Integrative Cancer Research
INTEGRATE
Through Innovative Biomedical Infrastructures

SEVENTH FRAMEWORK PROGRAMME

ICT-2010-270253

INTEGRATE

Driving excellence in Integrative Cancer Research through Innovative Biomedical Infrastructures

STREP
Contract Nr: 270253

Deliverable: D2.7 Final System Architecture and Security Framework

Due date of deliverable: 01-31-2014
Actual submission date: 01-31-2014

Start date of Project: 01 February 2011 Duration: 36 months

Responsible WP: Custodix

Revision: final

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Service	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (excluding the Commission Services)	

0 DOCUMENT INFO

0.1 Author

Author	Company	E-mail
Kristof De Schepper	Custodix	kristof.deschepper@custodix.com
Brecht Claerhout	Custodix	brecht.claerhout@custodix.com
Elias Neri	Custodix	Elias.neri@custodix.com
Marita Borms	Custodix	marita.borms@custodix.com
Jasper van Leeuwen	Philips	jasper.van.leeuwen@philips.com
Anca Bucur	Philips	anca.bucur@philips.com
Georgios Manikis	FORTH	gmanikis@ics.forth.gr
Ioannis Karatzanis	FORTH	karatzan@ics.forth.gr
Evaggelia Maniadi	FORTH	maniadi@ics.forth.gr
Thanasis Iliopoulos	FORTH	ailiop@ics.forth.gr
Sergio Paraíso	UPM	sparaiso@infomed.dia.fi.upm.es
David Pérez del Rey	UPM	dperez@infomed.dia.fi.upm.es
Raúl Alonso	UPM	ralonso@infomed.dia.fi.upm.es
Philippe Hennebert	IJB	philippe.hennebert@bordet.be

0.2 Documents history

Document version #	Date	Change
V0.1	22/11/2013	Starting version, template
V0.2	22/12/2013	Definition of ToC
V0.3	20/01/2014	First complete draft
V0.4	21/01/2014	Integrated version (send to WP members)
V0.5	21/01/2014	Updated version (send PCP)
V0.6	27/01/2014	Updated version (send to project internal reviewers)
Sign off	31/01/2014	Signed off version (for approval to PMT members)
V1.0	31/01/2014	Approved Version to be submitted to EU

0.3 Document data

Keywords	Architecture, view, viewpoints, final, security framework
Editor Address data	Name: Kristof De Schepper Partner: Custodix Address: Kortrijksesteenweg 214, 9830 Sint-Martens-Latem Belgium Phone: +32 (0) 9 210 78 90 Fax: +32 (0) 9 211 09 99 E-mail: kristof.deschepper@custodix.com
Delivery date	31/01/2014

0.4 Distribution list

Date	Issue	E-mailer

1 Table of Contents

2 Introduction

This deliverable specifies the final version of the architectural description and the security framework of the INTEGRATE project. It is result of an iterative interaction process between end-users, legal people and software architects. Starting from the initial version of the architecture and security framework (described in deliverable 2.4), the content have been gradually extended, corrected and improved. This work was documented in the intermediate architectural deliverable 2.6 and finally in this document.

The deliverable is divided in two main distinct parts:

- The INTEGRATE architecture description
- The INTEGRATE security framework

2.1 PART I – Architectural Description

The input for the INTEGRATE architectural description was provided by three main input sources (see Figure 1):

- Technical use cases: deliverable 1.5 (and his predecessor 1.4) described technical use cases that provided an initial decomposition of the end-user scenarios.
- Evaluation of the previous iterations of the architecture: comments raised during the review meetings of year 1 and 2 were implemented. Next to this several sections where specified more in detail.
- Feedback from the demonstrators: the implementation of the architecture provided feedback about technical limitations, suggested alternatives and requests for refinement.

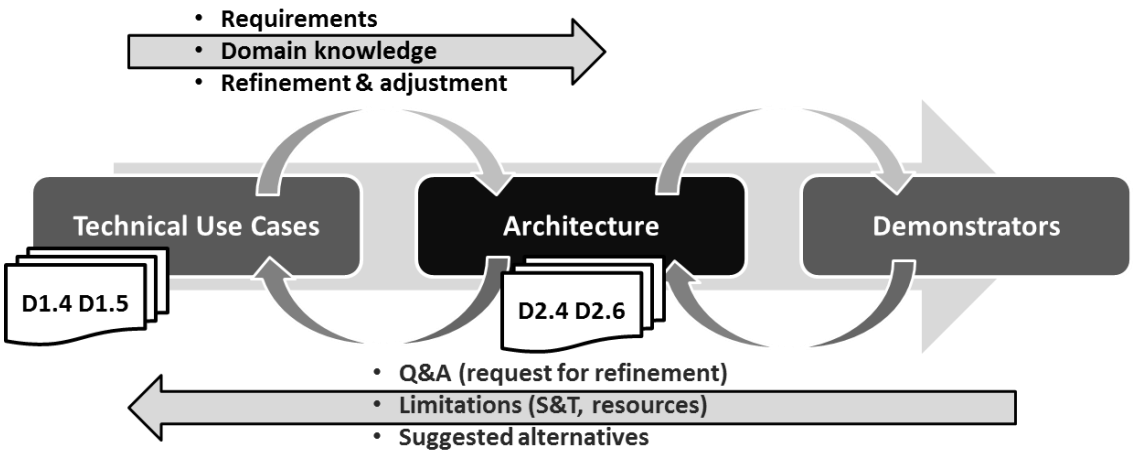


Figure 1: Deliverable 2.7 Input sources

The INTEGRATE architectural description follows the principles of the View - Viewpoint Model, as formalised in *ANSI/IEEE 1471-2000, ISO/IEC 42010:2007*. This model enables architects to define and comprehend complex architectures. Central in this model is the concept of view. A view is a representation of a system from the perspective of a set of related concerns (expressed by the stakeholders). The set of

conventions on how to construct, interpret and use a view is called a viewpoint. A viewpoint specifies the models to be used for describing the concepts that are relevant to that view (e.g. UML static structure diagram used in the information model view). Some views may cover concerns that affect many of the other views (called cross-cutting concerns). A typical example is a security view, which is likely to interact with many other views (e.g. functional, operational, development...).

What views are best suited for describing a software architecture, is a decision that is in general left up to the architects. However, there are quite some reference models (frameworks) that bundle some commonly used sets of views like the 4-1 View Model[1] and three schema approach[2]. This document does not follow one of the reference model, but defines its own viewpoints (and their content) which suit to describe the particularities of INTEGRATE (e.g. the focus on "semantic integration" is rather specific in the INTEGRATE context).

This document follows the principles laid down by the IEEE specification, but does not strictly adhere to it. Given the (research) nature of the project, the latter would cause a lot of overhead without bringing much added value to the project. For example, providing a tight specification of the viewpoints is one such task which is very resource consuming, but would not add to the project. Apart from that, it should be noted that "adhering to the principles, but not strictly to the specification" is common practice in software development teams.

In this document three views were identified to be useful for the INTEGRATE architecture description: the functional, information and deployment view. The definition of these views can be found in the corresponding underlying sections.

2.2 PART II – Security Framework

The second part of this document gives the final iteration of the INTEGRATE security framework. It provides a technological solution that covers all identified security requirements and guarantees compliance of the complete INTEGRATE platform to the legal framework governing the project. For this, modular components were developed dealing with authentication, authorisation, audit and privacy enhancing techniques.

[1] The “4+1” View Model of Software Architecture. *Philippe, Kruchten. November 1995, IEEE Software 12, pp. 42-50.*

[2] The ANSI/SPARC DBMS Mode. Jardine, Donald A. s.l. : North-Holland Pub. Co., 1977. ISBN 0 7204 0719 2

3 (PART I) System Stakeholders

System stakeholders are people or organisations that take a particular interest in a platform. Each of them has particular concerns relating to their perspective on the system. Identification of these stakeholders and their associated concerns is an important step when designing a system and thus also part of the architecture description.

Figure 2 gives an overview of the stakeholders identified in INTEGRATE, The underlying tables provide a short explanation of each stakeholder. These stakeholders and their concerns are to be seen in the context of the use of INTEGRATE by the Breast International Group (BIG). However, exploitation needs to go beyond that use, be it through exploitation of the system as a whole or by exploitation of individual components (see Deliverable 7.4). Future exploitation (reflected in "usefulness", "scalability", "genericness" ...) is a concern of all system-owners and is not further discussed.

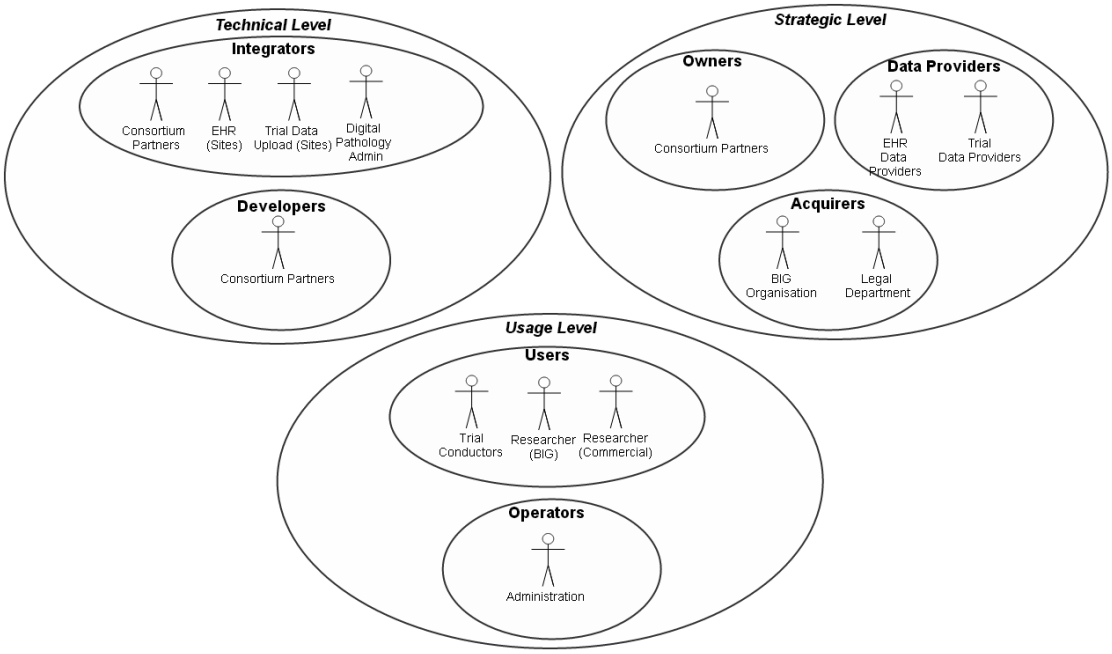


Figure 2: Overview Stakeholders

3.1 Strategic Level Stakeholders

Role	Stakeholder	Description
Acquirers	BIG Organisation	People who provide and prioritise scenarios, used as input for designing the INTEGRATE platform. They also check if the proposed requirements coming from the scenarios are fulfilled by the end of the project.
	BIG Legal Department	Legal people from BIG that define the legal, ethical and regulatory requirements for the INTEGRATE platform. They will check if these requirements are met at the project ending.

Data Providers	EHR Data Providers	CIO's and directors responsible for the EHR's at the local sites which plan to provide access to EHR data.
	Trial Data Providers	Site managers at the local BIG sites which plan to provide trial data in the INTEGRATE research environment.
Owners	Consortium Partners	The consortium partners who own the different components of the INTEGRATE platform

3.2 Usage Level Stakeholders

Role	Stakeholder	Description
Users	Trial conductors	End-users that interact with the INTEGRATE platform as part of the molecular testing and central review (see D1.2[4]), this includes: - **Investigator**: A treating clinician or a trial nurse, not necessarily part of the trial, but acts in the clinical care domain. - **Central laboratory member**: person who performs molecular tests - **Clinical data manager**: person that gathers clinico-genomic data from completed trials and uploads them to the platform - **Reviewer**: a pathologist working on central review of pathology images
	Researcher (BIG)	End-users, associated to BIG, which interact with the "research"-part of the INTEGRATE platform, meaning the services which provide access to the aggregated research data (secondary use). They perform queries on the INTEGRATE repositories, download and analyse data.
	Researcher (Commercial)	Same as above, but member of a commercial organisation (typically pharma customers).
Operators	Administrators	People responsible for administrating the INTEGRATE environment (platform infrastructure and application services) once it is deployed.

3.3 Technical Level Stakeholders

Role	Stakeholder	Description
Developers	Consortium Partners	Those responsible for developing the technical solutions to be deployed.
Integrators	Consortium Partners	Those responsible for integrating the

		technical solutions and making the platform deployable.
	EHR integrators (@sites)	Technicians in charge of linking the EHR at the local sites with the INTEGRATE platform, i.e. making the EHR site compliant with the INTEGRATE interfaces.
	CDMS (Clinical data management system) integrators (@sites)	Technicians charged with enabling trial data upload into the INTEGRATE platform, i.e. making the upload process compliant with the INTEGRATE interfaces.
	Digital pathology administrator	Technicians in char of exporting digital pathology images.

4 System Concerns

Each of the stakeholders has specific concerns about the system, typically fitting one of the following categories (corresponding to quality attributes): functionality, feasibility, usage, system purposes, system features, system properties, known limitations, structure, behaviour, performance, resource utilisation, reliability, security, information assurance, complexity, evolvability, openness, concurrency, autonomy, cost, schedule, quality of service, flexibility, agility, modifiability, modularity, control, inter-process communication, deadlock, state change, subsystem integration, data accessibility, privacy, compliance to regulation, assurance, business goals and strategies, customer experience, maintainability, affordability and disposability.

The tables below list the set of most important technical concerns associated with their respective stakeholder for the INTEGRATE platform. These concerns are further addressed in the different views.

4.1 Strategic Level Concerns

Role	Stakeholder	Concern	ID
Acquirers	BIG legal department	The INTEGRATE framework must comply with the legal, ethical and security requirements defined in INTEGRATE legal framework (deliverable D1.3).	CAC-001
	BIG organisation	*Currently no direct concerns identified for BIG organisation.*	CAC-002
Data Providers	EHR Data Providers	Offering EHR data access to the INTEGRATE platform that complies with the local regulations of the providing site.	CDP-001
	Trial Data Providers	Providing trial data to the INTEGRATE platform that complies with the local regulations of the providing site.	CDP-002

4.2 Usage Level Concerns

Role	Stakeholder	Concern	ID
Users	Trial Conductors	All the defined requirements concerning the molecular screening and central pathology review functionality are available in the INTEGRATE platform.	CUS-001
	Researcher (BIG)	All the defined requirements concerning the trial data querying and analytical tools functionality are available in the INTEGRATE platform.	CUS-002
		The overall performance of the trial data querying and analytical tools systems in the INTEGRATE platform. More specifically the systems should provide a good quality of service and responsiveness to the end-user.	CUS-005
	Researcher	All the defined requirements concerning the	CUS-003

	(commercial)	trial data querying and analytical tools functionality are available in the INTEGRATE platform.	
		The overall performance of the trial data querying and analytical tools systems in the INTEGRATE platform. More specifically the systems should provide a good quality of service and responsiveness to the end-user.	CUS-006
	Trial Conductors	The overall performance of the molecular screening and central pathology systems in the INTEGRATE platform. More specifically the systems should provide a good quality of service and responsiveness to the end-user.	CUS-004
Operators	Administrators	As an end-user, the administrator needs an idea of the functionality of the platform to assess the scope of the administration tools.	COP-001

4.3 Technical Level Concerns

Role	Stakeholder	Concern	ID
Developers	Consortium partners	Having flexible and modular interfaces/components in the INTEGRATE platform. By defining these interfaces/components, the platform functionality becomes clear to each of the partners. The split up in components makes it possible to define partner responsibilities and tasks to each component at the start of the implementation phase. Finally the overall complexity of the platform becomes visible, in this way resources can be allocated by the partners for each component.	CDE-001
		Security components of the INTEGRATE platform provide generic interfaces, so security can be integrated in the INTEGRATE services in a relatively straightforward way.	CDE-002
		Knowing the structure and content of all data and meta-data available in the INTEGRATE platform in order to correctly query/manipulate them and tune the interfaces of the different architectural components that are exchanging them.	CDE-003
Integrators	Consortium partners	Connecting the separately developed software blocks of the INTEGRATE platform into one integrated system	CIN-001
	EHR integrators (@sites)	Make the EHR datawarehouses interfaces (situated on the sites) compliant to the requirements of the INTEGRATE platform.	CIN-002

	CDMS integrators (@sites)	Compliance with the trial data upload process (situated on the sites) to the interfaces of the INTEGRATE platform.	CIN-003

5 Overview

The main objective of the INTEGRATE framework is to build a technical solution that covers the needs of the different end-users and maximises the probability of sustainability and exploitation. This requires that the INTEGRATE framework should be flexible enough to deal with multiple business models, in other words integrators of the platform should be able to instantiate their own INTEGRATE compatible-/based infrastructure instead of be limited to one possible instance of the platform.

