should be a closed system, where the lack of an authorisation rule implies no access.
- Access for the users of the system should be defined by their roles.
- Different types of roles have specific access constraints.
- Administrators of security. We need to have traditional security administrators that define roles, assign users to roles, create groups, and perform similar global functions.
- Attribute and credential-based authorisation[2]. In an environment where not all the users that may need access to the system and its data are known in advance, we need to have authorisation models that can consider user attributes and credentials to determine access rights.
- Explicit audit[3]. The model should make explicit those aspects that need to be logged for future audit. Audit is particularly important when we have context-dependent authorisation because of the possible legal implications of overriding or adding authorisations.

Moreover, the LMDS Security Model should consider more specific-oriented requirements focusing on this type of application. Such requirements are coming from the requirement analysis conducted in "D1.1 Requirement Analysis" and the analysis of legal and ethical issues performed in "D2.1 Legal and Ethical

[2] This requirement will be addressed by the authentication model (section 8)
[3] This requirement will be addressed by the audit model (section 8)

Requirements" and in the current version of "D2.2a Linked2Safety Data Privacy Framework and Consent Forms".

Requirements coming from data providers (D1.1 Requirement Analysis)

F11. Allow policy-based (authenticated) access to interlinked Linked2Safety data and clinical knowledge extracted by processing healthcare data

NF10. Support user authentication, authorisation and policy-based access control

NF11. Support Linked2Safety data security

Requirements coming from monitoring the legal and ethical issues in the project (D1.2 Legal and Ethical Requirements and the current version of D2.2a Linked2Safety Data Privacy Framework and Consent Forms)

1. Need for high security standards

"As regards Linked2Safety, since it is dealing with medical data which are considered to have sensitive character and at the same time number of people that may have access can be relatively high, high security standards will be required. "

2. Need for access control and auditing.

"In those two phases (development and testing) the processing of data can be therefore well controlled and audited within the consortium. External stakeholders will get access to relevant information in this phase only if they agree with the Linked2Safety confidentiality agreement that will monitor the data processing."

3. Access control based on expert user's purpose

"Particular conditions for the Expert user and his registering to the Linked2Safety platform through which he/she will be able to access the data will have to be developed. As far as data processing is concerned, even though the data will be already in anonymous form, Linked2Safety will still have to make sure that the processing is in line with the original patient's consent. The patient could for instance provide consent only for particular studies related to cardiovascular diseases. In such a case, Linked2Safety would need to make sure that the Expert user will use data only for studies which are in line with the consent. This might be done by concluding End user agreement through registration process or by accepting specific rules that regard specific data from the specific data provider."

4. Need for privacy preserving techniques

"Therefore, the guarantee of anonymity of data-cubes should not base only on the minimum threshold number, but other privacy preserving techniques related to data mining should be used. The better these technical (k-anonymity, access control, identification etc.) and legal (Non Disclosure Agreements (NDAs), binding rules) instruments are used the relatively lower the minimum threshold number can be."

5. The platform manager needs to fill the personal information of the user into a form when the user involves a natural person.

"If the platform manager creates a new user profile, he has to fill the personal information of the user (e.g. name, login id or openID/WebID, password, email address, skype id, phone number, user active status, etc.) into a form. Unless the user is a legal person the platform manager fills no personal data in the sense of the Data Protection Directive and or the proposal of a general data protection regulation into the form, because the data does not relate to an identifiable natural person. As soon as there is a natural person involved and the platform manager fills its data into the form, he is processing personal data."

An interesting point that emerges from the above analysis is that besides expert user's role, the LMDS Security Model should consider expert user's purpose in order to decide whether access to the Linked2Safety data will be authorised. Furthermore, data providers expressed the opinion that they might need to restrict access to their RDF data-cubes based on additional expert user's characteristics such as their origin, working area, company, etc.[4]

7.2. Background on Access Policy Models

The scope of this section is to survey the relevant literature in the domain of access policy models and present the background knowledge in the area. Moreover, it aims to identify the most important concepts that should be taken into account for the creation of an adaptable Access Policy Model based on the nature of data that will be controlled.

In Linked2Safety, the original healthcare data derived from clinical and medical data providers is transformed into aggregated, anonymised and non-identifiable data by the means of the data-cube approach. In particular, the data-cube generation process applies a set of anonymisation techniques to protect the patient data and ensure patient anonymity and non-identification (see in detail D1.2 Linked2Safety Reference Architecture p.33, p.88, and p.121-132). The produced data-cubes are inserted in the internal processes of Linked2Safety platform and are semantically described in RDF format by means of a common Data Cube Reference EHR Model (see in detail D1.4 Linked2Safety Semantic EHR Model). The semantically described data-cubes are then incorporated into a Linked Data cloud where access to this data is governed by adaptable access policies depending on the data owner's requirements and on the nature of this data.

We have identified three conceptual dimensions for the creation of an access policy model that will be applied on RDF data-cubes and we have accordingly divided the rest of this section as follows: subsection 7.2.1 presents the general idea of the access policy and the basic methods for controlling access to data. Section 7.2.2 outlines methods, techniques and models recently introduced to ensure controlled access to RDF data (linked data). Section 7.2.3 introduces the concept of OLAP and access models for restricting access to OLAP data. Finally, section 7.2.4 concludes the section summing up the main findings.

[4] This list of profile attributes/characteristics can be extended based on data providers' needs.

7.2.1. Basics in Access Policy Models

The objective of an access policy model (also known as access control model) is to protect data and resources against unauthorised disclosure (secrecy) and unauthorised or improper modifications (integrity), while at the same time ensuring their availability to legitimate users [4]. The design of an efficient access policy model and, especially, the definition of high-level rules and their formal representation constitute the first step for the creation and establishment of an access control system. In literature, there is a large number of access control models proposed so far to address the issue of restricting access to data [5]. Below are outlined some of the most commonly used approaches.

The **Discretionary Access Control (DAC)** [6] is a user-centric access control model that enforces access policies on the basis of the identity of the requesters while the data owner is the only responsible to assign permissions and define access rules. On the other hand, the **Mandatory Access Control (MAC)** [7] is an access control model that enforces access policies on the basis of the regulations mandated by a central authority. Moreover, each user is categorised into a security class and the resources are tagged with security labels that are used to restrict access to authorised users.

The **Role-Based Access Control (RBAC)** tries to close the gap between DAC and MAC since DAC is considered to be too flexible while MAC to be too rigid [5]. RBAC model uses permissions and rights that are assigned to roles to control access to resources. The roles are then assigned to users depending on their capabilities and the job requirements they hold. Access to a resource is determined based on the relationship between the requester and the organization or owner in control of the resource; in other words, the requester's role will determine whether access to the resource will be granted or denied.

The **Attribute Based Access Control model (ABAC)** [8] takes access control decisions based on a set of characteristics/attributes associated with the requester, the environment, and/or the resource itself. On the other hand, the **Purpose Based Access Control model (PBAC)** [9][10] is based on the notion of relating data objects with their purpose. A purpose can determine for what reason data is collected and what they can be used for. Moreover, the **Task Role Based Access Control model (T–RBAC)** [11] as the name implies, is based on the RBAC model while the user has a relationship with permission through role and task. Task is not only a "sub-role", it has a separate meaning from role. Finally, **Cryptographic Access Control model (CAC)** [12] is an attempt to design multilevel security models that are more general and capable of providing security in different contexts without requiring extensive changes to the fundamental architecture. Cryptographic keys for the various user groups requiring access to part of the shared data in the system are defined by classifying users into a number of disjoint security groups Ui, represented by a poset. By definition, in the poset, Ui<Uj implies that users in group Uj can have access to information destined for users in Ui but not the reverse.

The authorisation process is conducted by means of authorisation rules/policies which constitute the key outcomes of an access policy model. For example in [13], the authorisation rules are created to decide access in XML documents for a given access request. These are defined by a 5-tuple <subject, object, privilege, type, sign> where subject is the user (a user group, a role or credentials describing properties (e.g., age, gender, domain) of the user); object defines the element for which the access is requested; privilege is the type of access rights

that are assigned to a subject for the specific object (read, write); type specifies whether the authorisation on an element can be propagated to all of its direct and indirect sub elements (by cascade option), or to only its attributes; and sign can be positive (granting access), or negative (denying access). Similar references for the definition of authorisation rules can be found in [14] where a rule is defined by a 5-tuple <r (role), pt (privilege type), opr (operation), obj (object), at (authorisation type)> and in [15] where a rule is defined as (AR(s), t, o, x) indicating that it is permissible for a subject in role r to access object o in mode x using transaction t.

The main difference between DAC and MAC access policies is that the first are based on the identity of the requester and on the access rules defined by the owner of the resource while the second are based on mandated regulations set by a central authority. MAC approach seems to be more appropriate than DAC for the Linked2Safety system. The reason is that access policies in Linked2Safety must be common for all end-users and applied by a central authority. To implement this, each data provider needs to serve as a central authority specifying the access policies to the RDF data-cubes s/he possesses. Moreover, RBAC policies enable access based on the users' role within the system while the Attribute, the Purpose, the Task-Role and the Cryptographic Access Control model are different perspectives of the RBAC model that enable access based on user's attributes, purpose, task-role and cryptographic keys. The LMDS Security Model needs to adopt a combination of these models since access to the RDF data-cubes should be computed based on expert user's role, purpose, as well as other profile attributes/characteristics.

7.2.2. Access Policy Models for Linked Data

Current social web applications provide users with means to easily publish their information on the Web. However, once published, users cannot control how their data can be accessed apart from applying generic preferences (such as "friends" or "family"). In this section we present a set of ontologies/models recently proposed to address this issue and describe access control rules and preferences in the linked data.

The **Privacy Preference Ontology (PPO)**[5] [16][17] is a light-weight ABAC vocabulary that allows users to describe fine-grained privacy preferences for restricting or granting access to non-domain specific Linked Data elements. Specifically, a privacy preference contains properties that enable defining: (i) the resource, statement, named graph, dataset or context it must grant or restrict access to; (ii) the conditions refining what to grant or restrict; (iii) the access control type; and (iv) a SPARQL query, (known as AccessSpace) i.e. a graph pattern representing what must be satisfied by the user requesting information. The PPO can be also applied to any social data as long as it is modeled in RDF.

[5] http://vocab.deri.ie/ppo#

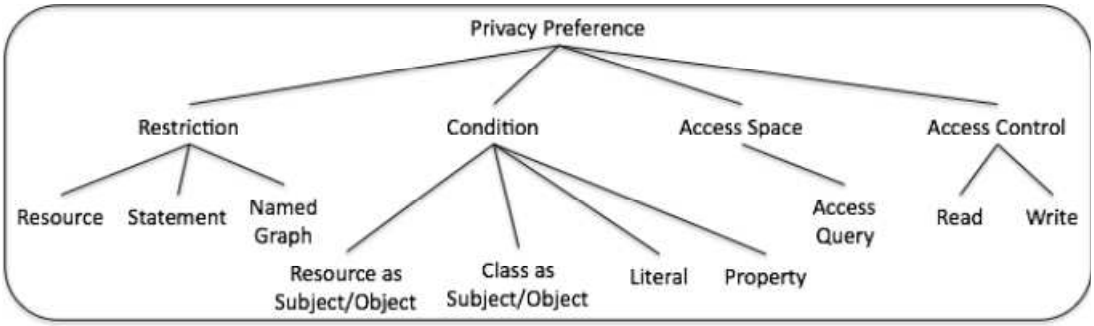

Figure 24: The PPO Ontology

The **Social Semantic SPARQL Security for Access Control vocabulary (S4AC)** is a lightweight vocabulary which allows the information providers to specify fine-grained access control policies for their RDF data [18][19]. S4AC re-uses concepts from SIOC[6], SCOT[7], NiceTag[8], WAC[9],TIME[10], GEO[11], and the access control model as a whole is grounded on further existing ontologies, as FOAF[12], Dublin Core[13], and RELATIONSHIPS[14]. At the core of S4AC model is the Access Condition which is a SPARQL 1.1[15] ASK clause that specifies the condition to be satisfied in order to grant the access to a resource. Moreover, the users can define Access Conditions based on tags which restrain the conditions to the resources tagged with such tags, e.g., resources tagged "friends", "amici", "ami". Finally, the Access Condition is associated with a temporal and spatial validity.

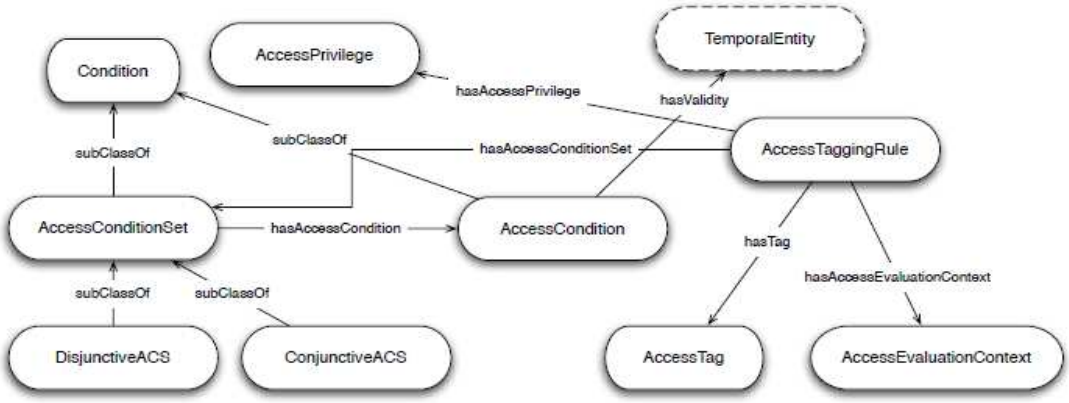

Figure 25: The S4AC Vocabulary

In order to facilitate the easy description of access policies, the authors of [20] have developed short notation for policies and rules named as **Prologstyle SWRL Format (PsSF)**. Each rule consists of i) a label, ii) a rule antecedent

[6] http://rdfs.org/sioc/spec/
[7] http://scot-project.net/
[8] http://ns.inria.fr/nicetag/2010/09/09/voc.html
[9] http://www.w3.org/wiki/WebAccessControl
[10] http://www.w3.org/TR/2006/WD-owl-time-20060927/
[11] http://www.w3.org/2003/01/geo/wgs84 pos
[12] http://xmlns.com/foaf/spec/
[13] http://dublincore.org/documents/dcmi-terms/
[14] http://vocab.org/relationship/.html
[15] http://www.w3.org/TR/sparql11-query/

describing the condition under which the rule is satisfied and iii) a consequent. Both the antecedent and the consequent contain a collection of predicates joined by the logical AND condition. Each access rule can be defined according to different types of access including query and update. Conditions are expressed in a positive fashion, while negation is not supported.

The **Semantic Based Access Control model (SBAC)** is an access control model for protecting Semantic Web resources which authenticates users based on the credentials they offer when requesting an access right [21]. To succeed this, SBAC consists of three basic components: Ontology Base, Authorisation Base and Operations. Ontology Base is a set of ontologies used for modeling entities along with their semantic interrelations in three domains of access control, namely subject domain, object domain and action domain. Authorisation Base is a set of authorisation rules in form of (s, o,±a) in which s is an entity in SO, o is an entity defined in OO, and a is an action defined in AO. Last, the operations are executed on Authorisation Base and are for making decision about a request, granting an access right or revoking an access right.

The **Access Management Ontology (AMO)** [22] is an access model dedicated to the representation of the access rights given on resources shared by a network of users. AMO is made of a set of classes and properties for annotating the resources and a base of inference rules modelling the access control policy. The classes of Role, Action and AccessType are central to AMO while the class of foaf:Agent is the one used for the representation of the requester.

The **Relation based Access Control (RelBAC)** [23] is a new model for access control designed for open, highly dynamic environments. The key idea, which differentiates the RelBAC model from the state of the art, is that permissions are modelled as relations between users (called subjects) and data (called objects) while access control rules are their instantiations on specific sets of users and objects.

In [24], the proposed solution supports access control via an RDF metadata file that contains an access control list (ACL) [16] metadata. Authorisation is determined by running a set of SPARQL queries to determine if the user is granted access either as an agent or as a member of an agent class. In particular, for a given rule, acl:accessTo defines a resource that access is being granted to, acl:agent and acl:agentClass define an agent or agent class (such as "any foaf:Person") as being granted access, acl:mode defines the set of modes that are granted to the agent or agent class, and acl:defaultForNew optionally defines the default access rules for new documents in a directory.

After the analysis of the aforementioned access policy models, we identified that they share a set of common features. More specifically, the PPO specifies the requester's characteristics that should be met in order to access the linked data. Similarly, the S4AC and the PsSF use a condition concept that should be satisfied in order users can grant the access to a resource. The AMO contains a class used for the representation of the requester. The LMDS Security Model should take into consideration these findings and enforce policies (preferences) on linked RDF data-cubes based on requester' conditions. These conditions should be satisfied in order access to the linked data will be permitted.

[16] http://www.w3.org/ns/auth/acl#

7.2.3. Access Policy Models for OLAP Data

Online Analytic Processing (OLAP) is one of the most popular decision support techniques, which enables the exploration of large amounts of data in Data Warehouses (DW). Even though most OLAP are implemented on top of relational databases, security measures and access policy models specified for relational databases are not appropriate for OLAP. The main reason is that these models specify protection in terms of database tables, rows and columns. Instead, protection for OLAP must be defined on a Multidimensional (MD) basis, since OLAP users query the OLAP in terms of facts, dimensions, and so on. Thus, a security model for OLAP must be defined on a MD basis, supporting all features of the data-cube concept.

At present, few approaches have been proposed to restrict access to OLAP and data warehouses.

The **Access Control and Audit (ACA) model** [25] has been proposed to regulate access to objects in a MD model. To succeed this, it considers a combination of MAC and RBAC model which is based on the classification of subjects and objects in the system. In particular, an access class is defined on the basis of three different but compatible ways of classifying users: i) by their security level, ii) the role they play and iii) by the compartments they belong to. This allows specifying Sensitivity Information Assignment Rules (SIAR) over multidimensional elements (facts, dimensions, etc.). In addition, the model allows defining Authorisation Rules (AUR) that represent exceptions to the general multilevel rules, where the designer can specify different situations in which the multilevel rules are not sufficient. Finally, a set of Audit Rules (AR), which represent the corresponding audit requirements, can be included in the model.

A security model for OLAP is proposed in [26] that is based on the assumption of a central security policy. The access restrictions are defined as authorisation constraints making the identification of security objects and subjects necessary. In addition to the main elements (cubes, dimensions, etc.) of an OLAP the element role is introduced. Authorisation constraints can either be positive (explicit grants) or negative (explicit denials) while they limit the security model to read access. For specifying a security constrain, the authors introduce a Multidimensional Security Constraint Language (MDSCL) that is based on MDX. For example, <hide cube statement>= HIDE CUBE <cube name> FOR ROLE <role name>.

The authors of [27] specify a simplified authorisation model for users of an OLAP based on the following set: <S, F, D, T> where S sets of subjects, F of facts, D of dimensions and T of allowed operations. In addition to the previous specification they apply the rule of a closed world assumption so that operations are not permissible unless explicitly granted.

A rule-based user role profile to manage the security and access issues in OLAP is presented in [28]. The security model has only roles as security subjects. That means authorisations can only be granted to roles. The authorisations are presented as rules to the OLAP (using DAC model). In general, a security rule is a 4-ruple <s,a,o,p> where s is subject, a is the access type a to access security object o within the range of predicate p.

The access policy models proposed for restricting access to OLAP data seems to share similar concepts with the models reviewed. For example, the ACE model

creates policies that restrict access to OLAP data based on subject's characteristics including their security level, the role they play and the compartments they belong to. Moreover, the security model for OLAP proposed in [26] defines policies based on user's role. The only difference of OLAP authorisation rules is that the authorisation object should be defined in a multidimensional basis taking into account dimensions, measures and hierarchies of the data-cubes.

7.2.4. Summary

The analysis of existing literature revealed a number of aspects that can be taken into account for the design of an efficient access policy model in Linked2Safety system.

First, we analysed the basic methods for controlling access to data. The LMDS Security Model shares fundamental concepts with MAC as well as a combination of other access policy models including RBAC, PBAC and ABAC. Similar to MAC, LMDS Security Model should enable specific personnel of each corresponding data provider to serve as a central authority. This central authority will establish rules on his/her distributed system for the RDF data-cubes s/he owns and define which remote expert users can access which portion of these RDF data-cubes. Furthermore, LMDS Security Model should restrict access based on the expert user's role, purpose as well as on other expert user's attributes such as origin, working area, company, etc.