In order to get this flexibility the INTEGRATE architecture leverages several design principles for the INTEGRATE services and modules in the design and development phase of the INTEGRATE software development life-cycle:

- The services and modules developed in the framework should be designed according to the principles of a loosely coupled, open and scalable Service Oriented Architecture (SOA), focusing mainly on the interoperability and interfacing between the different systems and services. The loosely coupled approach will limit the level of knowledge that one component in the system needs to know about other.
- The services and modules developed in the framework should adopt international standards as much as possible (see deliverable 2.1).

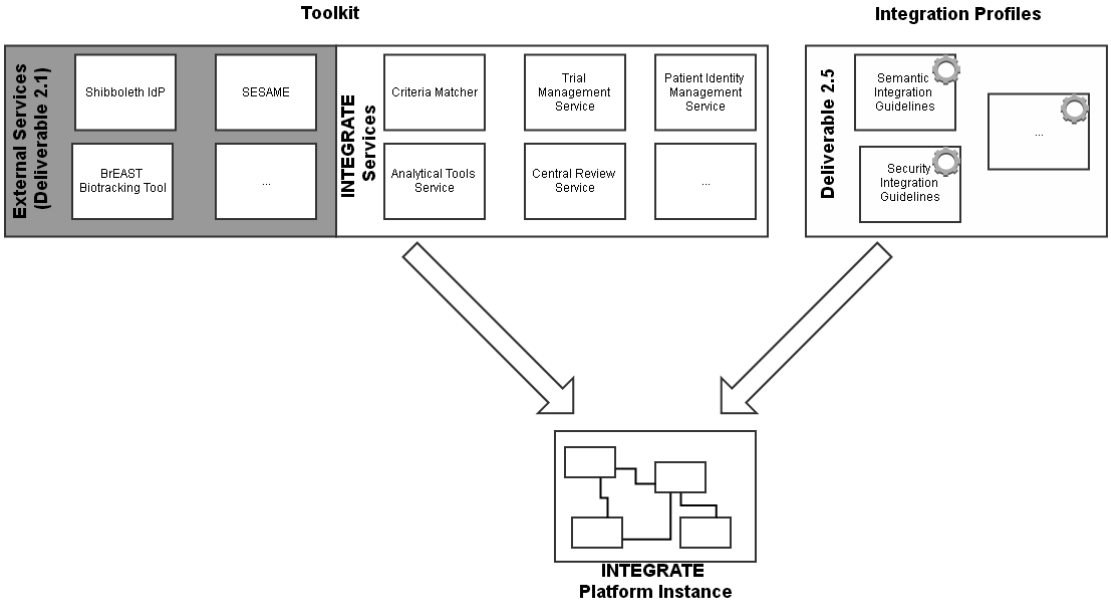


Figure 3: INTEGRATE Framework

The services and modules developed during the implementation phase can be seen as the **basic toolkit** of the INTEGRATE framework (see Figure 3). This basic toolkit can be extended with existing “non-INTEGRATE” services and modules (see deliverable 2.2 for an overview) as long as they are compliant with the specification of the INTEGRATE framework. Each service and module in the toolkit can be placed in many different deployment configurations, thanks to the flexibility.

This flexibility in the deployment comes at a price. With each new service or module in the toolkit, the amount of deployment possibilities raises. For integrators of the system

deployment can become a complex task (how to determine good deployment configurations). For this INTEGRATE will guide the integrators by offering **integration profiles**, describing commonly used deployment configurations for a subset of the toolkit (see deliverable 2.5).

5.1 Logical Architecture

The INTEGRATE platform is designed as a multi-layered architecture, with responsibilities assigned to the various architectural layers. Every component/service designed within INTEGRATE can be mapped to one of these layers (or spanned over multiple layers). Figure 4 shows four horizontal and one vertical layers that can be distinguished in the INTEGRATE architecture.

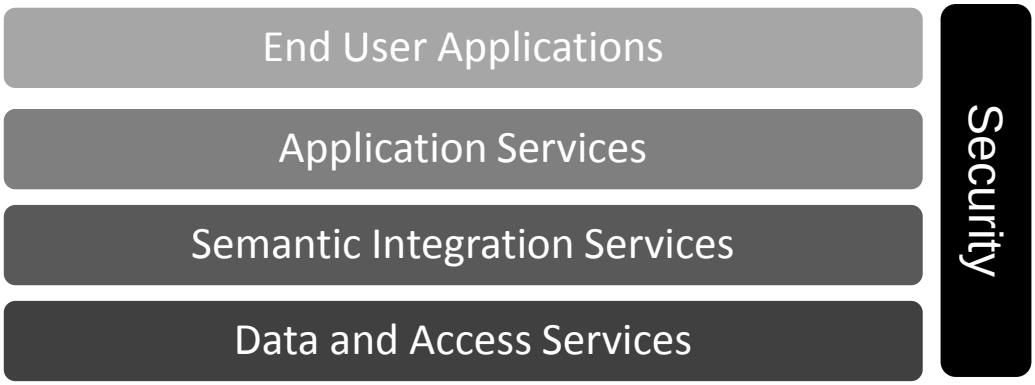


Figure 4: Multi-layered architecture

The **presentation layer** is the top layer. The components situated in this layer can be seen as endpoints to the end users of the system, presenting the underlying back-end functionality in an intuitive and user-friendly way. The components typically make use of a (advanced) graphical user interface (GUI) for displaying (complex) back-end functionality.

The **business layer** provides the core functionality of the INTEGRATE services as it houses a variety of application services. The components contain the functional algorithms that handle information exchange between the semantic integration layer and the presentation layer. Where possible, INTEGRATE promotes the approach of providing re-usable services in the application layer.

The **semantic integration layer** utilises the ontology based information model and translates or maps the model to the underlying data and information sources. The semantic integration layer will abstract the underlying data sources for the upper application layers. Next to providing a uniform data access method, this layer will present data to applications according to a single central data model with well understood semantics according to the Core Dataset (see deliverable D3.5 initial proposal semantic interoperability layer). This ensures a clear separation of concerns between integration of data sources and building of applications that make use of these data sources. Integration of new data sources or new information content of a data source should be done towards a common information model, regardless of the application. Applications can be developed in a generic way, based solely on the common information model.
More information on this layer is in section 6.3.

The **data layer** contains the various data and the metadata repositories. Services on this layer are responsible for the actual data access. The data warehouses all expose a standardized query interface, and queries are expressed using the INTEGRATE core dataset.

The **security layer** is typically a vertical layer as security impacts the architecture at all levels. Security is discussed in part II of this deliverable.

5.2 1.000 Feet View

In Figure 5 the main components identified in the INTEGRATE framework are mapped to the layers that were described above. This figure gives an overview of how the main INTEGRATE components interact. A more detailed description of each of these components is given in the functional view.

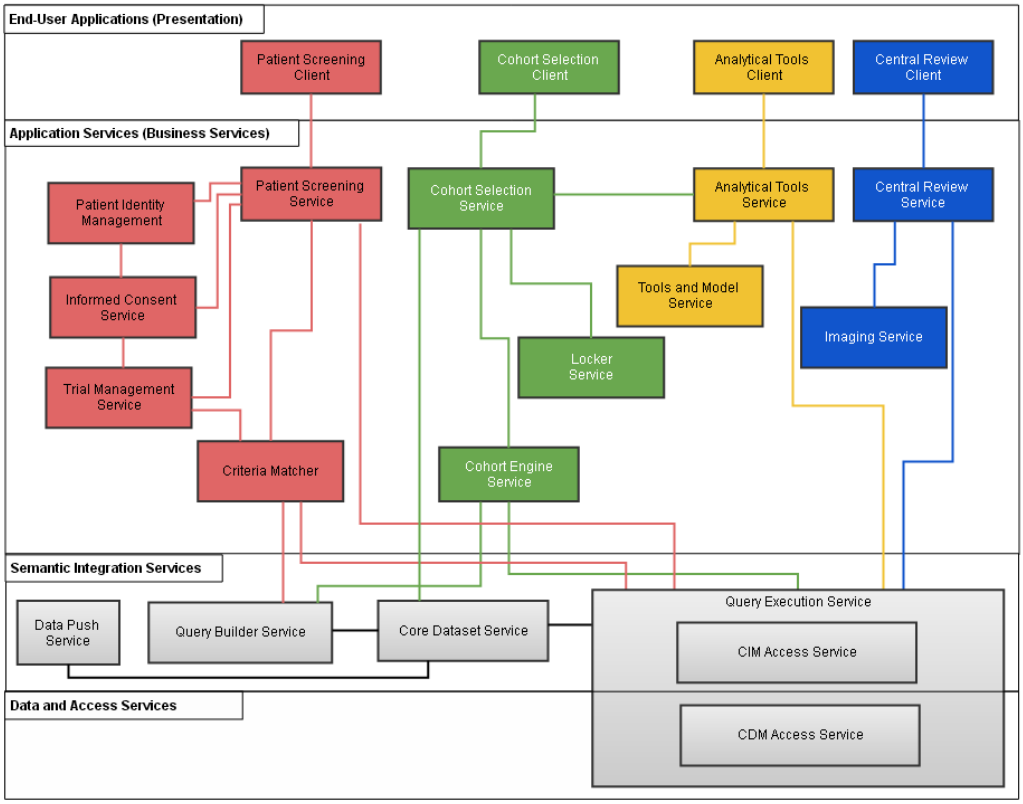


Figure 5: 1.000 feet view INTEGRATE

6 Functional View

6.1 Patient Screening

6.1.1 Introduction

Most of the functional related concerns that stakeholders have regarding the *molecular testing scenario* (see deliverable 1.2) are addressed in the patient screening view. For this it offers a set of general architectural building blocks. Starting from the use cases (see deliverables 1.4 and 1.5), six main architectural screening process components were identified. Some of these components are linked with components defined outside the screening process view. The main functionality of and connections between these components are explained in the next subsections.

Concerns addressed in this view (see paragraph 4)
CDE-001, CUS-001 *(1)*, **COP-001** *(2)*, **CAC-001** *(3)* *(1) From the point of the trial conductors (end-users), this view shows the main features of each screening component. The available functionality to an end-user is defined in the interfaces between these users and the components. Also the main interaction between the components gives a general idea of the behaviour of the platform to the end-user.* *(2) In this view the patient identity management service, trial management service and informed consent service provide administrator oriented interfaces.* *(3) In order to comply with legal requirements, an informed consent is needed before screening can be conducted. This means that a component for registration of informed consents must be available.*

6.1.2 Diagram

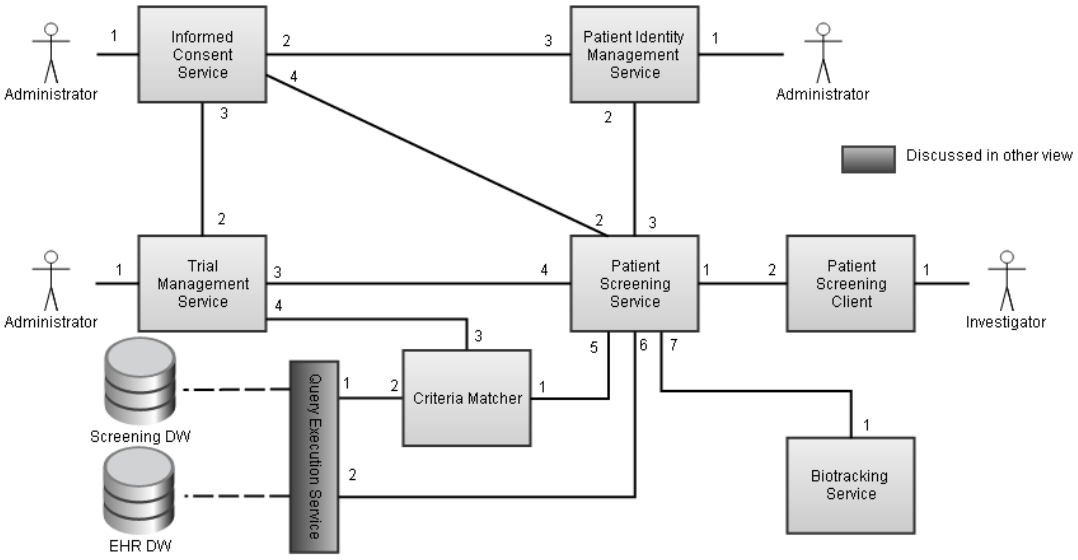


Figure 6: Screening Functional View

6.1.3 Components and Interfaces

6.1.3.1 Patient Screening Client (end-user application)

An investigator (end-user), that wants to check a patient's eligibility for a trial, will interact with the system through the patient screening client. This interaction is based on a provided advanced graphical interface which enables the investigator to screen a patient in an intuitive and user-friendly way.

Related to use cases: **UC.1**, **UC.22**, **UC.22.b**

Interface	Description
1	An investigator interacts with the front-end part (GUI based) of the patient screening client. This client exposes following functionality: • A step by step visualisation for checking the patient's eligibility to be enrolled in a trial
2	*No interface exposed to the patient screening* service

6.1.3.2 Patient Screening Service

The patient screening service is the main driver component in the screening process view. It is a service that integrates and connects the different services that are needed to meet the specified requirements of the molecular testing scenario.

Related to use cases: **UC.1**, **UC.22**, **UC.22.b**

Interface	Description
1	The patient screening service exposes following functionality to the patient screening client: • The functional part of the step by step process for checking the patient's eligibility to be enrolled in a selected trial

2	*No interface exposed to the informed consent service*
3	*No interface exposed to the patient identity management service*
4	*No interface exposed to the trial management service*
5	*No interface exposed to the criteria matcher*
6	*No interface exposed to the query execution* service
7	*No interface exposed to the biotracking service*

6.1.3.3 Informed Consent Service

The molecular testing scenario specifies that an investigator should be able to register informed consent to the INTEGRATE platform. The informed consent service is responsible for managing this task. It offers functionality for registering and listing informed consent forms for a patient. Next to this, informed consent configurations are generated, these group informed consent of the same type in one configuration. Using these configurations will make the informed consent service more generic. Finally there is a verification tool to verify if an informed consent is registered for a particular purpose.

Related to use cases: **UC.IC.***

Interface	Description
1	An administrator interacts with the front-end part (GUI based) of the informed consent service. This front-end exposes following functionality: • Create, activate, list and edit informed consent configurations for a trial
2	*No interface exposed to the patient identity management service*
3	*No interface exposed to the trial management service*
4	The informed consent service exposes following functionality to the: • Verify if an informed consent is registered for a particular purpose for a patient and a selected trial

6.1.3.4 Patient Identity Management Service

Patients that are selected for trial screening need to be managed in the INTEGRATE platform according to the molecular testing scenario. The patient identity management service is responsible for registering, consulting and editing patients during this molecular screening. This service is closely connected with the authentication and pseudonymisation/de-identification components (not shown in the figure).

Related to use cases: **UC.2**, **UC.20**

Interface	Description
1	An administrator interacts with the front-end part (GUI based) of the patient identity management service. This front-end exposes following functionality: • Register a new patient on the platform • Edit the information of a patient of the platform • List the registered patients in the platform • Get detailed information of a selected patient
2	The patient identity management service exposes following functionality to the patient screening service: • Register a new patient in the platform

	• List the registered patients in the platform • Get detailed information of a selected patient
3	The patient identity management service exposes following functionality to the informed consent service: • List the registered patients in the platform

6.1.3.5 Trial Management Service

When reading the molecular screening scenario, it becomes clear that a trial management component needs to be available. More specifically a service needs to be provided to register and edit trials on the platform. In each such trial the end-user can generate inclusion/exclusion criteria (and demanded CRF), define trial arms, add informed consent configurations, etc.

Related to use cases: **UC.TRIALMGT.***, **UC.23**

Interface	Description
1	An administrator interacts with the front-end part (GUI based) of the trial management service. This front-end exposes following functionality: • Register a new trial in the platform • Edit a trial on the platform • Create inclusion/exclusion criteria for a trial
2	The trial management service exposes following functionality to the informed consent service: • List all the registered trials in the platform
3	The trial management service exposes following functionality to the patient screening service: • List all the registered trials in the platform • Get detailed information about a selected trial
4	The trial management service exposes following functionality to the criteria matcher: • Returns a executable inclusion/exclusion criterion

6.1.3.6 Criteria Matching Service

As part of the molecular testing scenario, the investigator should be able to verify if the available screening data for a particular patient considered for enrolment (coming from the HER and screening datawarehouses) matches the criteria for one or more selected trials present in the trial repository. The criteria matching service is responsible for this verification. It will match the criteria with the screening data and return a decision based on the result of this matching.