Afterwards, the access policy models proposed for specifying access policies (rules) in the linked data environment have been surveyed. Similarly, the LMDS Security Model is going to restrict access to RDF data-cubes retrieved from federated queries. Thus, the LMDS Security Model should adhere to linked data principles and policies. A first step to succeed this is to capture a condition concept that will match requester's criteria. Based on this, access to the requested linked RDF data-cubes will be permitted only if a number of conditions are satisfied.

Finally, the authorisation methods to access OLAP data have been investigated. This type of authorisation differs from conventional ones since an access policy model for restricting access to OLAP data should take into consideration concepts and principles drawn from the multidimensional design. Therefore, the LMDS Security Model which is a model for specifying access policies to RDF data-cubes should take into account the multidimensional character of the protected data introducing appropriate concepts for its description.

7.3. Design of the Access Policy Model

There's no one "correct" way or methodology for developing ontologies (i.e. identify the concepts and properties); the best solution almost always depends on the application that will use it and the context needed to be described. So deciding what we're using the ontology for will guide many of the modelling decisions down the road.

We have also to remember that the ontology development is an iterative process: after defining an initial version of the model, we would evaluate the overall framework and debug it trying to apply the model to the Linked2Safety use cases; as a result, these actions will potentially lead to partly revising the model.

This process of iterative design will likely continue through the entire lifecycle of the model (Deliverable 4.2a,b Linked Medical Data Space).

The rest of this section is structured as follows: Section 7.3.1 provides basic information about the model including the modelling language, the imported vocabularies, the introduction of concepts influenced by the requirements and the literature analysis and the model's prefix. Section 7.3.2 presents the structure of the model and the specification of its main entities. Finally, section 7.3.3 provides examples of possible access policies that can be enforced on RDF data-cubes.

7.3.1. LMDS Security Model

The creation of a stable and consistent access policy model will be facilitated by the adoption of a semantic language and re-using, when possible, already existing proposals, to avoid re-inventing the wheel.

There are several languages that can be used to implement an ontology/semantic model. Very generic and flexible ones (such as OWL) enable the expression of complex relationships between concepts and roles in the domain; ontologies using such formalisms are called heavyweight ontologies in contrast to lighweight ontologies which use simpler formalisms (as RDF or RDF schema) with fewer possibilities to express complex relationship. Choosing the proper formalism is strictly linked to the computational operations to be performed on the model.

The LMDS Security Model adopts a meta-level approach to specify access policies on top of the RDF data-cubes using OWL language while at the same time, it re-uses existing vocabularies. More specifically, it re-uses concepts and properties derived from the RDF Data Cube vocabulary[17] to describe the RDF data-cubes while it incorporates the Web Access Control (WAC)[18], a vocabulary that is used to define aspects related to the access privilege types. Moreover, it re-uses the FOAF vocabulary[19] for the description of users' characteristics. Figure 26: Importing structure of the LMDS Security Model shows the importing structure of upper-level ontologies to the LMDS Security Model.

Furthermore, the LMDS Security Model adopts a similar conceptual analysis with that proposed by the PPO ontology[20] in order to define access control policies for the RDF data-cubes. As described above, the PPO ontology is a lightweight ontology which specifies access preferences on top of linked data. Last, the LMDS Security Model introduces a set of new concepts and properties in order to address the data providers' requirements (e.g. the "role" concept as well as several user profile-related concepts including the "origin", the "working area", etc), the ethical and legal considerations (e.g. the "purpose" concept[21]), and requirements oriented to the type of data that will be controlled (i.e. concepts related to the linked and the multidimensional nature of data).

[17] http://publishing-statistical-data.googlecode.com/svn/trunk/specs/src/main/html/cube.html#ref_qb_DataSet

[18] http://www.w3.org/wiki/WebAccessControl

[19] http://xmlns.com/foaf/spec/

[20] http://vocab.deri.ie/ppo#ConditionOperator

[21] The purpose concept will be inserted to the user' profile indicating the purpose for retrieving the RDF data cubes. In case the purpose is not in line with that specified by the access policy, the requested RDF data cubes will not be authorised (and will not be retrieved).

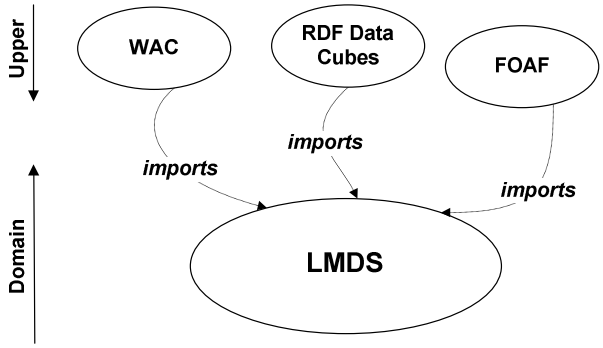

Figure 26: Importing structure of the LMDS Security Model

The prefix-URI used within LMDS Security Model is depicted in Table 2.

Table 2: Namespaces used for the LMDS Security Model

Prefix	URI	Description
lmds	http://www.linked2safety.eu/lmds#	Linked medical data space

7.3.2. LMDS Security Model Structure

In this section, we present the structure of the LMDS Security Model (first version) and the specifications of its main entities (concepts and properties), together with the main relations between them. An overview of the LMDS Security Model is illustrated in Figure 27.

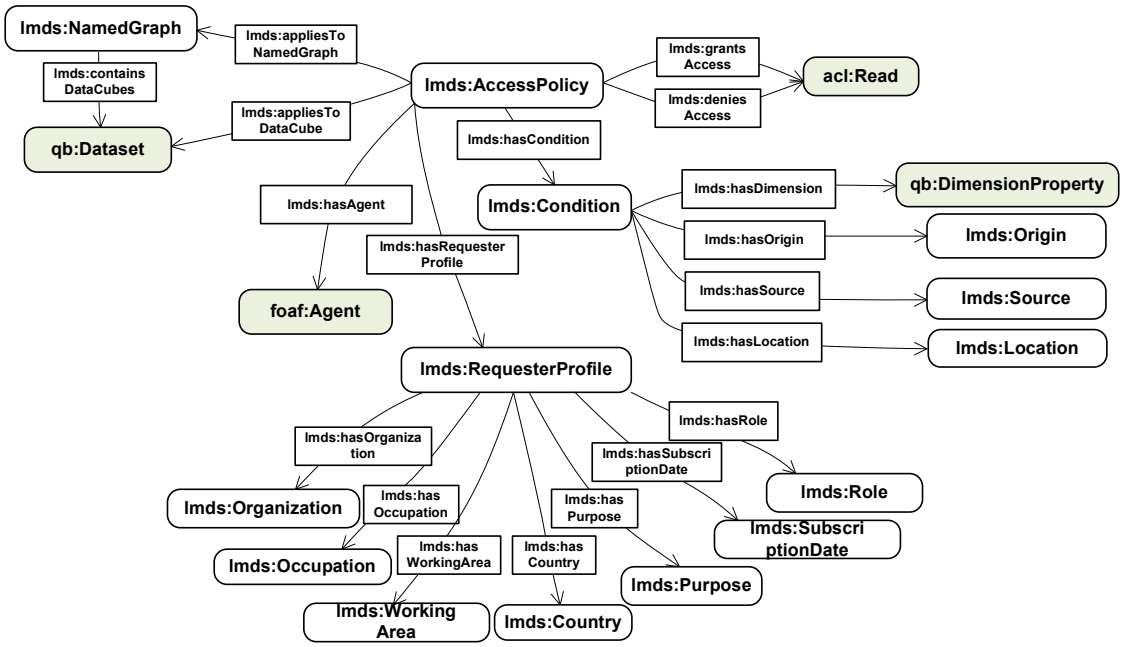

Figure 27: LMDS Security Model

The LMDS Security Model provides a main class called **AccessPolicy**. An access policy contains properties that enable restricting access to RDF data-cubes by defining:

- the restriction domain or else the data-cubes that are going to be protected by the specific access policy (by granting or denying access to);
- refined conditions about which data-cubes are going to be protected based on data-cubes' characteristics;
- the access control privilege type; and
- the requester's criteria that describe the requirements (attributes) an expert user must satisfy in order to have access to the requesting information.

Before proceeding to the description of the model, we should mention that restrictions/access policies can be applied in three different ways:

- **Directly**, by defining the restriction domain and thus granting/denying access to specific data-cubes (denoted by their URIs);
- **Indirectly**, by defining conditions. In this case, access policies will be applied to every data-cube satisfies these conditions.
- In a **Hybrid** way, by specifying both the restriction domain (i.e. the URIs of the data-cubes needed to be protected) and a set of conditions. In this case, from the initial list of data-cubes defined by the restriction domain, only the cubes that satisfy the conditions will be associated with the specific access policy.

lmds:AccessPolicy	
URI	http://www.linked2safety.eu/lmds#AccessPolicy
Definition	An Access Policy contains statements about restricting access to RDF data-cubes
SubClassOf	Owl:Thing
In-Domain-Of	lmds:appliesToNameGraph, lmds:appliesToDataCube, lmds:hasCondition, lmds:grantsAccess, lmds:deniesAccess, lmds:hasRequesterProfile, lmds:hasAgent (It can also contain combinations of Conditions) lmds:hasConditionOperator
In-Range-Of	

7.3.2.1. Restriction Domain

An access policy can be applied to hide or allow access to a restriction domain that includes a data-cube or a group of data-cubes (stored in a "named graph"). From a microscopic level, an access policy can be applied to hide observations or slices of a data-cube (e.g. by defining the dimension(s) that should be hided). However, these concepts are not included in our model since queries in Linked

Medical Data Space are performed over distributed RDF data-cubes and data providers need to assign access policies to restrict/allow access to a complete data-cube rather than dimensions or measures within it.

The LMDS Security Model introduces a number of classes and properties in order to define the restriction domain depicted in Figure 28.