Related to use cases: **UC.22**, **UC.22.b**

Interface	Description
1	The criteria matching service exposes following functionality to the patient screening service: • Match a criterion defined in a trial with provided screening/patient data (coming from the datawarehouses)
2	*No interface exposed to the query execution* service
3	*No interface exposed to the trial management service*

6.1.3.7 Query Execution Service (CDM/CIM)

In the molecular testing scenario, an investigator needs to be able to receive data stored in the screening datawarehouse and the site EHR datawarehouse(s). For this the patient screening service needs a link with the semantic layer, by means of the query execution component. This layer (worked out in paragraph 6.3.3.3) will provide functionality to query the datasets of the EHR and screening datawarehouses. It abstracts the underlying data sources for the upper screening service and presents data to applications according to a single integrated data model.

Related to use cases: **UC.SEM.3, UC.SEM.4, UC.SEM.5**

Interface	Description
1	The query execution service exposes following functionality to the criteria matcher service: • Retrieval of screening data from the INTEGRATE datawarehouse(s) • Retrieval of EHR data from the site(s) datawarehouse(s)
2	The query execution service exposes following functionality to the patient screening service: • Retrieval of screening data from the INTEGRATE datawarehouse(s) • Retrieval of EHR data from the site(s) datawarehouse(s) • Storage of screening data to the CDM

6.1.3.8 Biotracking Service

Although the Biotracking system is out-of-scope for the INTEGRATE project, it is listed here for completeness. It is important that clear interfaces are defined between the biotracking and screening service in order to provide easy integration between both components. The central accredited labs will interact with this component.

Related to use cases: -

Interface	Description
1	The biotracking service exposes following functionality to the patient screening service: • Register, track and analyse biological samples of a patient

6.2 Cohort Selection

6.2.1 Introduction

The trial data querying scenario (see deliverable 1.2) demands that research should be able to generate and execute queries on the INTEGRATE datawarehouses in the research domain in order to retrieve cohorts of patients for research information. The functional concerns of this scenario are addressed in the cohort selection view. A domain specific language was developed in INTEGRATE (called SNAQL) that enables to generate scripts that allows researcher to rely only on domain knowledge. It does not require extensive technical knowledge (e.g. query languages such as SQL, SPARQL) or awareness of the underlying data models (Claerhout et al., 2013). As described before, this querying of the datawarehouses happens in collaboration with the semantic layer (see paragraph 6.3.3). Four main components were identified using

the use cases of deliverables 1.4 and 1.5. These components together with the connected semantic and analytics components are explained in the next subsections.

Concerns addressed in this view (see paragraph 4)
CUS-002, CUS-003, CDE-001

6.2.2 Diagram

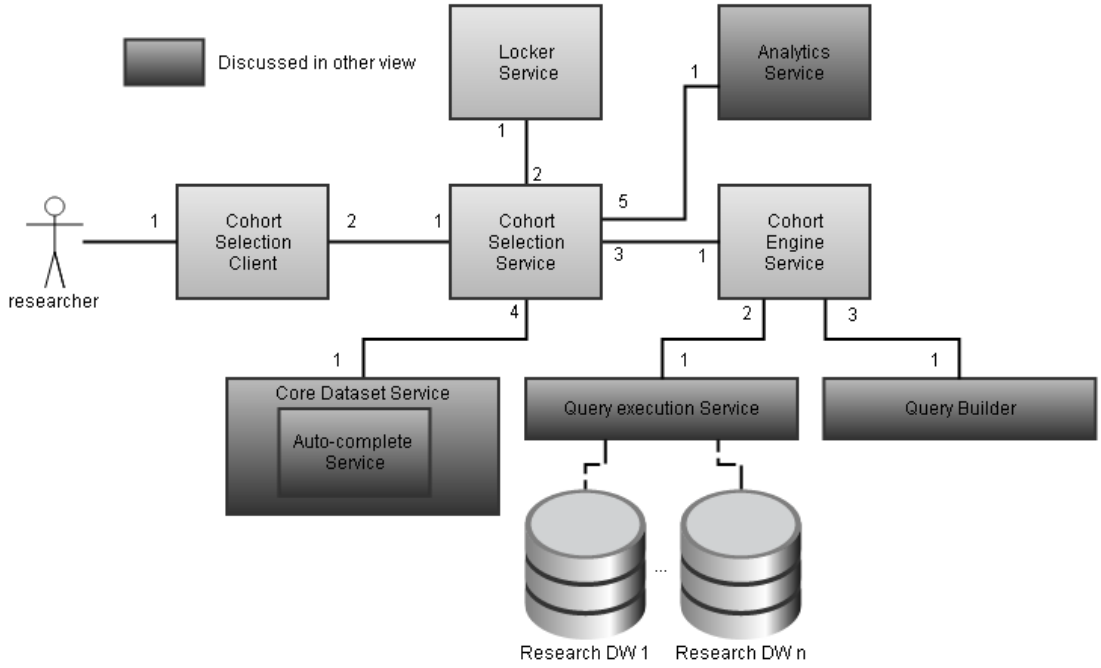


Figure 7: Cohort Selection Functional View

6.2.3 Components Interfaces

6.2.3.1 Cohort Selection Client

The application front-end of the trial data query scenario is provided by this client. It offers an (advanced) graphical user interface (GUI) to the researcher which visualises the SNAQL script building, filtering, analysing and executing functionality in an intuitive and user-friendly flexible way.

Related to use cases: **UC.TQ.1, UC.TQ.2, UC.TQ.6**

Component	Interface
1	A researcher interacts with the front-end part (GUI based) of the cohort selection client. This client exposes following functionality: • The visual part of the step by step process for creating new SNAQL scripts and the execution/analysis of these scripts
2	*No interface exposed to the cohort selection service*

6.2.3.2 Cohort Selection Service

The cohort selection service is the main driver component in the cohort selection view. It is a service that integrates and connects the different services that are needed to meet the specified requirements of the trial data querying scenario.

Related to use cases: **UC.TQ.3, UC.TQ.4, UC.TQ.5**

Interface	Description
1	The cohort selection service exposes following functionality to the cohort selection client: • The functional part of the step by step process for creating new SNAQL scripts and the execution/analysis of these scripts
2	*No interface exposed to the locker service*
3	*No interface exposed to the cohort engine service*
4	*No interface exposed to the core dataset service*
5	*No interface exposed to the analytics service*

6.2.3.3 Cohort Engine Service

The cohort engine service will execute the scripts defined in the cohort engine service. For this, it uses a query builder service for translating the scripts to queries that are specific to the underlying semantic layer. These resulting queries are sent to the query execution service which will return the resulting patient cohorts.

Related to use cases: **UC.TQ.3, UC.TQ.4, UC.TQ.5**

Component	Interface
1	The cohort engine service exposes following functionality to the cohort selection service: • Executes a given script and returns the found cohort result set
2	*No interface exposed to the query execution service*
3	*No interface exposed to the query builder service*

6.2.3.4 Locker Service

The locker service is a platform service within INTEGRATE. The locker is a generic service, facilitating storage structured documents (json) and blobs for a user. For this scenario, the locker stores amongst others:

- Available data sources
- Cohorts and cohort details (patient id's)
- Filter templates
- Analysis filters
- Worksets

*Related to use cases: **UC.TQ.1, UC.TQ.2***

Component	Interface
1	The Locker service exposes following functionality: • Create, Retrieve and Delete structured documents (json) • Create, Retrieve and Delete BLOBS(json)

6.2.3.5 Analytics Service

The analytics Service (which is worked out in paragraph 6.5.3.1) performs descriptive statistics on patient cohorts. A variety of parameters can be used (e.g. Age, tumor stage, Lymph node involvement, tumor grade, HER2 status, TOP2 status, ESR1 status, ERBB2 status and pathological complete response. In addition, survival analysis (DMFS and OS can be split out according to the previous parameters).

Related to use cases: ***UC.TQ.2***

Component	Interface
1	The analytics service exposes following functionality to the cohort selection service: • Given a patient cohort, descriptive statistics are returned

6.2.3.6 Query Builder Service

The query builder service (which is worked out in paragraph 6.3.3.1) of the semantic layer will help to translate SNAQL fragments or more specific, vocabulary concepts, to queries that can be executed in the underlying semantic layer.

Related to use cases: **UC.SEM.1**

Component	Interface
1	The query builder service exposes following functionality to the cohort engine service: • Returns a defined query template for a given concept in order to retrieve information from the CDM/CIM

6.2.3.7 Core Dataset Service

The cohort selection service uses the auto-complete component of the core dataset service (which is worked out in paragraph 6.3.3.4). This component enables the researcher to query the core dataset using a search string mechanism for finding concepts.

Related to uses cases: **UC.SEM.1**

Component	Interface
1	The core dataset service exposes following functionality to the cohort selection service: • Returns a list of matching core concepts for a given search string

6.2.3.8 Query Execution Service

The trial data querying scenario states that a researcher needs to be able to query data stored in the INTEGRATE research datawarehouses (research domain). For this the cohort engine service needs a link with the semantic layer, by means of the query execution component. This layer (which is worked out in paragraph 6.3.3.3) provides functionality to query the datasets of the INTEGRATE datawarehouses. It abstracts the underlying data sources for the upper cohort selection services and presents data to applications according to a single integrated data model.

Related to use cases: **UC.SEM.3, UC.SEM.4, UC.SEM.5**

Component	Interface
1	The query execution service exposes following functionality to the cohort engine service: • Execute queries on the INTEGRATE datawarehouses and return the matching result sets

6.3 Semantic Layer

6.3.1 Introduction

The objective of the semantic interoperability layer in INTEGRATE platform is to provide a homogenous interface to retrieve clinical data for the different components of the platform. The first process relevant for the semantic interoperability layer is the Extraction, Transformation and Load (ETL) process of the information contained in the different data sources into a data warehouse following a Common Data Model (CDM) (Schema of the Common Information Model, CIM) and using core dataset concepts (vocabulary of the CIM). Two services are therefore involved in this process: The data push service (responsible for storing information on the data warehouse) and the core dataset service.

Another objective of the semantic interoperability layer is to facilitate the composition of CIM-based queries and to store additional data introduced by the users (e.g. informed consent) among other information. The query builder service describes how concepts from the core dataset are stored, and the Query Execution Service is responsible for retrieving information from the data warehouse.

In the next sections a diagram of the semantic interoperability layer, the different services and their relations with the use cases is presented.

Concerns addressed in this view (see paragraph 4)
CUS-002, CUS-003, CDE-001

6.3.2 Diagram

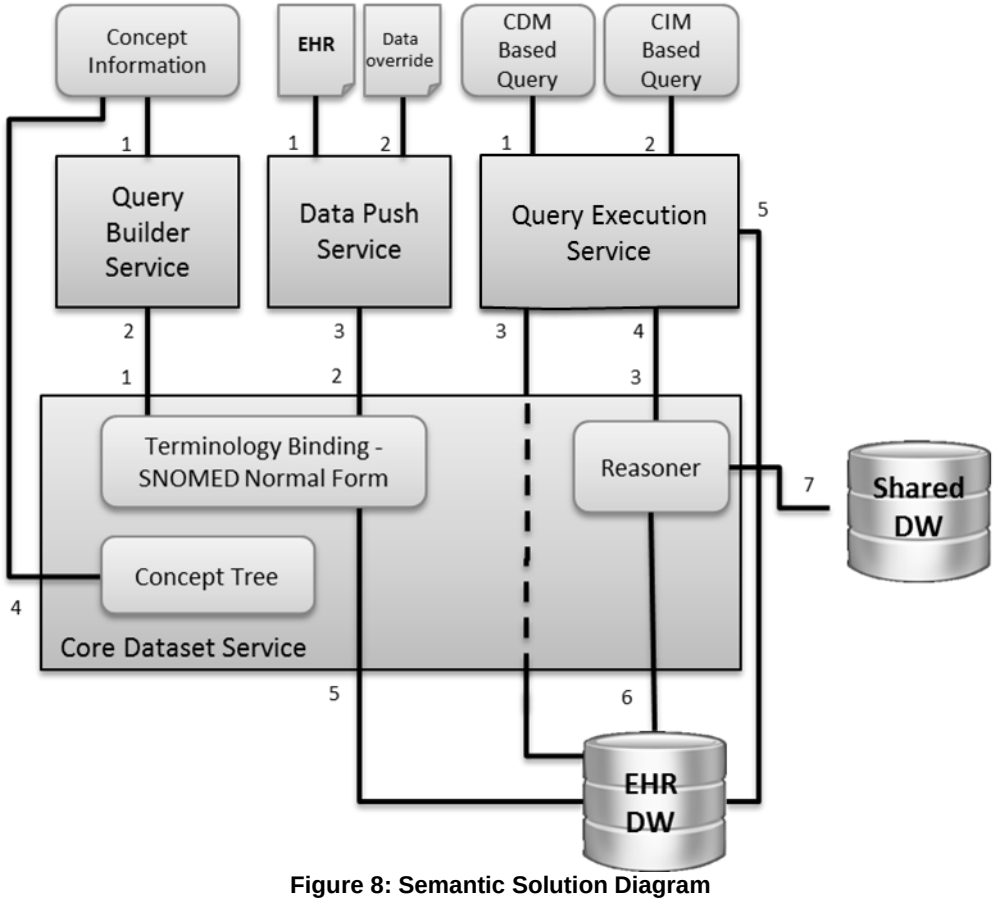


Figure 8: Semantic Solution Diagram

6.3.3 Components and Interfaces

In this section all the components of Figure 8 are described.

6.3.3.1 Query Builder Service

The query builder service is responsible for defining how to retrieve information from the CIM. This service receives vocabulary information and returns a list of SPARQL templates in XML format to facilitate building a query to retrieve the required information, where the first template is the most indicated to the given concept information. The query builder service interacts with the core dataset service to translate the given concept into the SNOMED normal form and to translate this information into classes and attributes of the CDM.

Related to uses cases: **UC.SEM.1**

Interface	Description
1	The query builder service exposes the following functionality to the requesting INTEGRATE services: • Returns defined SPARQL templates for a given SNOMED information (core dataset concepts or CDM attributes) in order to retrieve information of the CDM
2	*No interface exposed to the core dataset service.*

6.3.3.2 Data Push Service

The data push service is responsible for transforming and loading of original source data (based on HL7 v3 and v2) into the EHR Data Warehouse and the override data (such as informed consent). Three other elements are related to this service:

- Data sources
- Core dataset service: The SNOMED normal form module and the terminology binding module
- EHR Data Warehouse

Related to uses cases: **UC.SEM.2**

Interface	Description
1	The data push service exposes following functionality to the INTEGRATE services: • Store the EHR source data (based on HL7 v2 or v3) in the EHR Data Warehouse after the data is validated against an XSD schema.
2	*The data push service exposes the following functionality to the INTEGRATE services:* • Store additional data (such as informed consent) introduced by INTEGRATE users into the EHR Data Warehouse
3	*No interface exposed to the core dataset service*

6.3.3.3 Query Execution Service

The Common Data Model component stores all the data of the different sources using the same data model. CDM contains a relational database and a SPARQL wrapper to query information with the SPARQL language. The information is stored by the data push service and it is coded with concepts of the core dataset. Therefore, it is possible to retrieve the information using CIM based queries and CDM based queries.
CIM based queries are coded in SPARQL language and the queries can be expanded with core dataset knowledge by the reasoner module. Once the query is executed, the service returns the results to the Query Execution Service as SPARQL results format [REF]. CDM based queries are SPARQL queries without explicit knowledge of the Core Dataset. Once the query is executed, the Query Execution Service returns the results to a SPARQL resulset format[3].

Related to uses cases: **UC.SEM.3, UC.SEM.4, UC.SEM.5**

Interface	Description
1	The Query Execution service exposes the following functionality to the requesting INTEGRATE services: • Execute incoming CDM based queries on the EHR Data Warehouse and return the matching result sets
2	The Query Execution service exposes the following functionality to the requesting INTEGRATE services: • Execute incoming CIM based queries on the EHR Data Warehouse and return the matching result sets. These queries could be expanded in the core dataset service.

[3] http://www.w3.org/TR/rdf-sparql-XMLres/

3	*No interface exposed to the EHR Data Warehouse*
4	*No interface exposed to the core dataset service*
5	*No interface exposed to the EHR Data Warehouse*

6.3.3.4 Core Dataset Service

The core dataset component is composed of the biomedical vocabulary used in the INTEGRATE platform, the SNOMED reasoner, the concept tree, the terminology binding and the SNOMED normal form module. Core dataset is the vocabulary that standardizes the concepts used in the INTEGRATE platform, including relationships to perform semantically aware queries. A semantic reasoner is required to exploit the relationships of the core dataset by expanding SNOMED concepts. The reasoning module is used to infer semantic knowledge from biomedical vocabularies, and within the scope of INTEGRATE inference will be restricted to SNOMED (not to extensions of the core dataset with domain specific vocabularies such as MedDra, LOINC or HGNC). The concept tree is developed to facilitate the GUI navigation between the different vocabularies. The SNOMED normal form module and the terminology binding module, translates SNOMED concepts into the normal form and relates this information with the classes and attributes of the CDM.