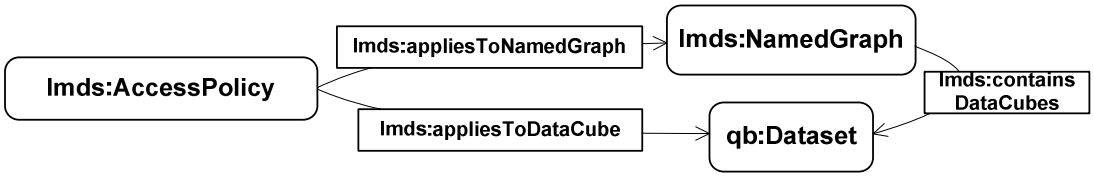

Figure 28: Restriction Domain

The restriction domain contains either a RDF data-cube or a group of RDF data-cubes, stored as RDF named graphs, which are needed to be restricted using similar access policies.

For the description of the data-cube, the LMDS Security Model imports the RDF Data Cube vocabulary[22] and, more specifically, the qb:Dataset class which is used to represent a data-cube.

On the other hand, the LMDS Security Model uses the named graphs to combine data-cube statements and apply an access policy to the graph, using the lmds:appliesToNamedGraph property. A named graph consists of (1) a name denoted by a URI and (2) a set of data-cubes statements (an RDF graph) mapped to this name.

qb:Dataset	
URI	http://purl.org/linked-data/cube#DataSet
Definition	Represents a collection of observations, possibly organised into various slices, conforming to some common dimensional structure.
SubClassOf	Qb:Attachable
In-Domain-Of	qb:slice, qb:sliceKey, qb:structure
In-Range-Of	lmds:appliesToDataCube, lmds:containsDataCubes

lmds:NamedGraph	
URI	http://www.linked2safety.eu/lmds#NamedGraph
Definition	A Named Graph contains a group of data-cube statements

[22] http://publishing-statistical-data.googlecode.com/svn/trunk/specs/src/main/html/cube.html

SubClassOf	Owl:Thing
In-Domain-Of	lmds:containsDataCubes
In-Range-Of	lmds:appliesToNamedGraph

The access policies can be applied to the restriction domain using the following set of properties:

lmds:appliesToDataCube	
URI	http://www.linked2safety.eu/lmds#appliesToDataCube
Definition	An access policy that applies to an qb:Dataset. When an access policy has this property it means that the access policy applies to a data-cube.
Domain	lmds:AccessPolicy
Range	qb:Dataset

lmds:appliesToNamedGraph	
URI	http://www.linked2safety.eu/lmds#appliesToNamedGraph
Definition	An access policy that applies to an lmds:NamedGraph. When an access policy has this property it means that the access policy applies to a group of data-cubes.
Domain	llmds:AccessPolicy
Range	lmds:NamedGraph

lmds:containsDataCubes	
URI	http://www.linked2safety.eu/lmds#containsDataCubes
Definition	A Named Graph contains a group of data-cubes
Domain	lmds:NamedGraph
Range	qb:Dataset

7.3.2.2. Condition

A condition defines fine-grained restrictions within an access policy. More specifically, the condition class defines which data-cubes are going to be granted or restricted access based on their properties (i.e. meta-information). Hence, data providers can use this class to define conditions about the data-cube's component properties (i.e. dimensions, measures, attributes, measure dimensions) as well as data-cube's provider/origin, source or location[23]. For example, a data provider can deny access to every data-cube having as dimension the property MBI or allow access to every data-cube coming from a specific source e.g. a respiratory clinical trial or from a specific location/country.

Figure 29 presents the concepts and properties used for the definition of a condition.

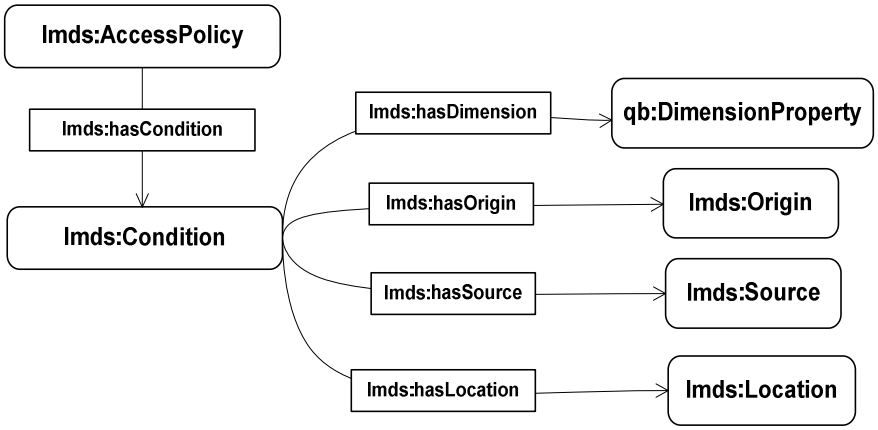

Figure 29: Condition

lmds:Condition	
URI	http://www.linked2safety.eu/lmds#Condition
Definition	A condition contains properties that will be tested to specify the data-cubes that will be protected with the access policy.
SubClassOf	Owl:Thing
In-Domain-Of	lmds:hasDimension, lmds:hasOrigin, lmds:hasSource, lmds:hasLocation
In-Range-Of	lmds:hasCondition, lmds:ConditionOperatorOf

[23] These properties will be expanded to match the metadata information that is kept for each data cube

qb:DimensionProperty	
URI	http://purl.org/linked-data/cube#DimensionProperty
Definition	The class of components which represent the dimensions of the cube
SubClassOf	qb:ComponentProperty, qb:CodedProperty
In-Domain-Of	
In-Range-Of	lmds:hasDimension

lmds:Origin	
URI	http://www.linked2safety.eu/lmds#Origin
Definition	The Origin defines the origin/provider supplied a set of data-cubes
SubClassOf	Owl:Thing
In-Domain-Of	
In-Range-Of	lmds:hasOrigin

An origin is identified by a URI that denotes the origin of the data-cubes. This is mainly used in case of a CRO. By this property, a CRO user can grand or deny access to data-cubes coming from a specific origin/provider. For instance, a data provider needs to deny access to all data-cubes coming from Pharma A (instance of origin).

lmds:Source	
URI	http://www.linked2safety.eu/lmds#Source
Definition	The Source defines the source of a set of data-cubes
SubClassOf	Owl:Thing
In-Domain-Of	
In-Range-Of	lmds:hasSource

A source is identified by a URI that denotes the source of the data-cubes. A data provider can generate data-cubes from several medical sources. Sources are the clinical trials from which a set of data-cubes have been generated. For instance, a data provider needs to deny access to all data-cubes coming from a respiratory clinical trial (instance of source class).

lmds:Location	
URI	http://www.linked2safety.eu/lmds#Location
Definition	The Location defines the location of a set of data-cubes
SubClassOf	Owl:Thing
In-Domain-Of	
In-Range-Of	lmds:hasLocation

A location is identified by a URI that denotes the location of the data-cubes. A clinical trial can be performed on several locations. Data providers can use location class to apply privacy policies to all data-cubes coming from a specific location. For instance, a data provider needs to deny access to all data-cubes coming from Greece (instance of location class). [24]

lmds:hasCondition	
URI	http://www.linked2safety.eu/lmds#hasCondition
Definition	The conditions which an access policy has
Domain	lmds:AccessPolicy
Range	lmds:Condition

lmds:hasDimension	
URI	http://www.linked2safety.eu/lmds#hasDimension
Definition	A condition that defines the qb:DimensionProperty to be a specific attribute.
Domain	lmds:Condition
Range	qb:DimensionProperty

lmds:hasOrigin	
URI	http://www.linked2safety.eu/lmds#hasOrigin

[24] To succeed this, the properties Origin, Source, and Location should be defined in the metadata file that is created along with the data-cube.

Definition	A condition that defines the lmds:Origin to be a specific attribute
Domain	lmds:Condition
Range	lmds:Origin

lmds:hasSource	
URI	http://www.linked2safety.eu/lmds#hasSource
Definition	A condition that defines the lmds:Source to be a specific attribute.
Domain	lmds:Condition
Range	lmds:Source

lmds:hasLocation	
URI	http://www.linked2safety.eu/lmds#hasLocation
Definition	A condition that defines the lmds:Location to be a specific attribute.
Domain	lmds:Condition
Range	lmds:Location

A data provider can define more than one conditions that will be applied simultaneously (connected by "or", "and" "and not" operators). This is achieved by means of Condition Operator concept (Figure 30).

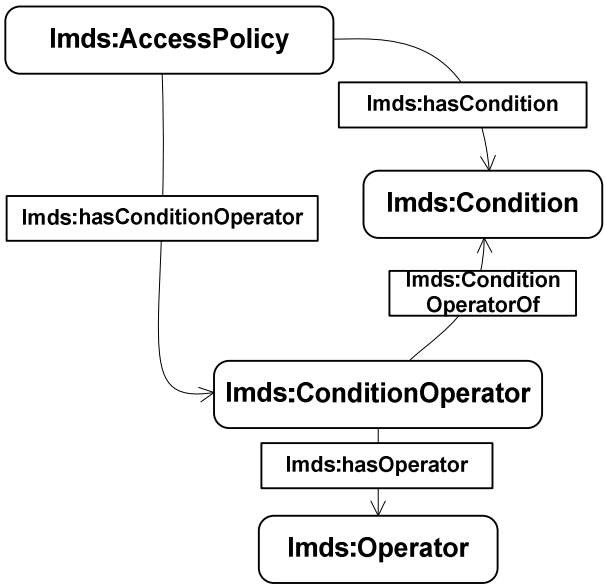

Figure 30: Condition Operator

A data provider can use the class lmds:ConditionOperator to combine more than one Conditions (connected with the property lmds:ConditionOperatorOf). This is achieved by means of class lmds:Operator which provides logical operators consisting of conjunction, disjunction and negation. Thus, lmds:And, lmds:Or and lmds:Not are subclasses of lmds:Operator (Figure 31).

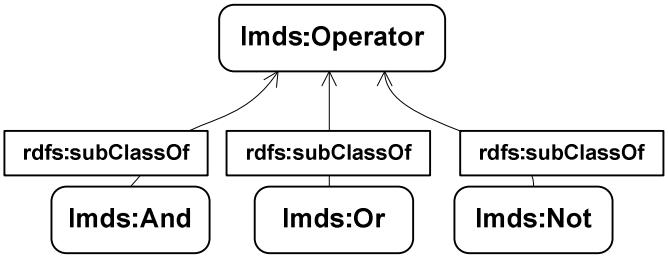

Figure 31: Operators

For the respective properties, a short description is provided below:

lmds:hasConditionOperator: this property defines that an Access Policy can have a Condition Operator to include more than one Conditions

lmds:ConditionOperatorOf: the Conditions that are used for the creation of a Condition Operator

lmds:hasOperator: a Condition Operator contains an Operator

7.3.2.3. Access Control Privilege

The LMDS Security Model provides two properties (i.e. lmds:grantsAccess and lmds:deniesAccess) that describe the type of access control privilege that is granted or restricted to the user when an access policy applies. The access control privilege that is enforced to the data-cubes is the read privilege since the

read operation is the most common operation for the final users of multidimensional environments [26][29].