Related to uses cases: **UC.SEM.1, UC.SEM 2, UC.SEM.3, UC.SEM.4, UC.SEM.5**

Interface	Description
1	The core dataset reasoner service provides the following functionality to the query builder service: • Translates a concept into the normal form and indicates in which classes and attributes of the Data Model this concept is stored.
2	The core dataset reasoner service provides the following functionality to the data push service: • Translates a concept into the normal form and indicates in which classes and attributes of the Data Model the concept has to be stored.
3	The core dataset reasoner service provides the following functionality to the CDM access Service: • Expands a SNOMED concept if it is needed.
4	The core dataset service provides the following functionality: • Return the SNOMED concepts that are subclasses of given concepts.
5	*No interface exposed to the EHR Data Warehouse*
6	*No interface exposed to the EHR Data Warehouse*
7	*No interface exposed to the Shared Data Warehouse*

6.4 Central Pathology Review

The Central Pathology Review platform is a one stop shop solution which offers to the stakeholders (mainly to the pathologists, but also to all the other authorized and approved stakeholders) the following tools/functionalities:

- viewing of pathology images in a way similar to a virtual microscope,

- reviewing and annotation of pathology images, by highlighting regions of interest (ROIs), by adding detailed comments (for the selected ROIs) and by filling detailed reports which are dynamically generated and finally
- allowing for collaboration between reviewers (providing sharing, commenting and resolving functionalities)

Concerns addressed in this view (see paragraph 4)
CUS-001 *(1)*, **COP-001** *(2)*, **CDE-001** *(1) Requirements concerning the reviewers* *(2) In this view the INTEGRATE Central Review Service provides administrator oriented interfaces.*

6.4.1 Diagram

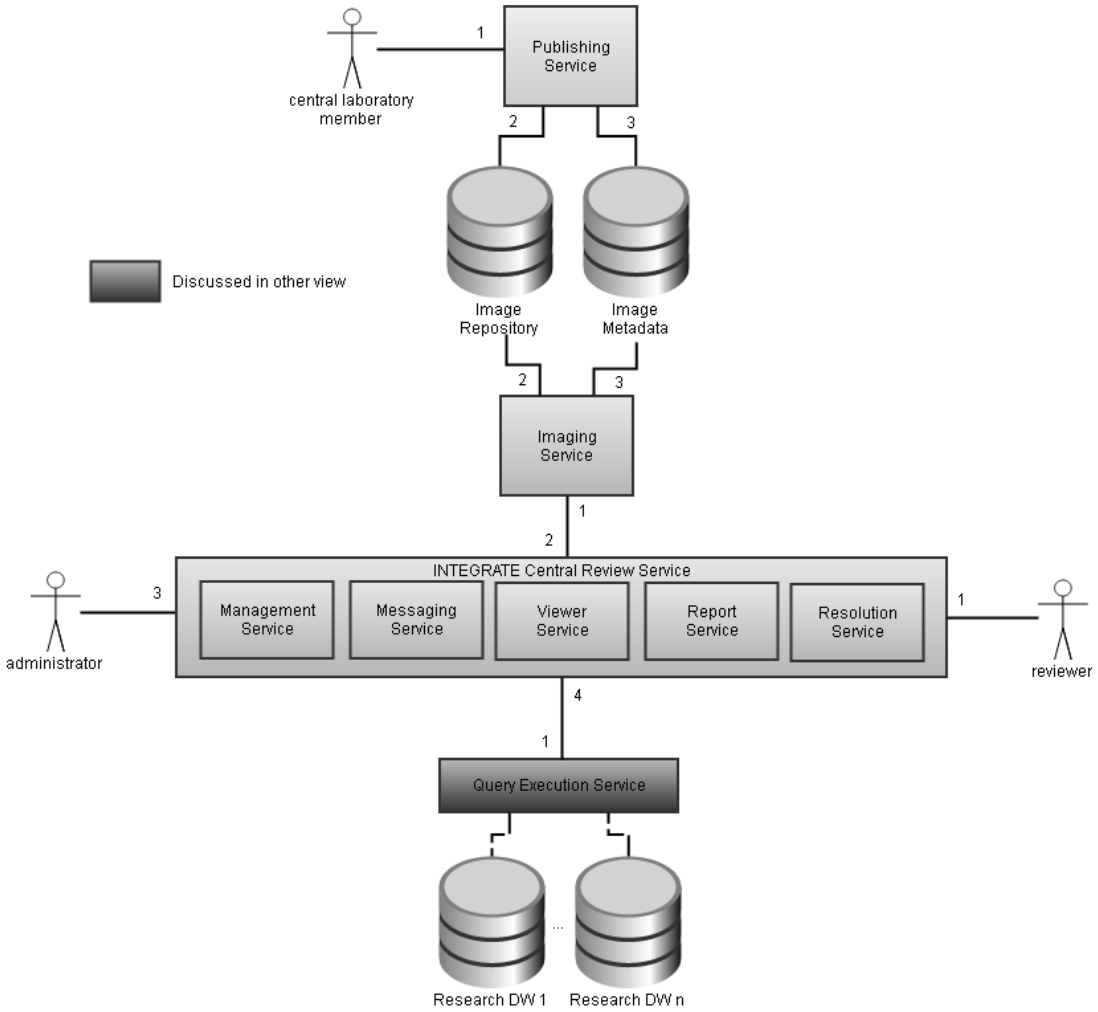


Figure 9: Central Pathology Review

6.4.2 Components and Interfaces

6.4.2.1 Publishing Service

This component is a front-end GUI with directions on how a central laboratory member can upload and store a new medical image to the images repository as also the actual interface for image upload.

NOTE**:** *The publishing service is not a part of the architecture of the central review platform, but its existence and implementation is necessary in order to upload the images in the CPR internal infrastructure. The publishing service is described in another scenario (see deliverable 1.2) -and not in the central review scenario- and therefore will be defined and implemented by the architecture defined there. The separation has been made for reasons of better understanding.*

Related use cases: -

Interface	Description
1	The administrator interacts with the front end part (GUI based) of the publishing service. This front end exposes following functionality: • Detailed directions in a user-friendly form for uploading a new medical image. • Interface for uploading (secure FTP) raw pathology images to the imaging service infrastructure.
2	*No interface exposed to the image repository*
3	*No interface exposed to the image metadata*

6.4.2.2 Imaging Service

Imaging Service is a standalone component of the Central Pathology Review platform responsible for polling the *Imaging Server* for new (uploaded) Images, slicing them into tiles, extracting metadata from the images and pushing these metadata to the data warehouse. Subsequently, the INTEGRATE central review and collaboration platform should be able to get access to the images and the relevant meta-information deployed in the scenario of the central review. Therefore, the imaging service is the only component responsible for processing raw pathology images, storing them into the INTEGRATE infrastructure and requesting an image (tiles) and its meta-information for displaying into the CPR platform.

Related use cases: **UC.CR.3, UC.CR.4, UC.CR.5, UC.CR.6, UC.CR.7**

Interface	Description
1	The imaging service exposes the following functionality to the INTEGRATE central review service: • Providing access to the images (tiles) in the image repository (read only) via a secure URL (CPR infrastructure) • Providing access to the metadata in the image meta-data repository via a secure SOAP service – (Data warehouse)
2	*No interface exposed to the image repository*
3	*No interface exposed to the image metadata*

6.4.2.3 Management Service

The management service provides all the necessary functionality and tools in order to create collaboration groups, define new protocol (image) types, schedule new review protocols and any other functionality which is oriented to management.

Related use cases: **UC.CR.1, UC.CR.2, UC.CR.3, UC.CR.4, UC.CR.6, UC.CR.7**

Interface	Description

1	The management service exposes to the *reviewers* a GUI from which they can:
	• View a listing of images (thumbnail view) which are pending to be reviewed as also their corresponding protocol tasks.
	  ○ View tasks details & statuses
	• View a listing of user's tasks per image and per protocol.
	  ○ View tasks details & statuses
	• Edit pending or acquired tasks and submit reviewer's answer back to the platform
	• View internal notifications

2	*No interface exposed to the* to *the Imaging Service*
3	The management service exposes to the *administrators* a GUI from which they can: • Enter CPR platform configuration details, such as Imaging Server parameters, e-Mail configuration settings, secure FTP settings etc. • Create/edit a collaboration group • Synchronize/Import available images from the data warehouse to the CPR platform. In this process an image type is associated to each image. • Define a new protocol type (Image Type). Define its collections of variables and respectively its variables. • Create new collection of variables. • Create new variable (available review form fields) of particular type. Associate this variable to a particular collection. • Schedule new Review Protocols by a) defining Collections of Variables to be reviewed, b) selecting the images to be reviewed, c) selecting reviewers to participate in the protocol and finally, d) assign a title and a description (HTML) to the protocol. • Administer existing review protocols by checking their tasks (per protocol/per image) statuses and respectively reviewers' answers in an aggregated view. • Administrator can change tasks statuses from Open to Close or Conflict, forcing review process workflow to proceed in its lifecycle. • Finally if all tasks of a specific review protocol are marked as closed, administrator can close the entire review protocol. • Administrator can also delete (or archive) completed protocols from the platform. • View a listing of all images synchronized into the platform in thumbnail view. View review protocols' tasks per image.

	• View incoming internal notifications..
4	*No interface exposed to the query execution service*

6.4.2.4 Messaging Service

The messaging service provides the INTEGRATE central review platform with messaging and notifications functionality.

Related use cases: **UC.CR.6**

Interface	Description
1	The messaging service provides the reviewers with all the tools and mechanisms needed to receive email messages and internal notifications in order to participate in a review process and resolve a case of a disagreement while reviewing an image.
2	*No interface exposed to the imaging service*
3	The messaging service provides notifications to the administrators regarding: • The status of the images which are registered in an active review process or • Requests from users or • Issues and errors
4	*No interface exposed to the query execution service*

6.4.2.5 Viewer Service

The viewer service provides a GUI to the users and enables them to view pathology images (and in the future could probably be expanded in order to display DICOM images) and annotate them.

Related use cases: **UC.CR.3, UC.CR.4, UC.CR.5, UC.CR.6**

Interface	Description
1	The viewer service provides to the reviewers a graphical user interface from which they can view and annotate images stored in the image repository.
2	*No interface exposed to the imaging service*
3	The viewer service provides to the administrators a graphical user interface from which they can view images stored in the image repository.
4	*No interface exposed to the query execution service*

6.4.2.6 Report Service

The report service is a simple GUI that enables the pathologists to fill in the required report for the pathology review and to store the data in the appropriate repository. Reports' structure (review forms) are generated automatically by Central Pathology Review platform based on image's type as also on the collections of variables selected during review protocol registration process.

Related use cases: **UC.CR.5, UC.CR.6, UC.CR.7**

Interface	Description
1	• The report service is the mandatory report form which the reviewers are filling in every review, as a web form. The reviewers can either fill in, or just view the reports. • The *report forms* are generated dynamically by CPR platform during review protocol registration process. Each protocol task's report form depends on both image type and the collections that selected during review protocol registration process.
2	*No interface exposed to the imaging service*
3	The administrators can see the reports which are filled by the reviewers.
4	*No interface exposed to the query execution service*

6.4.2.7 Resolution Service

The resolution service provides to the INTEGRATE central review service the capability to check the images under review, if there is a disagreement among the reviewers it provides the means to resolve it.

Related use cases: **UC.CR.6**

Interface	Description
1	The resolution service is responsible for checking the content of the images/tasks which are under review (the annotations and the data of the corresponding report). The resolution service exposes to the reviewers the following functionality: • It informs reviewers for conflicts in their answers regarding specific protocol (image) tasks. Their status is marked as conflicting. • A reviewer must review his/her answer and re-submit the findings to the platform.
2	*No interface exposed to the imaging service*
3	The resolution service exposes to the administrator a GUI from which the administrator can: • View the status of the images' tasks being reviewed (how it is characterized by each reviewer) • It merges information from all reviews in a common portal page, side by side, in such a way that the moderators can easily compare reviewers' answers per variable and per task. • See an overview regarding the image, which merges the information from all the reviews in a simple and common page. • If the moderator is not been able to find any inconsistent findings, then he/she marks the task as "Closed". If all protocol tasks for this specific image are "Closed" by the moderator, the resolution service sets the corresponding status flag and stores it to the database. • If there is a disagreement between the reviewer's answers for each protocol image moderator marks the measurement/value which is in question and automatically the image is marked to be "For further investigation" (task status

	flag is set to “Conflict”). In this situation where there is not an agreement between all the reviewers, the resolution process tries to address the issue using an internal message exchange mechanism which starts a “conversation” among the reviewers until they reach to an agreement.
4	*No interface exposed to the query execution service*

6.4.2.8 Query Execution Service

The query execution service (see 6.3.3.3) enables bidirectional access to the INTEGRATE data repository and to the INTEGRATE metadata repository. The central review platform uses the functionality provided by the component in order to retrieve or store data to the appropriate repositories.

Related use cases: **UC.CR.5, UC.CR.6, UC.CR.7**

Interface	Description
1	The query execution service exposes to the INTEGRATE central review service the functionality to access, download and upload data to the data and meta-data repositories of INTEGRATE. All the communication between CPR and meta-data repositories is secure (secured SOAP requests). In more detail: • CPR access data warehouse for storing images Meta data information. • CPR access data warehouse for retrieving the relationship between patients and their images as also for synchronizing images for review. • CPR access data warehouse for storing reviewer’s answers.

6.5 Analysis Platform

6.5.1 Introduction

The INTEGRATE analysis platform is the main end-user platform in which a researcher can access a pool of available tools and models for the analysis of patient’s data in a user-friendly manner. The framework provides the researcher a list of tools and models in order to process clinic-genomic data stored in the central INTEGRATE data-warehouse. Further it exposes functionality for connecting to the tools & models repository and the tools & model meta-data repository.

The analysis is mainly divided into two categories; the tools for the statistical analysis and the models for prediction analysis. The analytical tools component is responsible for the implementation of the statistical analysis. The predictive modelling tools are the intermediate connection between the researcher and the predictive models. Depending on the nature of the selected data, tools and models address specific research questions (see D.1.2: Research queries on completed trial data and D.5.1).

Concerns addressed in this view (see paragraph 4)
CUS-002, CUS-003, CDE-001, CIN-003, CDP-002

6.5.2 Diagram

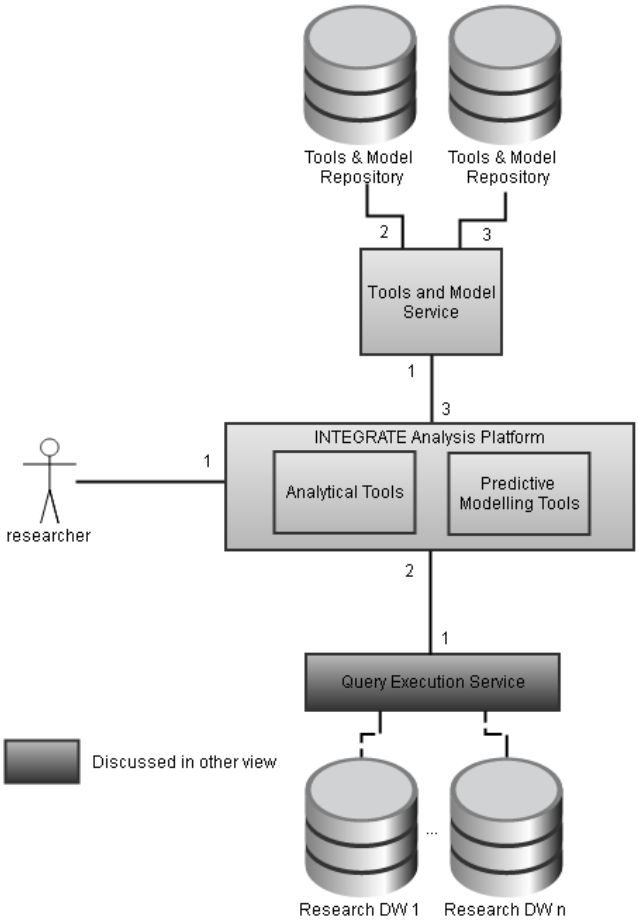


Figure 10: Analytical Tools

6.5.3 Components and Interfaces

6.5.3.1 Analytical Tools

The analytical tools communicate via query execution service with the INTEGRATE common data model, the tools & model repository and the tools & model meta-data repository for the selection, retrieval and management of the data, tools and models. Moreover, it provides a user-friendly framework for the visualisation, download and storage of a statistical analysis report.

Related to use cases: **UC.IAT.***, **UC.IAT_PM.***

Interface	Description
1	The researcher interacts with the analytical tools (a GUI based front-end) of the INTEGRATE analysis platform. This front-end exposes following step-by-step functionality: • A listed menu of several statistical tools which address a number of research questions (see D.1.2, Research queries on completed trial data).

	• The system retrieves the data queried for analysis and creates a new result set to feed the selected statistical tool (e.g. a specific number of clinical data items, etc.) • An interface for the visualisation of an analysis report. • All the analysis reports are available for download via the platform.
2	*No interface exposed to the query execution service*
3	*No interface exposed to the tools & model service*

6.5.3.2 Predictive modelling Tools

As a part of the INTEGRATE analysis platform, the predictive modelling tools has almost the same structure as the analytical tools.