For the definition of read privilege, the LMDS Security Model imports the WAC vocabulary and, more specifically, the acl:Read class which is used to represent the read operation.

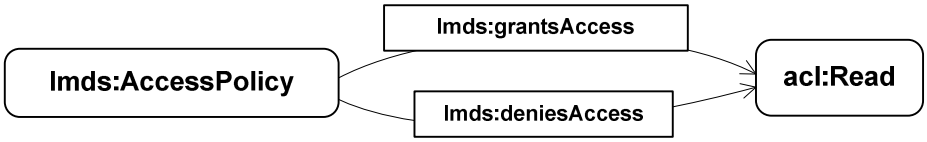

Figure 32: Access Control Privilege

Acl:Read	
URI	http://www.w3.org/ns/auth/acl#Read
Definition	The class of read operations
SubClassOf	acl:Access
In-Domain-Of	
In-Range-Of	lmds:grantsAccess, lmds:deniesAccess

lmds:grantsAccess	
URI	http://www.linked2safety.eu/lmds#grantsAccess
Definition	The access control privilege which is granted to the user. The access control is described using the Web Access Control vocabulary.
Domain	lmds:AccessPolicy
Range	acl:Access

lmds:deniesAccess	
URI	http://www.linked2safety.eu/lmds#deniesAccess
Definition	The access control privilege which is not granted to the user. The access control is described using the Web Access Control vocabulary.
Domain	lmds:AccessPolicy

Range	acl:Access

The WAC vocabulary, from which the class acl:Read is imported, is depicted in Figure 33.

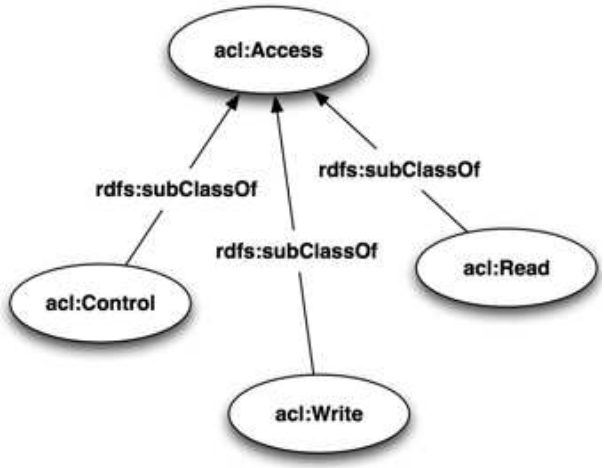

Figure 33: WAC Vocabulary

As mentioned earlier, the system should be a closed system, where the lack of an authorisation rule implies no access. However, we included both access types (allow/deny) since there are several cases even in a closed system where we need to apply "deny" access type (for example a data provider may need to rank access for a user and the first "Deny" that is reached overrides everything else).

7.3.2.4. Requester's Criteria

Finally, an access policy is associated with a specific user (using the class foaf:Agent) or with a requester profile. In case the data provider defines a requester profile that can have access to a set of data-cubes, access to that cubes is restricted according to a pattern/profile which users must satisfy, for instance having a particular working area. This will enable our system to keep track of the persons gaining access to the information and prevent unauthorised disclosures. Figure 34 presents the requester' profile class.

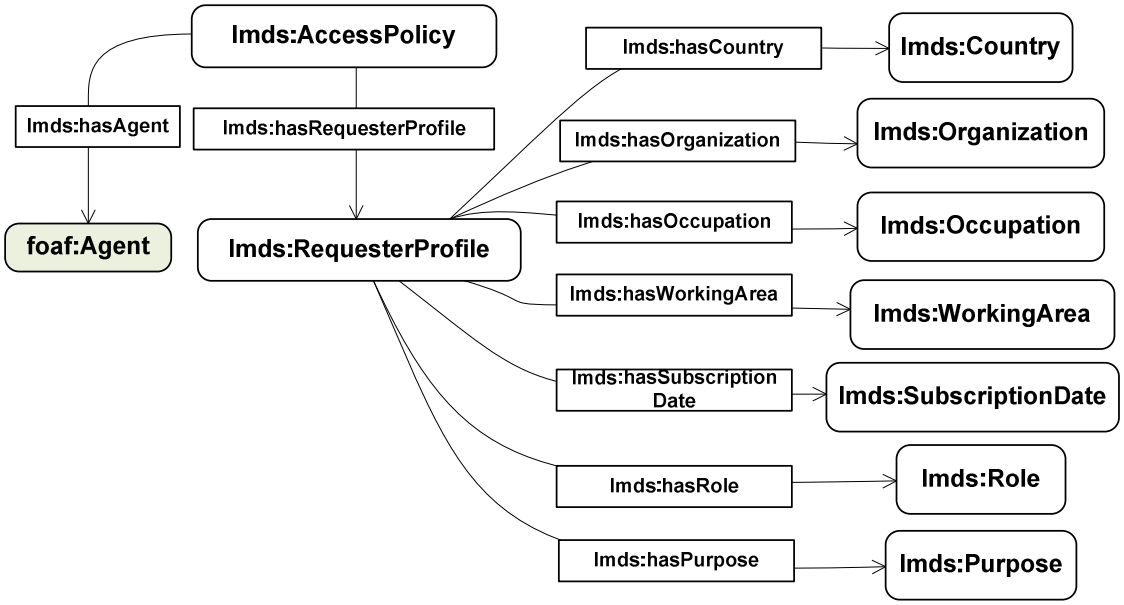

Figure 34: Requester' Criteria

The central class here is the lmds:RequesterProfile which is used to define certain requirements which a requester must meet.

lmds:RequesterProfile	
URI	http://www.linked2safety.eu/lmds#RequesterProfile
Definition	A Requester's Profile defines the requirements which a requester must satisfies in order to be granted/denied access to the data-cubes
SubClassOf	
In-Domain-Of	lmds:hasCountry, lmds:hasOrganization, lmds:hasOccupation, lmds:hasWorkingArea, lmds:hasSubscribtionDate, lmds: hasPurpose, lmds:hasRole
In-Range-Of	lmds:hasRequesterProfile

Although it is recommended to use lmds:RequesterProfile to determine who can be granted (or denied) the access control privileges, there are instances when users would want to grant (or deny) access to a specific agent without the need to test whether the agent satisfies specific attributes. Therefore, the data provider can define a specific Agent giving a url.

foaf:Agent	
URI	http://xmlns.com/foaf/spec/#term_Agent
Definition	An Agent can be person, group, software or physical artifact.

	The Agent class is the class of agents; things that do stuff. A well known sub-class is Person, representing people. Other kinds of agents include Organization and Group.
SubClassOf	
In-Domain-Of	weblog, icqChatID, msnChatID, account, age, mbox, yahooChatID, tipjar, jabberID, status, openid, gender, interest, holdsAccount, topic, interest, aimChatID, birthday, made, skypeID, mbox_sha1sum
In-Range-Of	lmds:hasAgent

lmds:hasAgent	
URI	http://www.linked2safety.eu/lmds#hasAgent
Definition	An agent who is granted or denied the access control privilege(s)
Domain	lmds:AccessPolicy
Range	foaf:Agent

lmds:hasRequesterProfile	
URI	http://www.linked2safety.eu/lmds#hasRequesterProfile
Definition	An set of attributes that denotes a pattern which requesters must satisfy
Domain	lmds:AccessPolicy
Range	lmds:RequesterProfile

Furthermore, a number of local properties have been introduced to give the data provider the ability to describe a requester's profile including lmds:hasCountry, lmds:hasOrganization, lmds:hasOccupation, lmds:hasWorkingArea, lmds:hasSubscribtionDate, lmds: hasPurpose, and lmds:hasRole.

Similarly, the classes lmds:Country, lmds:Organization, lmds:Occupation, lmds:WorkingArea, lmds:SubscribtionDate, lmds: hasPurpose, and lmds:Role describe properties of the requester's profile that should be met by an expert user

who want to access the data-cubes[25]. Regarding the meaning of each class, they are self reported.

Possible subcategories of the class lmds:Organization can be one of the following: a hospital, a clinical site, a pharmaceutical company, a CRO, academic institute, a medical center, a genetic lab, a bioinformatics company, a health authority, etc.

Finally, there are some properties that inserted to all classes helping interpretation and understanding of the model e.g. properties of rdfs:label and rdfs:comment.

7.3.3. Policies and Rule Description

This section presents a number of examples on how to define access policies.

Example 1. A data provider defines access policies for a set of RDF data-cubes coming from a clinical trial (phase IV) related to lung cancer. He prefers to disclose these data-cubes to anyone acting in similar studies e.g. oncology studies (positive permissions).

```
<ldms:AccessPolicy
rdf:about="http://www.linked2safety.eu/lmds#AccessPolicy1">
      <rdfs:comment>An access policy enforced to a group of RDF data-
      cubes coming from a clinical trial (phase IV) related to lung
      cancer</rdfs:comment>
      <rdfs:label>Access Policy allowing access to anyone acting
      similar studies</rdfs:label>
      <ldms:grantsAccess
      rdf:resource="http://www.w3.org/ns/auth/acl#Read_operation"/>
      <ldms:hasCondition>
            <ldms:Condition
            rdf:about="http://www.linked2safety.eu/lmds#Condition1">
                  <rdfs:comment>The access policy is applied to all
                  RDF data-cubes coming from a clinical trial related
                  to lung cancer named
                  CT_LungCancer_01</rdfs:comment>
                  <ldms:hasSource
                  rdf:resource="http://www.linked2safety.eu/lmds#CT_L
                  ungCancer_01"/>
            </ldms:Condition>
      </ldms:hasCondition>
      <ldms:hasRequesterProfile>
            <ldms:RequesterProfile
            rdf:about="http://www.linked2safety.eu/lmds#RequesterProf
            ile1">
                  <ldms:hasWorkingArea
                  rdf:resource="http://www.linked2safety.eu/lmds#Ongo
                  logy"/>
```

[25] The same attributes are kept for each user registered in the Linked2Safety system in order to be able to check if the profile is met by the user who requests the data-cubes

```
            </ldms:RequesterCriteria>
      </ldms:hasRequesterCriteria>
</ldms:AccessPolicy>
```

Example 2. Another example could be a data provider that prefers applying an access policy to his RDF data-cubes to everyone but those coming from a specific location e.g. coming from ExampleLocation1 because he knows that a competitor Pharma is located there (negative permissions).

```
<ldms:AccessPolicy
rdf:about="http://www.linked2safety.eu/lmds#AccessPolicy2">
      <rdfs:comment
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string">An
      access policy enabling to everyone access the RDF data-cubes
      except for those coming from the same origin to</rdfs:comment>
      <ldms:deniesAccess
      rdf:resource="http://www.w3.org/ns/auth/acl#Read_operation"/>
      <ldms:hasRequesterCriteria>
            <ldms:RequesterCriteria
            rdf:resource="http://www.linked2safety.eu/lmds#RequesterC
            riteria2"/>
                  <ldms:hasCountry
                  rdf:resource="http://www.linked2safety.eu/lmds#Exam
                  pleLocation1"/>
            </ldms:RequesterCriteria>
      </ldms:hasRequesterCriteria>
</ldms:AccessPolicy>
```

Example 3. An access policy will be applied to restrict access to data-cubes if the data-cube's dimension has a specific attribute e.g. dyslipidemia as the data provider is currently working on this area and he doesn't want to disclose relative data-cubes in case of being viewed by a competitor pharmaceutical company working on similar medications.

```
<ldms:AccessPolicy
rdf:about="http://www.linked2safety.eu/lmds#AccessPolicy3">
      <rdfs:comment
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string">An
      access policy which is applied to restrict access to data-cubes
      that contain in their dimension a specific drug named
      ExampleDrug1</rdfs:comment>
      <ldms:deniesAccess
      rdf:resource="http://www.w3.org/ns/auth/acl#Read_operation"/>
      <ldms:hasCondition>
            <ldms:Condition
            rdf:about="http://www.linked2safety.eu/lmds#Condition2">
                  <ldms:hasDimension>
                        <qb:DimensionProperty
                        rdf:resource="http://www.linked2safety.eu/lmd
                        s#dyslipidemia"/>
                  </ldms:hasDimension>
            </ldms:Condition>
```

```
        </ldms:hasCondition>
</ldms:AccessPolicy>
```

8. Data Access Mechanism and Authentication Framework

In this section we are presenting the Data Access Mechanism and the design of a framework that will provide the authentication of expert users as well. In particular, we design Data Access and Authentication components that will allow restricted access to user-authenticated data sources (i.e. data-cubes) which are governed by an Access Policy Model (named LMDS Model), as were described in the previous section. First, we present briefly what has been done in the field of Data Access and Authentication. Second, we present the specifics of the problems we face in the Linked2Safety project. Last, we define the framework that provides the solution to the aforementioned problems.

8.1. Background on Data Access Mechanism and Authentication Framework

In this section we aim to present a short background on related technologies that deal with data access mechanisms.

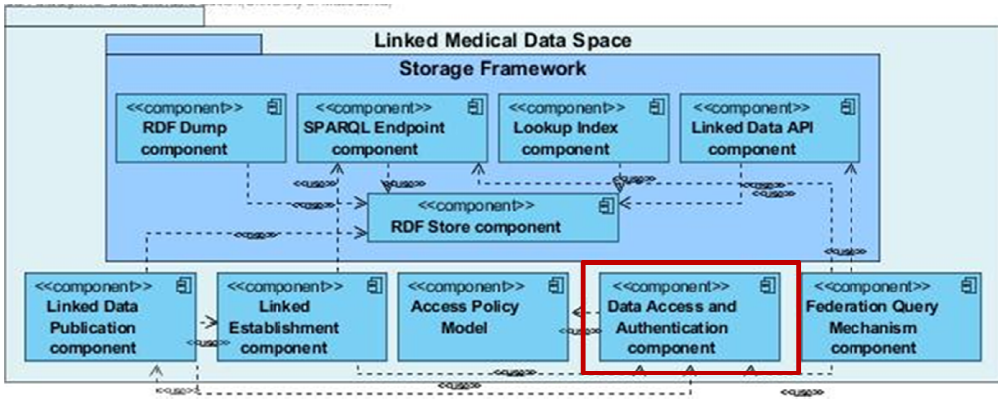

Figure 35: Linked Medical Data Space: Data Access and Authentication

Figure 35 shows the dependencies of the Data Access and Authentication component with the LMDS. Moreover, it aims to identify the most important concepts that should be taken into account for the creation of an adaptable framework on the nature of data that will be controlled, taking into consideration data the Linked2Safety related requirements. We will start by showing a selection of modern approaches in relation to authorisation and authentication.

8.1.1. Security and Authorisation Management

Recently, there have been several proposals that deal with either authorisation and/or authentication.

HTTP Basic Access Authentication is the first authentication method created on the web and the simplest one. The "basic" authentication scheme is based on the model that the client must authenticate itself with a user-ID and a password. The client sends an HTTP request which contains the "Authorisation" Header, a string with the Base64 representation of the concatenated string "username:password". The server will service the request only if it can validate the user-ID and the password for the protection space of the Request-URI. There are no optional authentication parameters. Although the scheme is easily implemented, it relies on the assumption that the connection between the client and server computers is secure and can be trusted.

Another protocol used for authorisation is **OAuth**. The OAuth protocol was originally created by a small community of web developers from a variety of websites and other Internet services who wanted to solve the common problem of enabling delegated access to protected resources. In the traditional client-server authentication model, the client uses its credentials to access its resources hosted by the server. With the increasing use of distributed web services and cloud computing, third-party applications require access to these server-hosted resources.

OAuth introduces a third role to the traditional client-server authorisation model: the resource owner. In the OAuth model, the client (which is not the resource owner, but is acting on its behalf) requests access to resources controlled by the resource owner, but hosted by the server. In addition, OAuth allows the server to verify not only the resource owner authorisation, but also the identity of the client making the request. OAuth provides a method for clients to access server resources on behalf of a resource owner (such as a different client or an end-user). It also provides a process for end-users to authorise third-party access to their server resources without sharing their credentials (typically, a username and password pair), using user-agent redirections.

At the time of writing, though OAuth is a perfect proposal for trilateral application schemes, it is not suitable for Linked2Safety since the latter features will function more a fully distributed multipart environment. Also, OAuth doesn't support any data integrity or security concerns.

8.2. Data Access Mechanism and Authentication Objectives in relation to Linked2safety

The problem we face in Data Access Management for the Linked2Safety project is three-fold: first, we need to be able to understand which users request access to data-cubes (Authentication Mechanism). Second, we need to be able to understand the purposes of requesting access to data-cubes and keep track of such requests (Audit Mechanism). Last, we need to be able to check (based on the LMDS Model of the previous section) the kind of access allowed to a user based on his/her role as well as the restriction assigned to the data (i.e. data-cubes)(Authorisation Mechanism). As far as the data-cube access is concerned, one thing to be noted is that only read access option is applicable in cubes in general.

8.2.1. Security Objectives

Due to the significance of the security factor in this system, there is an emphasised need for secure user identification as well as powerful system authentication and authorisation. More analytically, the various security aspects are the following:

- **Authentication**: The means by which communicating entities (for example, client and server) prove to one another that they are acting on behalf of specific identities that are authorised for access. This ensures that expert users are who they say they are.
- **Authorisation**: The means by which interactions with resources are limited to collections of users or programs for the purpose of enforcing integrity, confidentiality, or availability constraints. This ensures that expert users have permission to perform operations or access data.
- **Data integrity**: The means used to prove that information has not been modified by a third party (some entity other than the source of the information). For example, a recipient of data sent over an open network must be able to detect and discard messages that were modified after they were sent.
- **Data Privacy**: The means used to ensure that information is made available only to users who are authorised to access it. This ensures that only authorised expert users can view sensitive data.
- **Non-repudiation**: The means used to prove that a user performed some action such that the user cannot reasonably deny having done so. This ensures that transactions can be proven to have happened.

8.2.2. Recording Log Information Objectives

The other important aspect of Data Access Mechanism is the management of auditing. An audit trail (or audit log) is a security-relevant chronological set of records, or destination and source of records that provide documentary evidence of the sequence of activities that have affected at any time a specific operation, procedure, or event. Audit records typically result from activities such as health care data transactions, or communications by individual people, systems, accounts, or other entities.

The system should be able to record the identity of operators entering or confirming critical data. The authority to amend entered data should be restricted to nominated persons. Any alteration (if such is possible) to an entry of critical data should be authorised and recorded with the reason for the change. Consideration should be given to building into the system the creation of a complete record of all entries and amendments (an "audit trail"). Also, for quality auditing purposes, it should be possible to obtain clear printed copies of electronically stored data.

8.2.3. Access Enforcement Objectives

The Linked2Safety project is a distributed environment. Each clinical partner has the sole responsibility of storing EHR records securely in their premises, never leaving them accessible via any network infrastructure. The data-cubes (containing aggregated data gathered by processing EHR and medical information) will be stored in a web-accessed triple-store. In this particular endpoint we need to be able to enforce policies that were described in the previous section. The clinical partner might not want to give full read access to everyone. In most cases, they will prefer to have categories of expert users (with roles) that have access to only a partial view of the data-cubes that are stored in the triple-store.

In Figure 36, we present the basic data flow between the basic components involved in the Linked Medical Data Space, as far as requesting data is concerned. Galaxy Server will initiate SPARQL requests to a SPARQL endpoint, allowing data to be retrieved for each corresponding triple store in the data providers premises. In the beginning of this process, we need to make clear whether the requesting agent has access rights to data provided from a clinical provider. Secondly, each of those requests needs to be logged for security monitoring. Last, every part of the responding communication should also provide security and data integrity checking. In yellow areas we present where the issues related to security, policies and authentication are located to the Linked Medical Data Space.