Related use cases: **UC.PM.***, **UC.IAT_PM.***

Interface	Description
1	The researcher interacts with the sharing of predictive models (a GUI based front-end) of the INTEGRATE analysis platform. This front-end exposes following step-by-step functionality: • A listed menu of prediction models which address a number of research questions (see D.1.2, Research queries on completed trial data). • The system retrieves the data queried for analysis and creates a new result set to feed the selected predictive model (e.g. genomic AND clinical data of all patients enrolled in a trial). • An interface for the visualisation of an analysis report. • All the analysis reports are available for download via the platform.
2	*No interface exposed to the query execution service*
3	*No interface exposed to the tools & model service*

6.5.3.3 Query Execution Service

The query execution service (see 6.3.3.3) provides functionality for querying data from the INTEGRATE common data model. Also interacts with the INTEGRATE analysis platform providing the desired data for statistical analysis or predictive modeling.

Related use cases: **UC.IAT.2**, **UC.PM.2**

Interface	Description
1	The query execution service exposes following functionality to the INTEGRATE analysis platform: • In case the analytical tools is requesting: executing a given query and returning the final result set for analysis. • In case the sharing of predictive models is requesting: executing a given query and returning the final result set(s) for the prediction analysis.

6.5.3.4 Tools and Model Service

The INTEGRATE analysis platform, on behalf of the analytical tools and the predictive modelling tools, should be able to access all the available tools and models which are deployed for addressing a number of research questions. The tools and model service is the only component responsible for requesting a tool or model to run, it connects the data with the tools or models and returns the analysis report to the main analysis platform.

Related use cases: **UC.IAT.1**, **UC.PM.1**

Interface	Description
1	The tools and model service exposes following functionality to the INTEGRATE analysis platform: • Sending the results and the metadata information related to the analysis deployed.
2	*No interface exposed to the tools & model repository*
3	*No interface exposed to the tools & model meta-data repository*

7 Information View

Concerns addressed in this view (see paragraph 4)
CUS-001 *(1)*, **CUS-004, CUS-005, CUS-006** *(1) Requirements concerning the reviewers*

7.1 Trial Meta-data Model

7.1.1 Introduction

In this chapter the content of the meta-data repositories will be specified. Input for this chapter comprises deliverables D1.2, D1.4/1.5(Use Cases) and the preliminary modelling performed in D4.2[4]. In specifying the meta-data models, we aim at leveraging BRIDG[5] as much as possible.

The biomedical research integrated domain group (BRIDG) is a collaborative effort engaging stakeholders from the clinical data interchange standards consortium (CDISC), the HL7 regulated clinical research information management technical committee (RCRIM TC), the national cancer institute (NCI) and its cancer biomedical informatics grid (caBIG®) and the US food and drug administration (FDA). The BRIDG model is an instance of a domain analysis model (DAM). The goal of the BRIDG model is to produce a shared view of the dynamic and static semantics for the domain of protocol-driven research and its associated regulatory artefacts. This domain of interest is further defined as: protocol-driven research and its associated regulatory artefacts: i.e. the data, organization, resources, rules and processes involved in the formal assessment of the utility, impact, or other pharmacological, physiological, or psychological effects of a drug, procedure, process, or device on a human, animal, or other subject or substance plus all associated regulatory artefacts required for or derived from this effort, including data specifically associated with post-marketing adverse event reporting.

Leveraging BRIDG serves multiple purposes. It ensures that the needs of a broad clinical audience are covered and aids interoperability with the relevant clinical trial standards (from CDISC[6]) and clinical practice standards (such as HL7v3[7]).

7.1.2 Information data models

For modelling the metadata, we leverage BRIDG by reusing classes and relationships (from version 3.2). This is indicated with the <<BRIDG>> stereotype in the UML diagrams. An INTEGRATE specific construct is introduced when no appropriate BRIDG construct can be found. The BRIDG definitions are used for the BRIDG classes in the class descriptions section.

[4] INTEGRATE Deliverable 4.2 - Detailed specification of the collaboration and data sharing tools
[5] Website: http://bridgmodel.org/
[6] Website: http://www.cdisc.org/
[7] Website: http://www.hl7.org/

The definition of the classes can be found in *Appendix A – trial metadata class definitions.*

7.1.2.1 Static Data Structure Model - Molecular Testing

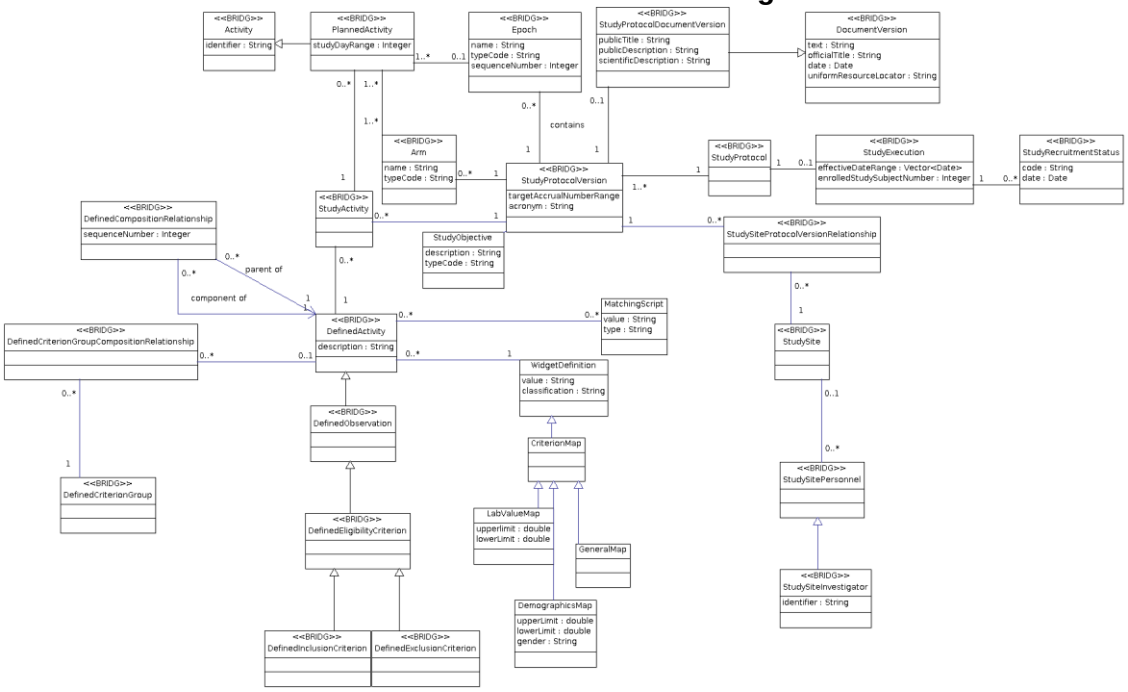


Figure 11 - static data structure model for Molecular testing

Figure 11 shows the classes relevant for the molecular testing use cases, relating to *UC 1 – Patient trial screening*. In the subsequent section we explain parts of the model and how they relate to the use case. In Appendix B, a more detailed picture can be found.

The main entry point is a versioned study protocol (StudyProtocolVersion). An investigator (named StudyStudyInvestigator in the model) can only access trials running in his hospital. The model captures this by relating a StudySiteInvestigator to a StudySite which on its turn is related to a StudyProtocolVersion.

The StudyProtocolVersion contains the bounds of the amount of study subjects that are to be accrued for the study. The related StudyExecution class specifies the (planned) period of time over which the study will execute. In addition, the number of enrolled study subjects can be obtained from this class. The StudyRecruitmentStatus captures the recruitment status of the trial, indicating whether a trial is open for enrolment.

A textual description of the study can be obtained using the StudyProtocolDocumentVersion class.

The final constructs relate to the criteria in the study. These are captured in the DefinedInclusionCriterion and DefinedExclusionCriterion classes and derive ultimately from DefinedActivity (which contains a textual description of the activity). The DefinedActivity is linked to MatchingScripts, these matching scripts contain the logic

required to retrieve and evaluate the required data items from a clinical datawarehouse using the common datamodel. In addition, the model specifies WidgetDefinitions to aid an application in visualizing the DefinedActivity's.

7.1.2.2 Static Data Structure Model - Trial Data querying

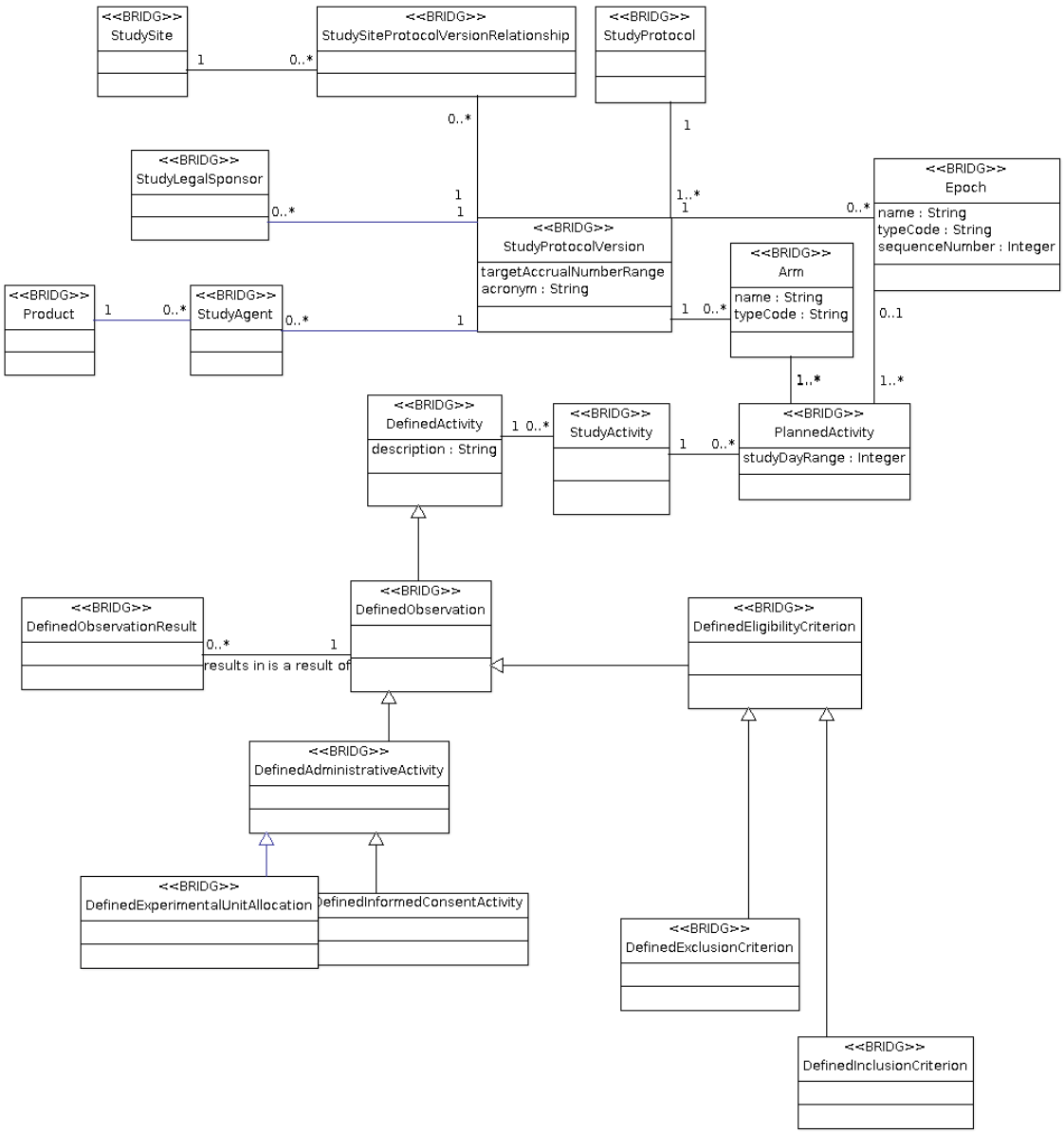


Figure 12 - static data structure model for Trial Data querying

The UML diagram depicted in Figure 12 shows the classes introduced to facilitate querying trial data. The actual trial data (stored in the Clinical DataWarehouse according the Common Information Model) is linked to a PlannedActivity. The Trial and Epoch can be found out using the PlannedActivity.

The patient data will contain an enrolment observation to specify the arm that a patient enrolled into. This enrolment observation will be linked to a PlannedActivity related to a DefinedExperimentalUnitAllocation.

This data structure allows finding out what study agents are used, who the legal sponsors are, and which study sites are participating.

7.1.2.3 Static Data Structure Model - Trial Management

In INTEGRATE, it is not yet defined who (in a clinical research workflow) can manage the information in the trial metadata repository. For now, an administrative interface will be made available.

7.1.3 Data flow

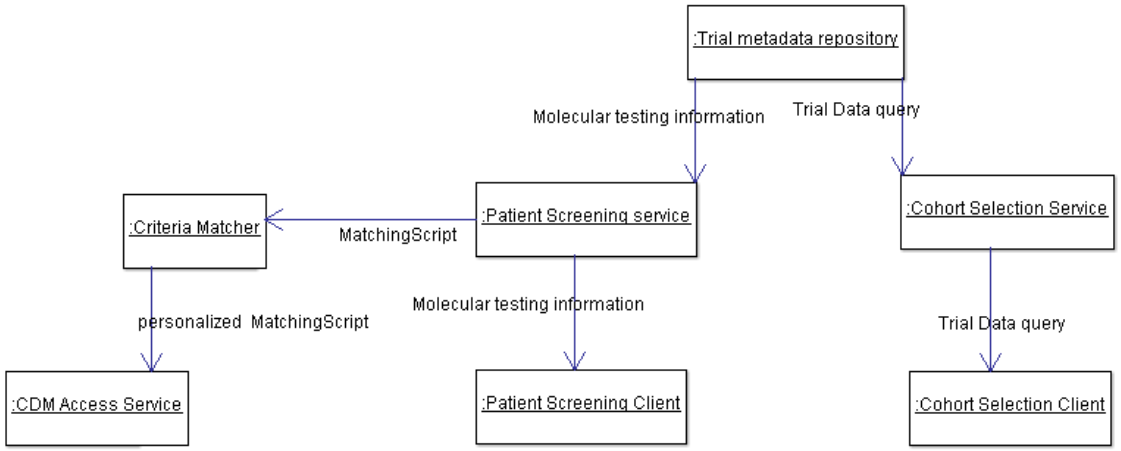


Figure 13 - collaboration diagram

Figure 13 shows the envisioned data flow for the trial metadata repository. The actual content of the data can be found in the static data structure models. The aim of the trial metadata repository is to be an “open” repository in the platform, collecting all the metadata related to clinical trials. It is envisioned that any service can query the repository (on behalf of an authenticated and authorized user).

7.1.4 Lifecycle models

Trial metadata will be created when the trial has been defined and agreed upon. When the information regarding a trial needs updating, a new StudyProtocolVersion will be created.

The screening components will see active use while patients are being enrolled into trials. Trial data querying will be performed only after a trial has been concluded and the data is available for analysis. It is not foreseen that information stored in the trial metadata repository will be removed.

7.2 Semantic Layer

7.2.1 Introduction

From the information viewpoint, it is necessary to describe data structures and flow models of processes related to the semantic interoperability layer. In the next sections, a general overview of such transfer of information is presented together with the different data structures of the semantic interoperability layer and the flow model of the information entities.

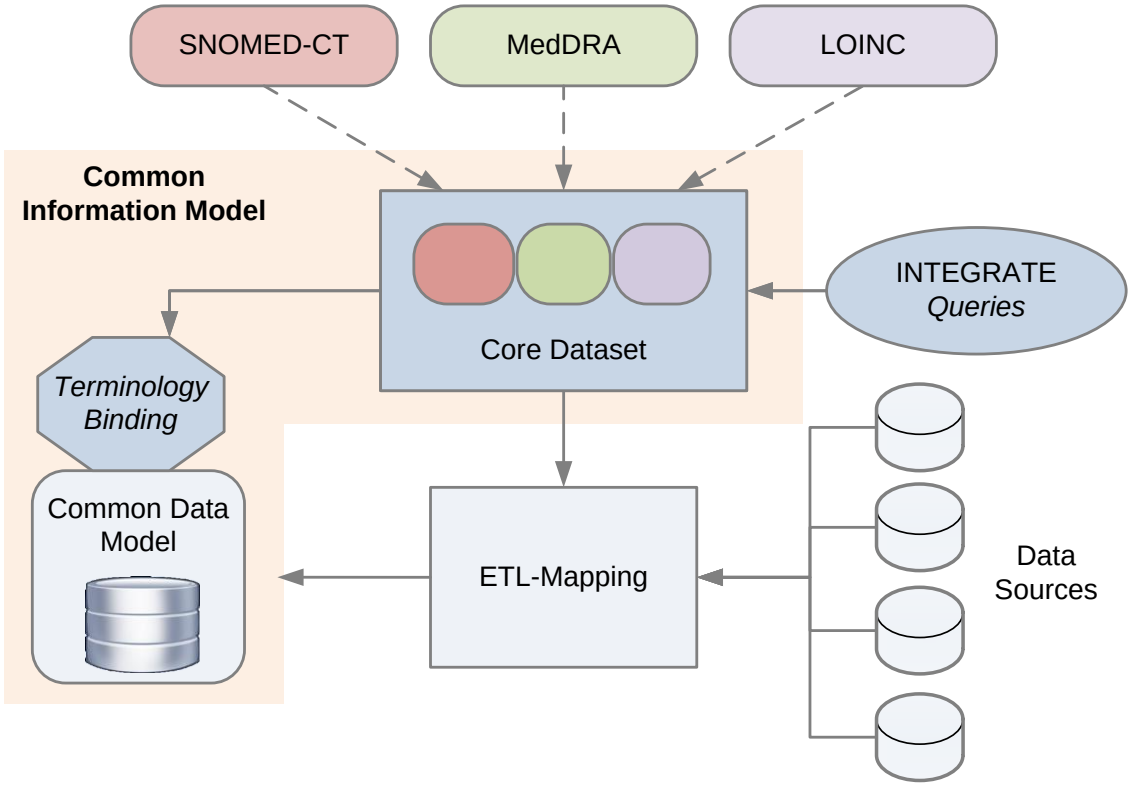


Figure 14: Semantic layer components diagram

7.2.2 Static Data Structure Model

In this section, the different data structures used in the model are described related to the corresponding use cases. It is focused on the common information model, specifically on the core dataset and the CDM.