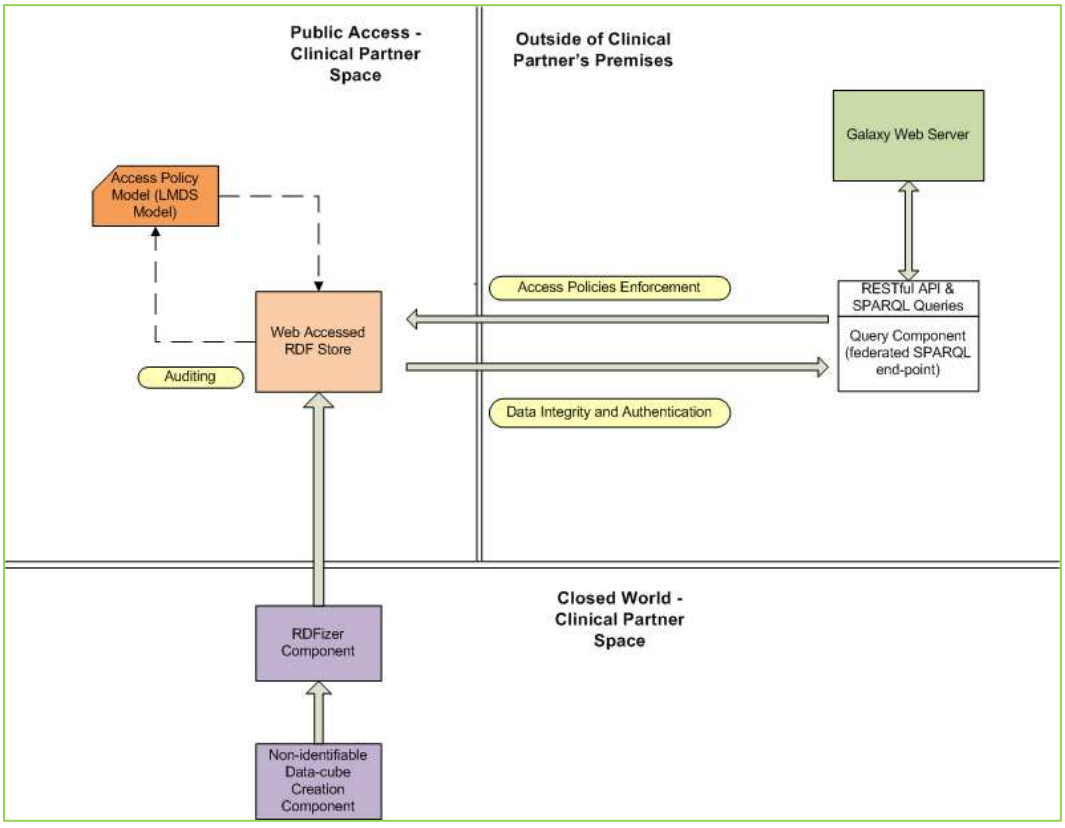

Figure 36: Identifying where the authorisation and authentication processes are located in Linked Medical Data Space

8.3. Design of Data Access Mechanism and Authentication

The most important aspect of security is the mechanism that enforces the aforementioned criteria of security and authentication. This mechanism, will incorporate both ciphering and policy constraints management in order to provide a robust and complete solution. This way address both accessing data-cubes via a policy mechanism, but also, ensure that all communication and data exchange is secure.

8.3.1. Security and Authentication Mechanism

Linked2Safety will provide the authentication, non-repudiation, data privacy and data-integrity facilities by incorporating a Public-Key Infrastructure. The Public Key Infrastructure (PKI) is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates. In cryptography, a PKI is an arrangement that binds public keys with respective user identities by means of a Certificate Authority (CA). The user identity must be unique within each CA domain. The binding is established through the registration and issuance process, which, depending on the level of assurance the binding has, may be carried out by software at a CA, or under human supervision. For each expert user, the user identity, the public key, their binding, validity conditions and other attributes are made un-forgeable in public key certificates issued by the CA.

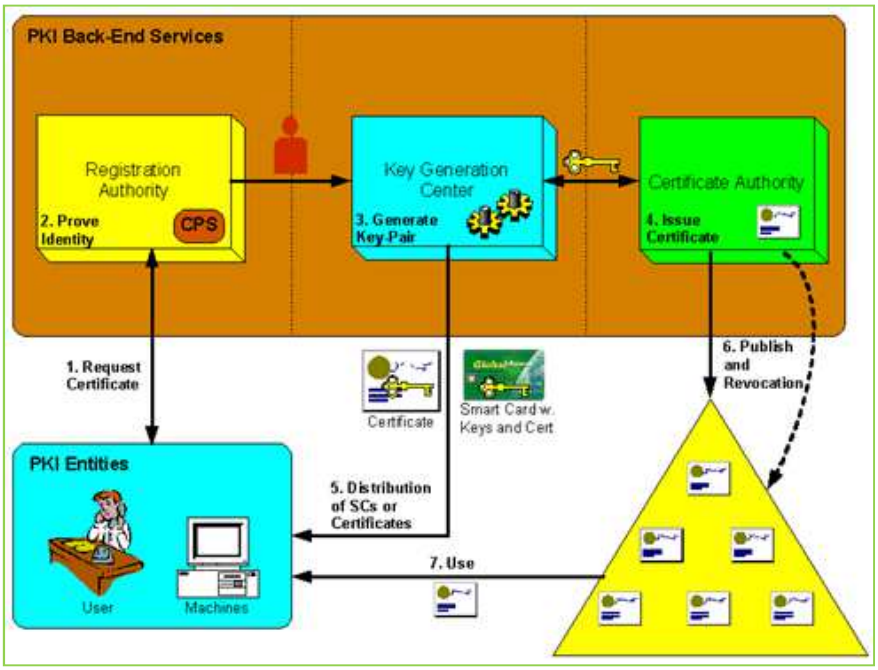

Figure 37: PKI Infrastructure overview Identifying where the authorisation and authentication processes are located in Linked Medical Data Space

Thus, the PKI (Figure 37) enables end-users of a basically insecure public network such as the Internet to securely and privately exchange data through the use of a public and a private cryptographic key pair that is obtained and shared

through a trusted authority, i.e. the Certification Authority. The PKI provides a digital certificate that can identify an individual or an organization and directory services that can store and, when necessary, revoke the certificates.

The PKI assumes the use of public key cryptography, which is the most common method on the Internet for authenticating a message sender or encrypting a message. Traditional cryptography has usually involved the creation and sharing of a secret key for the encryption and decryption of messages. This shared key system has the significant flaw that if the key is discovered or intercepted by someone else, messages can easily be decrypted. For this reason, public key cryptography and the public key infrastructure is the preferred approach on the Internet. The shared key system is sometimes known as symmetric cryptography and the public key system as asymmetric cryptography.

A public key infrastructure consists of:

- A certificate authority that issues and verifies digital certificate. A certificate includes the public key or information about the public key;
- A registration authority that acts as the verifier for the certificate authority before a digital certificate is issued to a requester; and
- A validation authority or a simple directory where the certificates (with their public keys) are held.

The certificate authority will be installed in clinical partners premises, in the server that will serve as a web triple-store.

8.3.2. Auditing Mechanism

To asses this particular issue, we will provide a log mechanism that will keep track information about the query, the expert user and data-cubes returned. As it can be seen in Table 3 we cover all relevant knowledge that helps a clinical partner to check. Each tuple in this file (called Data Access Request Log File) will keep record of the following:

Table 3: Data Access Request Log File contents

Unique Tuple Identifier (incremental number)
Initiation date-time of the non-identifiable data-cube(s) retrieval request
Actor name that initiated the non-identifiable data-cube(s) retrieval request
Actor IP address that initiated the non-identifiable data-cube(s) retrieval request
SPARQL query string that requests data-cubes in Linked Medical Data Space
Number of data-cubes that are returned in this session
Set of Instance URI's of of the non-identifiable RDF data-cube(s) that were

returned
Set of policy rules that were satisfied in order to provide access

8.3.3. Data Access Authorisation Mechanism

In this section we present the process of authorizing an expert user to access specific RDF data-cubes from a clinical partner's web triple-store. The process of allowing an expert user to have any form of access to RDF data-cubes is based on two axes: the first one is to authenticate the user. This is the first most important step, which allows the system to verify that the expert user is who he/she claims to be. This is also a pre-requisite in order to know the role that an expert user has in the Linked2Safety system since after verifying the user, we can extract his/her role. The second axis is to authorise the expert user to access the requested RDF data-cubes if certain criteria based on his/her profile information are met including role, working area, origin, etc.

Before executing any kind of data analysis algorithms, the expert user needs to log-in to the system by providing a username and password or certificate. The expert user is logged in to the Galaxy Platform where he/she can create a workflow specifying the type of analysis s/he wants to use, combining several components together. This in turn will initiate the creation of a SPARQL query that handles the logic of requesting specific criteria. In this point, we should note that the access rights assigned to an expert user when he/she logs-in to the Galaxy Platform depend on his/her role. An extensive description of the Role Framework implemented in Linked2Safety was given in the previous section.

Then, the LMDS Model (presented in the previous section) is enforced upon the SPARQL queries in order to check whether the SPARQL endpoint can grant access to the expert user for the specific data-cubes requested. Finally, only authorised RDF data-cubes will be retrieved. The Policy Credentials Authorisation Algorithm takes into account the confidentiality assigned to the data-cubes as well as requester's criteria such as the role that has been assigned to the expert user (presented in Table 4):

Table 4: Pseudo-code for the Policy Credentials Authorisation Algorithm

a. Authenticate User (USER) via the PKI infrastructure b. For each corresponding clinical partner (CLINICAL_PARTNER) Web-access triple-store endpoint, find requested RDF data-cubes of interest c. For each RDF data-cube (DC), check if expert user and associated role has grant access (all following pseudo-rules must apply): *AccessPolicy->appliesToDataset(DC)*

AccessPolicy->Condition->hasProvider(CLINICAL_PARTNER)

AccessPolicy->hasRequesterCondition->hasRole(RoleOf(USER))->hasName(NameOf(USER))

AccessPolicy->grants Access->AccessRead

The final step in this process is providing the data integrity as well as the necessary privacy for the data-cubes to be transferred: we begin by signing the hash-digest of each data-cube with the private key of the clinical partner. This allows the client who has issued the query to verify that the data have not been altered from the source to the end of the communication transfer. The next step is to actually provide privacy for the data integrity and the data-cubes. For each data-cube ADC that has passed the authorisation sign it and encrypt it:

SendClient(encrypt(sign(SHA1(ADC)) || ADC))

We achieve the security by encrypting each data-cube we send (along with the signed hash-digest) with the public key of the client who requested the data (from his/her certificate).