7.2.2.1 Core Dataset

The core dataset component is the common vocabulary of the INTEGRATE platform. For that purpose, it is necessary to use a taxonomy that represents knowledge with a set of concepts and relationships between those concepts. In semantic web, ontologies are such vocabularies, used to represent knowledge about a specific domain. The following figures (Figure 15 and Figure 16) show an example of the visualization of a biomedical vocabulary (SNOMED-CT) and how it is represented in OWL (Ontology Web Language).

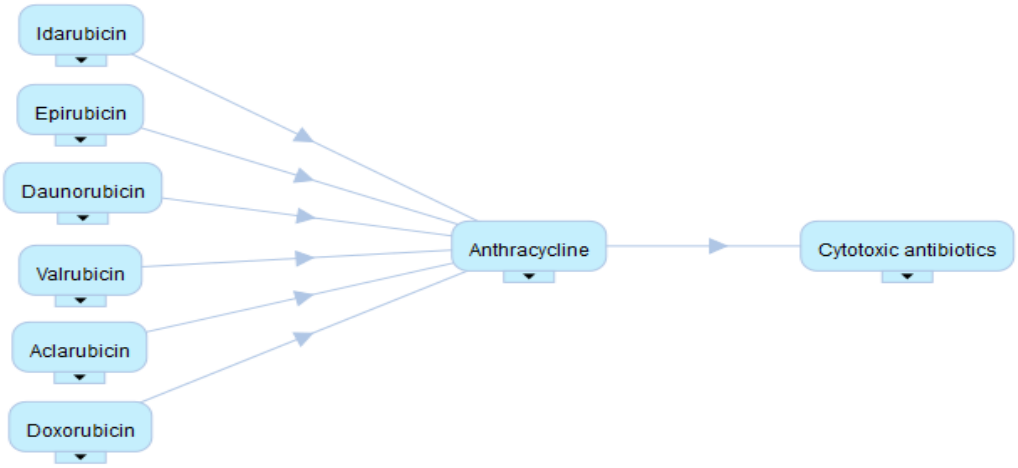


Figure 15: Example of SNOMED-CT visualization

```
<Class rdf:about="http://www.ihtsdo.org/SCT_10000006">
    <rdfs:subClassOf rdf:resource="http://www.ihtsdo.org/SCT_29857009"/>
    <rdfs:subClassOf rdf:resource="http://www.ihtsdo.org/SCT_9972008"/>
</Class>
```

Figure 16: Example of SNOMED-CT representation on OWL

The Core Dataset therefore consists of concepts from biomedical taxonomies, such as SNOMED CT, used within the CDM. Additional concepts, from other vocabularies or, inferred from the domain ontology are also included.

7.2.2.2 Common Data Model (CDM)

The CDM is the common data warehouse for the semantic interoperability layer, the homogeneous storage of patient-based information from the different sources. Therefore, the CDM receives the query built with core dataset concepts and temporal restrictions. The query is then executed to retrieve data from the CDM, and core dataset concepts within the query are used to introduce reasoning over the corresponding core dataset relationships. Finally, semantically interoperable information is returned to the user by the semantic interoperability layer.

7.2.2.3 Health Level 7 Reference Information Model

For this purpose, the CDM is based on the HL7 v3 Reference Information Model (RIM) standard[8], a solution which allows achieving the goal of keeping all the data stored homogeneously.

[8] Deliverable: 3.1 Canonical models of CTMS and HER systems

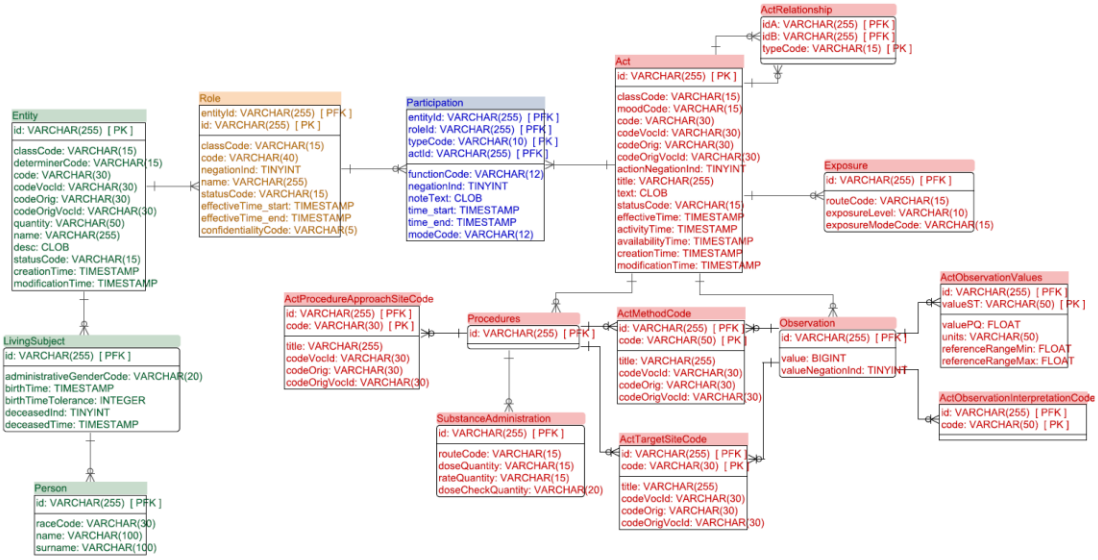


Figure 17: CDM structure based on HL7 RIM

As has been mentioned, the CDM is based on the RIM and only the relevant classes for the project are modeled in the CDM as the scope of the complete standard aims to cover any situation related to the health care environment. This model is compounded of mainly of four classes: Entity, Role, Participation and Act. The model allows for specialization of the four main classes, for example, LivingSubject as an Entity specialization, and Person as a LivingSubject specialization. These main classes are related through other classes. The classes which function is to relate the main ones are: ActRelationship (which link two different instances of Act), and RoleLink (which relate two different instances of Role). In the Figure 17 it can be observed how the CDM structure has been modeled following the standard HL7 RIM. In Appendix C, a more detailed picture can be found.

7.2.3 Information Flow Model

This section requires to describe four different *"information entities"*. These entities are related to new data sources, CIM based query, SPARQL template of the SNOMED concept and the shared data introduced by the users.

7.2.3.1 Data Source Information

The first entity is defined by the information contained in the data source. This information would be stored in the CDM.

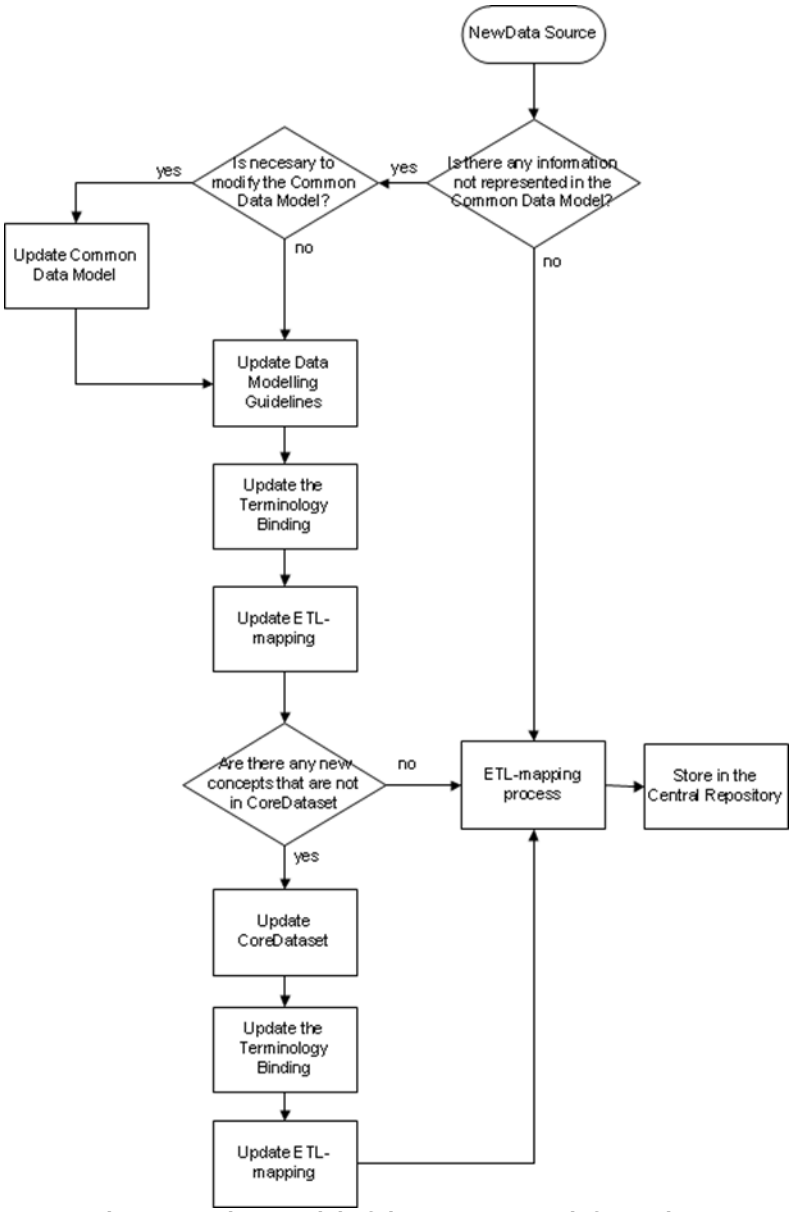


Figure 18: Flow Model of the Data Source information

As shown in Figure 18, the entity starts data source that will be stored on CDM. This data has to be in HL7 v3 or v2 format. In next steps, if the new information is not represented in the model, it could be necessary to extend the common data model and the other components. If the new information is represented in the model, then ETL mapping extracts the desired information from the data source and transforms and loads it into the CDM.

7.2.3.2 CIM and CDM Based Query Information

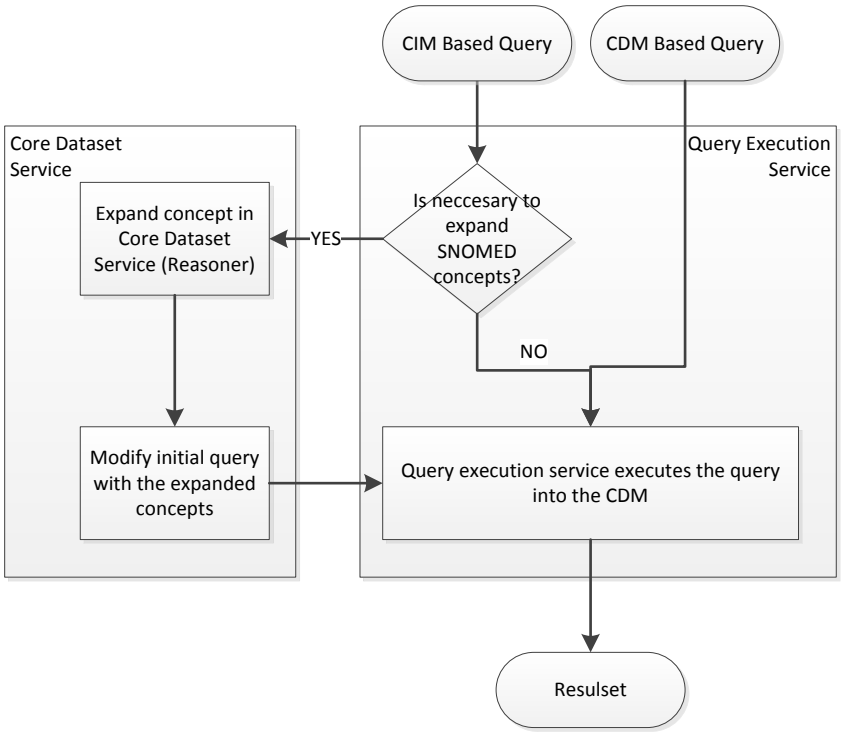


Figure 19: Flow Model of the CIM and CDM Based Query information

The CIM and CDM based query information is defined by the query sent to the semantic interoperability layer of the platform. These queries could be based on CDM or CIM components.
CIM based query, after required modifications, is sent to the CDM. As shown in Figure 19, the entity starts with the query launched to the platform. If is necessary to expand the SNOMED concepts that appear in the initial query, these will be expanded by the core dataset service and the query will be modified with the corresponding concepts. Finally, the modified query is executed including temporal restrictions defined by the original query. CDM based query is sent directly to the CDM.

7.2.3.3 Query template information

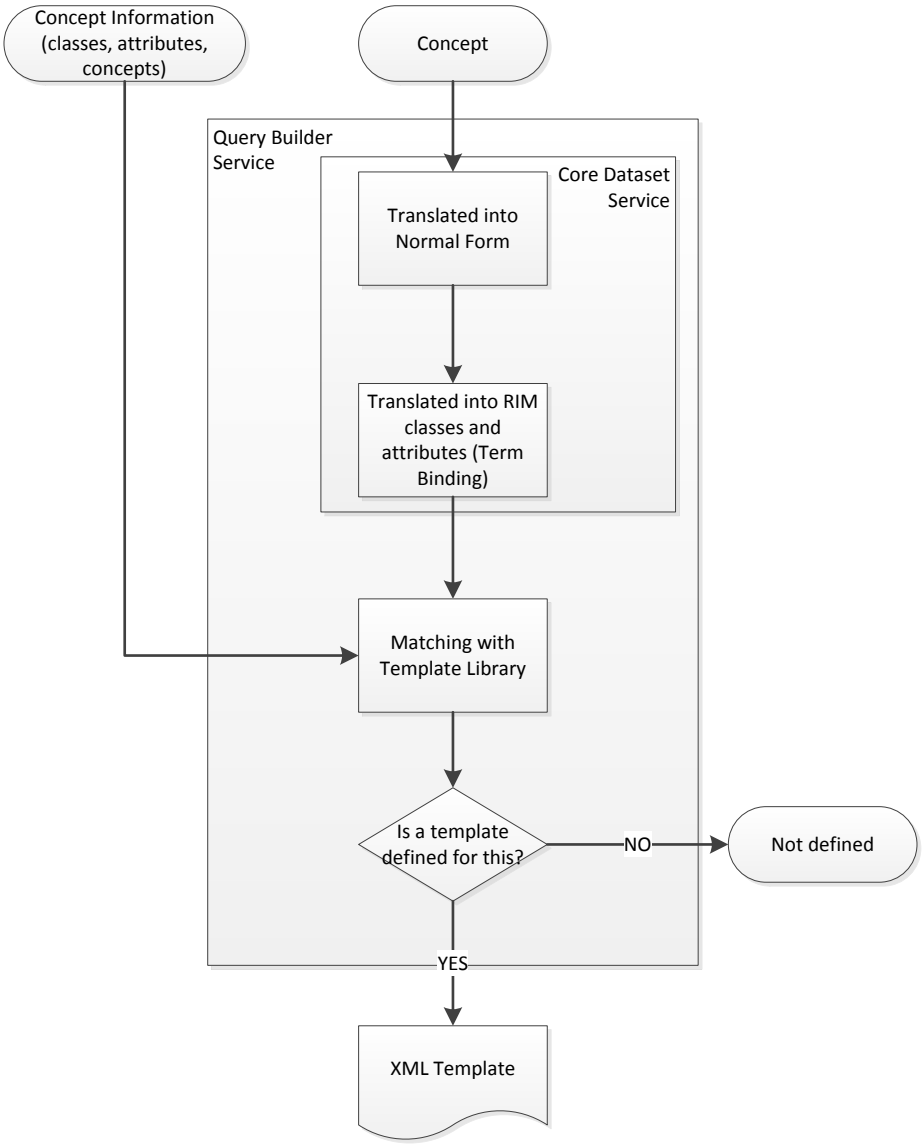


Figure 20: Flow Model of the query template information

The query template defined for a given core dataset concept is the information defined by the concept or the core dataset information (concepts, HL7 classes and HL7 attributes) sent to the semantic layer and the resulting query template. This template contains all the information needed to build a query and retrieve information from the Data Warehouse and where is located in the CDM.
As shown in Figure 20, the entity starts with the SNOMED concept given to the query builder service. This concept has to be translated into the SNOMED normal form. After this, terminology binding indicates which classes and attributes of the CDM correspond to the normal form. With this information, the query builder searches the corresponding template and returns the XML template. Other input could be the concept information. In that case, there is a matching between the templates and the input information to search the corresponding template.

7.2.3.4 Shared Data Information

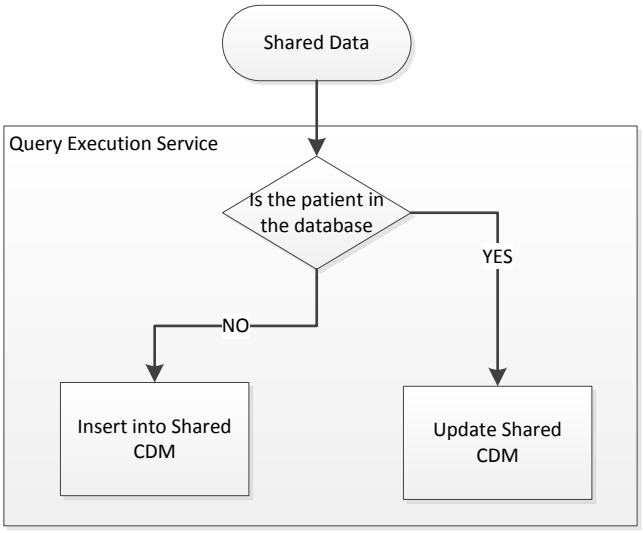


Figure 21: Flow Model of the Shared Data information

The shared data entity is defined by the information introduced by the user to the screening service. This service stores this information in the CDM.

8 Deployment View

8.1 Introduction

The deployment view gives the definition of the physical environment where each of the system components will run. The concerns regarding the technical/system requirements and dependencies for this view are extracted from the scenario document (deliverable D1.2). It needs to be noted that the deployments given in this view are possible platform instances of INTEGRATE framework and do not exclude other configurations of platform instances (see paragraph 5).

Concerns addressed in this view (see paragraph 4)
CUS-001 *(1)*, **CUS-004, CUS-005, CUS-006, CIN-001, CIN-002, CIN-003** *(1) Requirements concerning the reviewers*

8.2 Patient Screening

8.2.1 Diagram

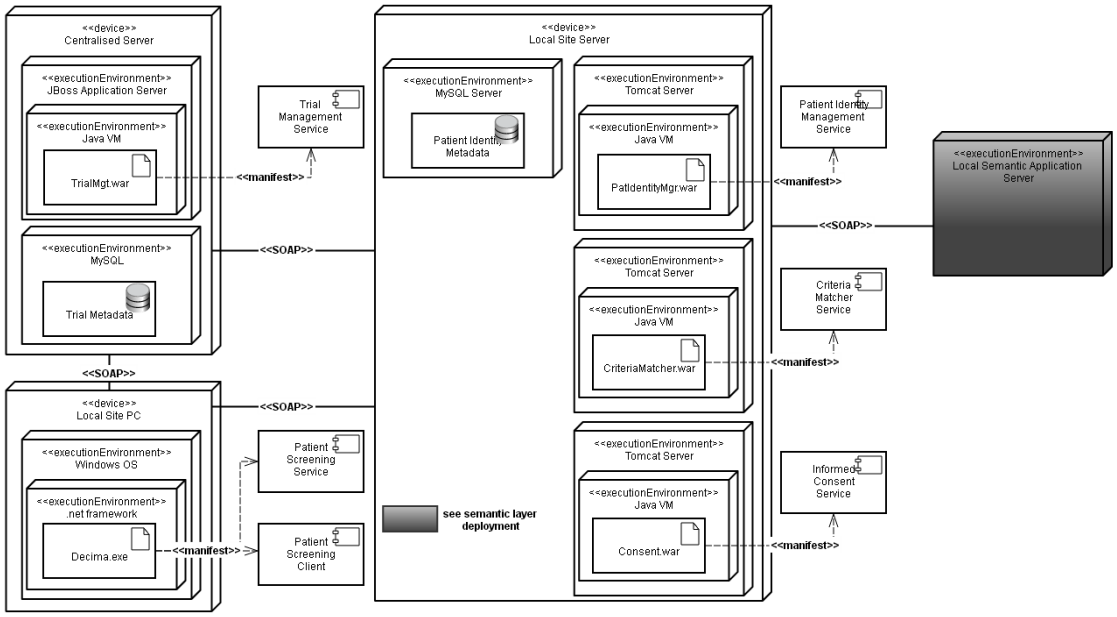


Figure 22: Patient Screening Deployment Diagram

8.2.2 Nodes

8.2.2.1 Centralised Server

Artifact	Component	Environment	Application Server	OS	Requirements
TrialMgt.war	Trial Management Service	Java VM version 1.6 or higher	Jboss	N/A	• 1 CPU Core • 1 GB RAM • 6 GB MEM
Trial Metadata	Trial Management Service	MySQL	N/A	N/A	• 1 CPU Core • 1 GB RAM • 6 GB MEM

8.2.2.2 Local Site PC

Artifact	Component	Environment	Application Server	OS	Requirements
Decima.exe	• Patient Screening Service • Patient Screening Client	.NET Framework	N/A	Windows 7 or higher	• 4 GB RAM • 3 GB MEM

8.2.2.3 Local Site Server

Artifact	Component	Environment	Application Server	OS	Requirements
PatIdentityMgr.war	Patient Identity Management Service	Java VM version 1.6 or higher	Tomcat	N/A	• 1 CPU Core • 1 GB RAM • 6 GB MEM
Patient Identity Metadata	Patient Identity Management Service	MySQL	N/A	N/A	• 1 CPU Core • 1 GB RAM • 6 GB MEM
CriteriaMatcher.war	Criteria Matcher Service	Java VM version 1.6 or higher	Tomcat	N/A	• 1 CPU Core • 1 GB RAM • 6 GB MEM
Consent.war	Informed Consent Service	Java VM version 1.6 or higher	Tomcat	N/A	• 1 CPU Core • 1 GB RAM • 6 GB MEM

8.3 Cohort Selection

8.3.1 Diagram

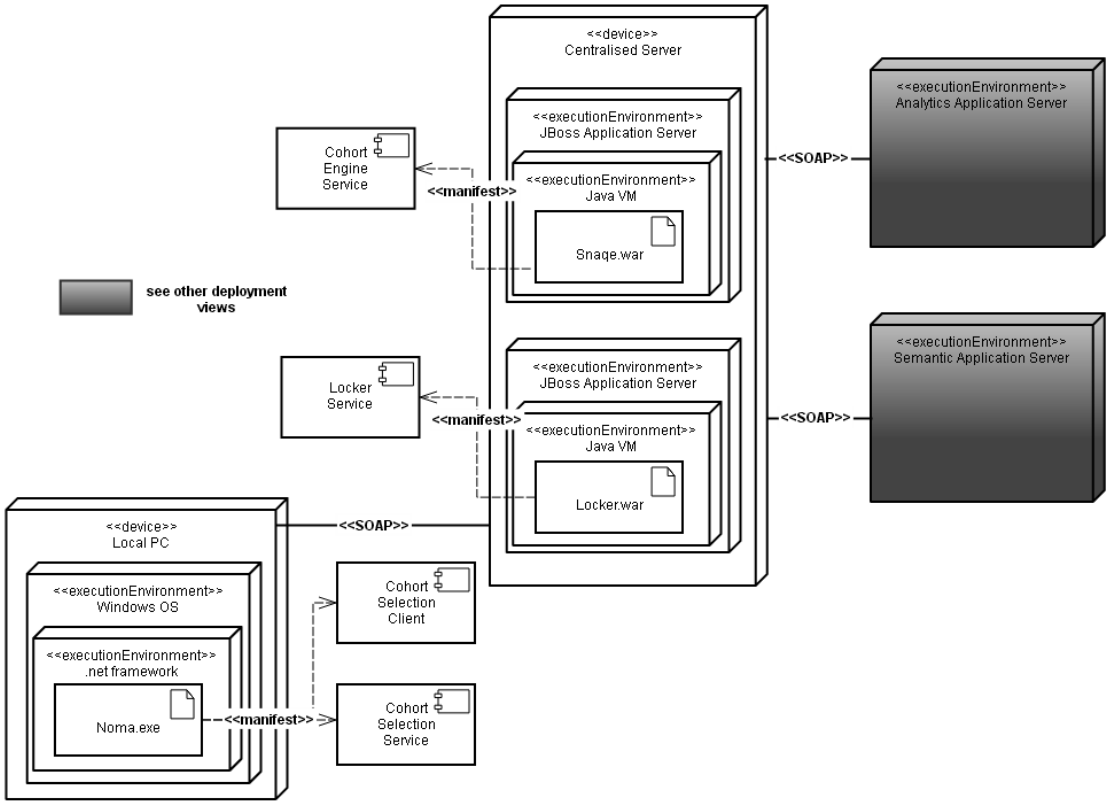


Figure 23: Cohort Selection Deployment Diagram

8.3.2 Nodes

8.3.2.1 Centralised Server

Artifact	Component	Environment	Application Server	OS	Requirements
Snaqe.war	Cohort Engine Service	Java VM version 1.6 or higher	Tomcat	N/A	• 1 CPU Core • 1 GB RAM • 6 GB MEM
Locker.war	Locker Service	Java VM version 1.6 or higher	Jboss	N/A	• 1 CPU Core • 1 GB RAM • 6 GB MEM

8.3.2.2 Local PC

Artifact	Component	Environment	Application Server	OS	Requirements
Noma.exe	• Cohort Selection Client • Cohort	.NET framework	N/A	Windows 7 or higher	• 4 GB RAM • 3 GB MEM

	Selection Service				

8.4 Semantic Layer

8.4.1 Diagram

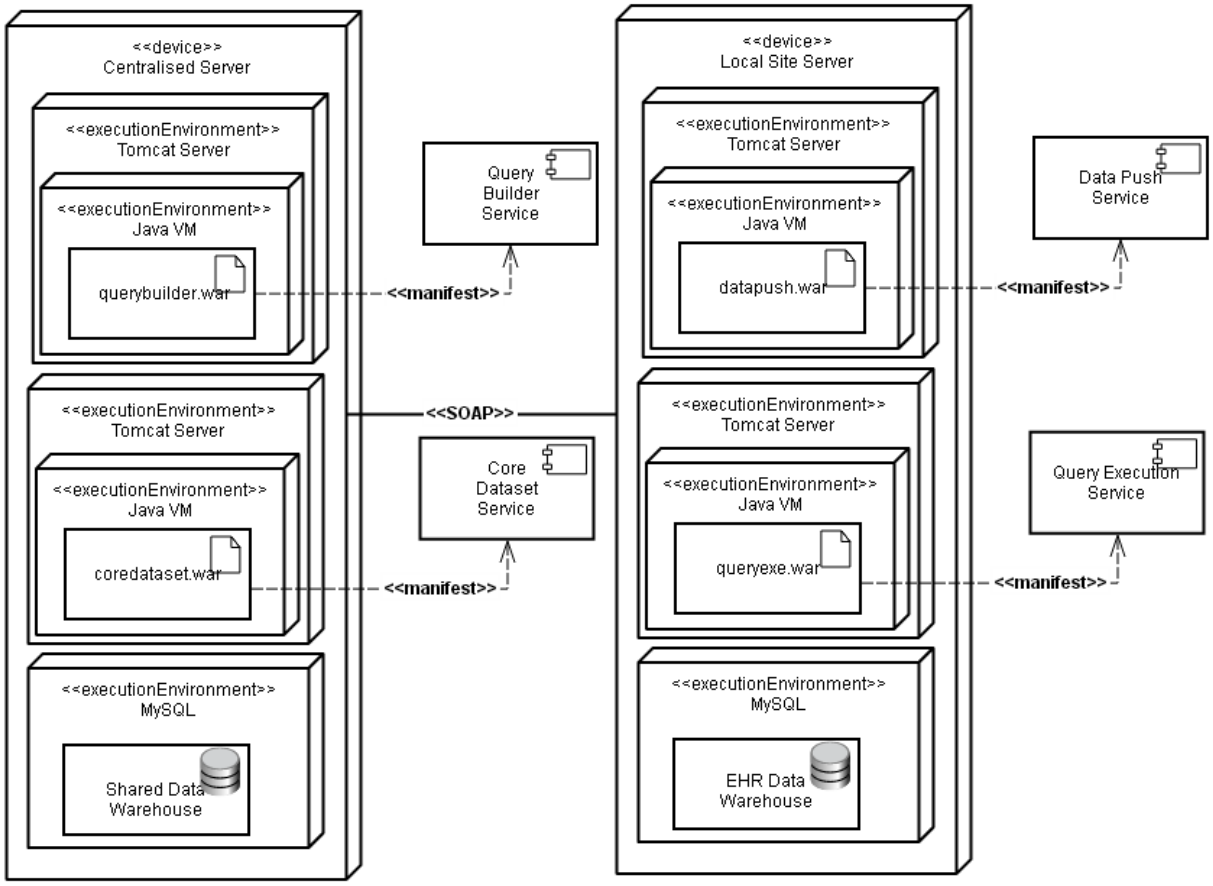


Figure 24: Semantic layer deployment view

As is shown in Figure 24, Semantic Interoperability Layer services use SOAP protocol for enabling the exchange of structured information between the different components. This layer interacts with the different applications of the INTEGRATE platform, as a core dataset information, different data sources and the SPARQL queries to be executed on CDM.

For that purpose, it is deployed in two different environments; local site and centralized. In the local site environment will be placed the services and database system related with the data because of the privacy and security protection. In this environment will be deployed the data push service, query execution service and the EHR data warehouse. In the centralized environment will be deployed the core dataset service and query builder service, because this services provide different methods for inferring knowledge and normalization and generation of queries. Therefore, the

heaviest services and components will be placed in a centralized server avoiding speed problems.

8.4.2 Nodes

8.4.2.1 Centralised Server

Artifact	Component	Environment	Application Server	OS	Requirements
querybuilder.war	Query Builder Service	Java VM version 1.7 or higher	Tomcat	UNIX	• 1 CPU Core • 1 GB RAM • 6 GB MEM
coredataset.war	Core Dataset Service	Java VM version 1.7 or higher	Tomcat	UNIX	• 1 CPU Core • 1 GB RAM • 6 GB MEM
Shared Data Warehouse		MySQL 5.5 or higher	N/A	N/A	• 1 CPU Core • 1 GB RAM • 6 GB MEM

8.4.2.2 Local Site Server

Artifact	Component	Environment	Application Server	OS	Requirements
datapush.war	Data Push Service	Java VM version 1.7 or higher	Tomcat	UNIX	• 1 CPU Core • 1 GB RAM • 6 GB MEM
dataexe.war	Data Execution Service	Java VM version 1.7 or higher	Tomcat	UNIX	• 1 CPU Core • 1 GB RAM • 6 GB MEM
EHR Data Warehouse		MySQL 5.5 or higher	N/A	N/A	• 1 CPU Core • 1 GB RAM • 6 GB MEM

8.5 Central Pathology Review

8.5.1 Introduction

The platform for the Central Review of Pathology Images of the INTEGRATE, is created using a set of open source components, where most of them are of proven value for use in heavy load production environments.

The only requirement for the stakeholders, in order to access the Central Review platform, is a modern web browser (the platform has been tested with the majority of the available web browsers, on desktop and mobile devices, at their most recent stable build).

A core component for the functionality of the image viewer of the platform is the Imaging Service, a component that slices the big pathology images (multi gigabyte images with huge resolutions) which cannot be displayed directly to the browser, to smaller compatible images (aka tiles) whose format is browser compatible and of small

size, appropriate for internet usage. These tiled images are subsequently available to CPR platform via a secure HTTP connection.

The Central Pathology platform is using SOAP services, in order to communicate with the Central INTEGRATE repositories and synchronize them with the local ones.
The above is graphically presented in the following deployment diagram.