The whole process can be seen in the Sequence Diagram in Figure 38.

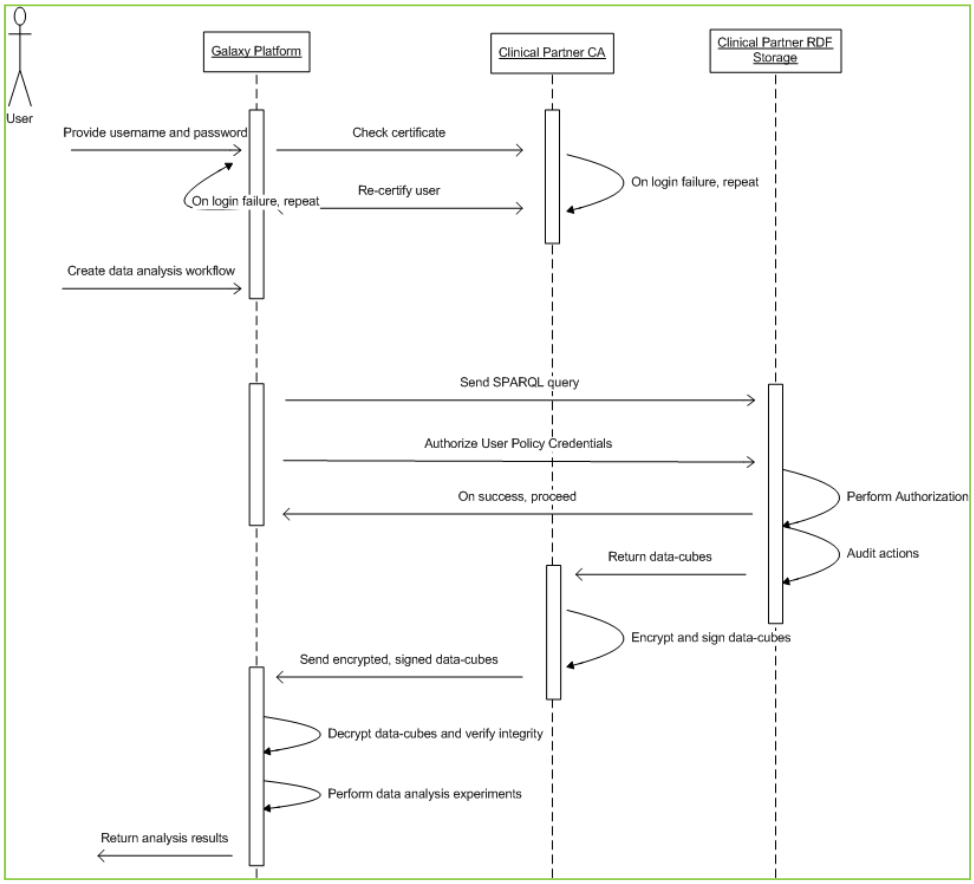

Figure 38: Data Access Mechanism

Upon successful verification, the expert user is authorised to perform that particular query over the SPARQL endpoint. The results are signed and encrypted by the clinical partner before returned to the expert user, using the certificates.

Galaxy automatically verifies the origin of the data (non-repudiation), the source that were sent from (authentication) and decrypts them (data integrity), before providing them to the workflow created by the user.

9. Conclusion

This deliverable D4.1 presents the design specification for the semantically-interlinked Linked Medical Data Space (LMDS) of the Linked2Safety project. The deliverable first presents the integration scenario describing various heterogeneous resources used by the clinical partners. Second, the deliverable presents the storage framework describing arrangement of the storage components within the LMDS. Third, the deliverable presents the linking principles and establishing links between internal and external links to the Linked Data Cloud. Fourth, the deliverable presents the querying component describing federation of a query over distributed clinical resources. Then, the deliverable describes the access policy model that allows user-restricted access to clinical resources. Finally, the deliverable presents the security component enforcing user-authentication mechanism on top of the access policy model.

The LMDS will contribute to the realisation of the Linked2Safety vision by providing a semantically-interlinked space of clinical resources. Specifically the LMDS will be utilized in:

- WP5: In the statistical data mining of heterogeneous and distributed data-cubes from various clinical partners.
- WP7: In showing the usability and applicability of the Linked2Safety project.

The LMDS design specification presented in this deliverable will be implemented as part of the deliverables 4.2.1 and 4.2.2.

10. References

[1] R. Cyganiak and D. Reynolds, “The RDF Data Cube Vocabulary, W3C Working Draft,” World Wide Web Consortium (W3C), Apr. 05 2012.

[2] J. Carroll, C. Bizer, P. Hayes and P. Stickler, “Named graphs,” *Journal of Web Semantics,* vol. 3, no. 4, pp. 247-267, 2005.

[3] T. Berners-Lee, “Design issues: Linked Data,” 2006. [Online]. Available: http://www.w3.org/DesignIssues/LinkedData.html.

[4] R. S. Sandhu and P. Samarati, “Access control: principles and practice , vol. , pp. ,,” *IEEE Communications,* vol. 32, pp. 40-48, 1994.

[5] P. Samarati and S. Vimercati, “Access Control: Policies, Models and Mechanisms vol. 2171, R. Focardi and R. Gorrieri, Ed.: , , pp.,” *Foundations of Security Analysis and Design (Tutorial Lectures),* vol. 2171, pp. 137-196, 2000.

[6] R. Sandhu and Q. Munawer, “How to do discretionary access control using roles,” in *ACM Workshop on Role-Based Access Control*, Fairfax, VA, 1998.

[7] R. S. Sandhu, “Lattice-based access control models,” *IEEE Computer,* vol. 26, pp. 9-19, 1993.

[8] P. A. Bonatti and P. Samarati, “A uniform framework for regulating service access and information release on the web,” *Journal of Computer Security,* vol. 10, pp. 241-271, 2002.

[9] N. Yang, H. Barringer and N. Zhang, “A purpose-based access control model,” in *3rd International Symposium on Information Assurance and Security (IAS)*, 2007.

[10] J.-W. Byon, E. Bertino and N. Li, “Purpose-Based Access Control of Complex Data for Privacy Protection,” in *10th ACM Symposium Access Control Models and Technologies*, 2005.

[11] S. Oh and S. Park, “Task-role-based Access Control Model,” *Information Systems,* vol. 28, pp. 533-562, 2003.

[12] A. V. D. M. Kayem, P. Martin, S. G. Akl and P. W., “A Framework for Self-Protecting Cryptographic Key Management,” in *2nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, Venice, Italy, 2008.

[13] C. Anutariya, S. Chatvichienchai, M. Iwaihara, V. Wuwongse and Y. Kambayashi, "A Rule-Based XML Access Control Model," in *2nd Workshop on Rules and Rule Markup Languages for the Semantic Web*, 2003.

[14] G. Motta and S. Furuie, "A contextual role-based access control authorization model for electronic patient record," *IEEE Transactions on Information Technology in Biomedicine,* vol. 7, pp. 202-207, 2003.

[15] D. Ferraiolo and R. Kuhn, "Role-Based Access Controls," in *NIST-NSA National (USA) Computer Security Conference*, 1992.

[16] O. Sacco and A. Passant, "A Privacy Preference Ontology (PPO) for Linked Data," in *Linked Data on the Web Workshop (LDOW 2011)*, 2011.

[17] O. Sacco and A. Passant, "A Privacy Preference Manager for the Social Semantic Web," in *2nd Workshop on Semantic Personalized Information Management: Retrieval and Recommendation (SPIM 2011), Workshop at The 10th International Semantic Web Conference (ISWC 2011)*, 2011.

[18] S. Villata, N. Delaforge, F. Gandon and A. Gyrard, "An access control model for linked data," in *7th international IFIP workshop on semantic web& web semantics (SWWS 2011)*, 2011.

[19] S. Villata, N. Delaforge, F. Gandon and A. Gyrard, "Social semantic web access control," in *4th international workshop social data on the web (SDoW 2011)*, 2011.

[20] H. Muhleisen, M. Kost and J. C. Freytag, "SWRL-based Access Policies for Linked Data," in *2nd Workshop on Trust and Privacy on the Social and Semantic Web*, 2010.

[21] S. Javanmardi, M. Amini and R. Jalili, "An access control model for protecting semantic web resources," in *2nd International Semantic Web Policy Workshop (SWPW 06)*, Athens, GA, USA, 2006.

[22] M. Buffa and C. Faron-Zucker, "Ontology-Based Access Rights Management," *Advances in Knowledge Discovery and Management Studies in Computational Intelligence,* vol. 398, pp. 49-61, 2012.

[23] F. Giunchiglia, R. Zhang and B. Crispo, "Relbac: Relation based access control," in *4th International Conference on Semantics, Knowledge and Grid*, 2008.

[24] J. Hollenbach, J. Presbrey and T. Berners-Lee, "Using RDF Metadata to enable Access Control on the Social Semantic Web," in *Collaborative Construction, Management and Linking of Structured Knowledge*, 2009.

[25] E. Fernández-Medina, J. Trujillo, R. Villarroel and M. Piattini, “Access control and audit model for the multidimensional modeling of data warehouses,” *Decision Support Systems,* vol. 42, pp. 1270-1289, 2006.

[26] T. Priebe and G. Pernul, “A Pragmatic Approach to Conceptual Modeling of OLAP Security,” in *20th International Conference on Conceptual Modeling*, Yokohama, Japan, 2001.

[27] E. Weippl, O. Mangisengi, W. Essmayr, F. Lichtenberger and W. Winiwarter, “An authorization model for data warehouses and OLAP,” in *Workshop on Security in Distributed Data Warehousing*, New Orleans, Louisiana, USA, 2001.

[28] R. Kirkgöze, N. Katic, M. Stolda and A. M. Tjoa, “A security concept for OLAP,” in *8th International Workshop on Database and Expert System Applications*, Toulouse, France, 1997.

[29] G. Pernul and T. Priebe, “Towards olap security design - survey and research issues,” in *3rd ACM International Workshop on Data Warehousing and OLAP (DOLAP'00)*, 2000.