8.5.2 Diagram

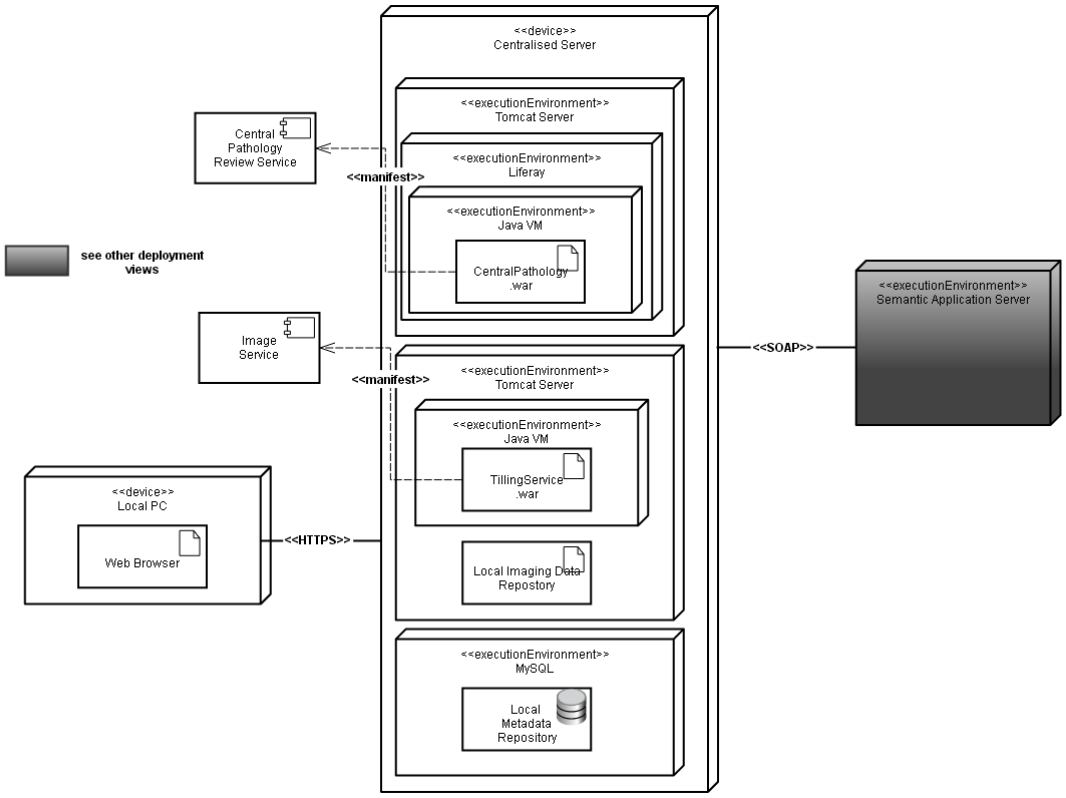


Figure 25: INTEGRATE Central Pathology Review Platform deployment view

8.5.3 Nodes

8.5.3.1 Local PC

CPR is a web based platform so it can be accessed by the end user from a web browser.

Artifact	Component	Environment	Application Server	OS	Requirements
Web Browser		N/A	N/A	N/A	• 2 GB RAM

8.5.3.2 Centralised Server (Central Pathology Review Platform)

The INTEGRATE Central Pathology Review (CPR) platform provides to the stakeholders all the necessary tools & functionality in order to have images reviewed by multiple Reviewers and to manage and log the whole procedure. Moreover it

provides collaboration capabilities among the stakeholders such as messaging, scheduling, and more.
Imaging Service is an independent component responsible for processing raw pathology images of big size so as to be accessible and viewable by the CPR platform.

Artifact	Component	Environment	Application Server	OS	Requirements
CentralPathology.war	Central Pathology Review Service	Java VM version 1.7 or higher	Tomcat	N/A	• 2 CPU core • 8 GB RAM
TillingService.war	Imaging Service	Java VM version 1.7 or higher	Tomcat	N/A	• 4 CPU core • 16 GB RAM
Local Imaging Data Repository		File System	N/A	N/A	• 1 CPU core • 2 GB RAM
Local Metadata Respository		MySQL	N/A	N/A	• 1 CPU core • 2 GB RAM

8.6 Analytical Tools

8.6.1 Diagram

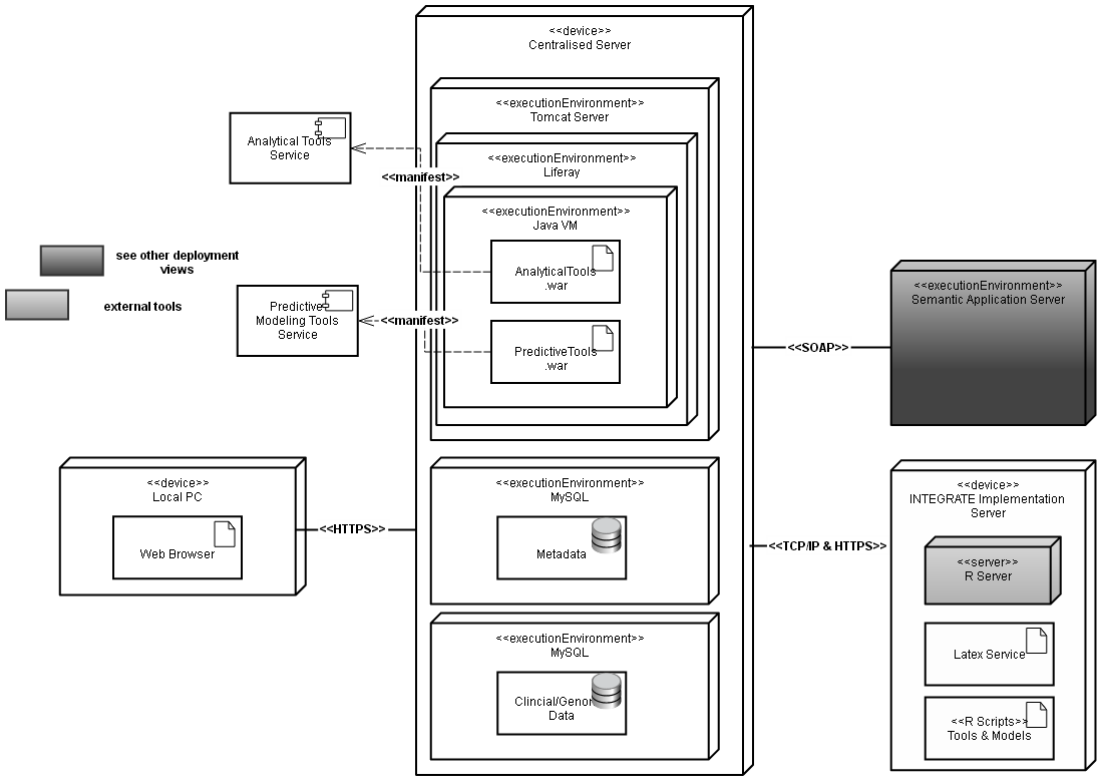


Figure 26: INTEGRATE Analysis Platform deployment view

8.6.2 Nodes

8.6.2.1 INTEGRATE Implementation Server

Due to the high computational cost of some analysis scenarios all the functionality required for running this analysis is performed by a server, named as "INTEGRATE Implementation Server". This server contains the core software for running the analysis (Rserve and R scripts) and generating dynamically the analysis reports (Latex service).

Artifact	Component	Environment	Application Server	OS	Requirements
R server		N/A	N/A	Latest Ubuntu Linux LTS	• RAM: 32 GB • Rserve: 0.6-8.1 or later
Latex Service		N/A	N/A	N/A	• Tex Live 2012 • Texmaker LaTex editor: 3.4 or later

8.6.2.2 INTEGRATE Analysis Server

The aforementioned server interacts with the "INTEGRATE Analysis Server", which is one of the main components of the INTEGRATE Platform Server as presented in deliverable D.2.4. The front-end is based on the Liferay Portal, an enterprise web platform based on Java technologies.

The "INTEGRATE Analysis Server" assist users in defining and running research queries and viewing the results of an analysis scenario. The available data are retrieved from the INTEGRATE Common Data Model via a web service and stored in the platform's database before the analysis.

Artifact	Component	Environment	Application Server	OS	Requirements
AnalyticalTools.war	Analytical Tools Service	Java VM version 1.6 or higher	Tomcat	N/A	• Portal Framework: Liferay Portal 6.1.1 Community Edition GA2
PredictiveTools.war	Predictive Modelling Tools Service	Java VM version 1.6 or higher	Tomcat	N/A	• Portal Framework: Liferay Portal 6.1.1 Community Edition GA2
Metadata	Metadata Management Service	MySQL 5.5.X	N/A	N/A	
Clinical/Genomic Data	Clinical/Genomic Management	MySQL 5.5.X	N/A	N/A	

	Service				

9 (PART II) Security Framework

9.1 Introduction

The goal of the INTEGRATE security framework is to provide a technological solution that covers all identified security requirements and guarantees compliance of the complete INTEGRATE platform to the legal framework governing the project (see deliverables 1.1, 1.2 and 1.3).

It will consist of modular components, respectively dealing with **authentication**, **authorisation**, **audit** and **privacy enhancing** techniques. The focus is on creating generic, re-useable components (in view of exploitation) that are developed according to the design principles defined in the architecture:

- The components should be designed according to the principles of a loosely coupled, open and scalable Service Oriented Architecture (SOA), focusing mainly on the interoperability and interfacing between the different systems and services. The integration guidelines of the security components were explained in deliverable 2.5/4.3.
- The components should adopt international security standards as much as possible (see deliverable 2.1).

The use of security standards and service level interfaces will be maximised. However, it cannot be denied that from a functional point of view, the different security components are rather tightly coupled. The challenge is to catch this coupling mainly by configuration, state transfer and (proprietary) glue logic.

In distributed environments, there are many cases in which advanced security functionality can only be implemented through a correct combination of identity provisioning and policy structure (for example role hierarchy with ABAC). In general with distributed configuration, care should be taken that no inconsistencies are introduced that could lead to a discrepancy between intended and enforced security policies (i.e. resulting in unwanted denial of service or security breaches).

9.2 High Level Overview

Figure 27 gives an overview of the security framework, containing the different security components and sub-components.

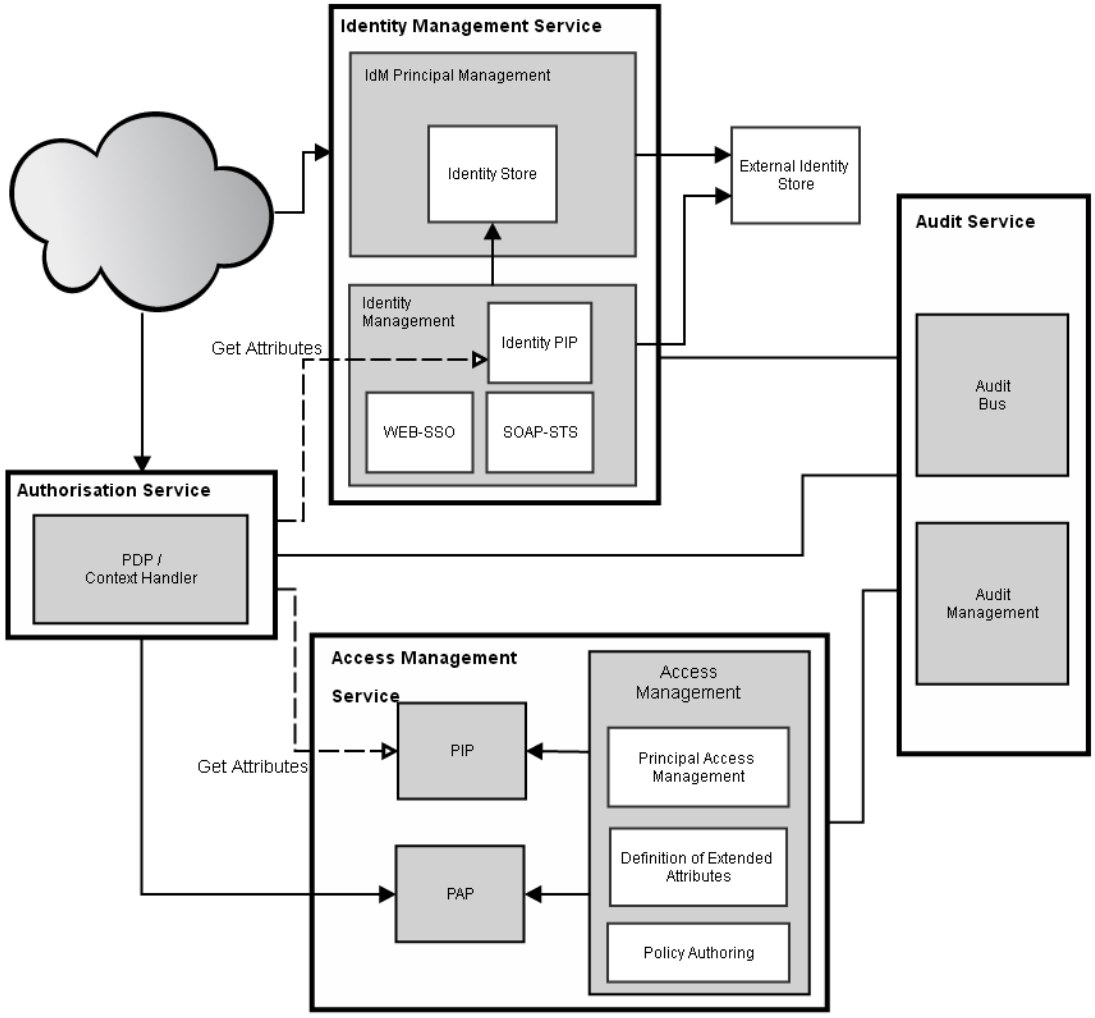


Figure 27: Security Framework Overview

The figure can be split up in two main blocks, on the left side the components of the framework are presented that are used for the security integration in the different INTEGRATE services/applications (discussed in deliverable 2.5) and on the right side the components that are part of the central security platform, namely:

- Identity Manager
  - The Identity Manager provides a front-end for user and service identity management and incorporates the modules for identity provision on a technical level, i.e. Identity Providers (IdPs) supporting Web Single Sign-On (SSO) and WS Security Token Service (STS) capabilities.
- Access Manager
  - The access management component is a management front-end that generates Access Control (AC) policies that are used by the authorisation services for evaluating access requests.
- Authorization Service
  - This service evaluates access request from all over the INTEGRATE infrastructure based on the security policies. The architecture and standards used allow this service to be easily implemented as a

distributed service (scalability). In the test-bed and pilot settings of INTEGRATE this is not expected to be necessary performance wise.

- Audit Service
  - Provides auditing mechanisms.

9.3 Central Security Framework

9.3.1 Identity Management Service

9.3.1.1 Principal Management

The principal management solution keeps track of the INTEGRATE principals (users and services) in a hierarchical tree of domains and organisations (see Figure 28). Domains are used to group principals. A domain is globally identified by its globally unique identifier (GUID). Domains are mutual exclusive; this means a principal can only be part of one domain. Next to a basic set of standard attributes (like name, organisation, etc.), it is possible to define custom attributes for the principals. For authenticating (see further) a credential (like username/password, credential, etc.) can be stored with each principal. Finally the principal management offers user registration functionality like account creation, reactivation, etc.

A graphical user interface has been developed to present the principal management functionality in a user-friendly way to the end-user (see Figure 28).

Figure 28: Identity Management

To manage all the principals, LDAP (Lightweight Directory Access Protocol) is used. LDAP is an application protocol that can be used to access and maintain distributed directory information services over an internet protocol network. INTEGRATE will include an already existing implementation of the LDAP protocol like OpenDS[9] or OpenLDAP[10].

9.3.1.2 Identity Provision

For the exchange of identity information an attribute-based approach is chosen, which means that user attributes (e.g., the user's name, his date of birth, his email address, his clearance level, the authentication assurance level, etc.) are exchanged in a standardised format called a security token.

[9] OpenDS, Open Source Java LDAP Directory Service, http://www.opends.org/
[10] OpenLDAP, http://www.openldap.org/

For the issuing of security tokens two issue components are provided, an Identity Provider (IdP) and a Security Token Service (STS), each implementing a specific part of the INTEGRATE required identity provision functionality.

The identity provider (IdP) component provides an implementation of the SSO browser based SAML profile[11]. If a user tries to access a protected resource of an INTEGRATE service through his **browser** (over HTTP(S)), he is redirected to the IdP component who will issue a security token after the user has authenticated him/herself to the system by means of a security credential (username/password, certificate, security token, etc.). Authentication will succeed if the presented credential exists in the identity store (see identity management). Finally the issued security token will be validated by the INTEGRATE service that hosts the protected resource and used to make an access decision (see further).

The Shibboleth[12] IdP implementation (see deliverable 2.2) was used as starting point for developing the INTEGRATE specific IdP. The standard functionality of Shibboleth IdP was extended and configured to meet the specific security requirements of the INTEGRATE platform.

The Secure Token System (STS) component provides an implementation of the WS-Trust[13] specification. In this case a user wants to access a protected resource of a service belonging to the platform through a **service or application** (over SOAP). The STS will issue a security token after the user has authenticated him/herself to the system (authentication will succeed if credential exists in the identity store). This security token will then be validated by the service that hosts the protected resource. The Apache CXF STS[14] implementation is used for issuing security tokens.

9.3.2 Access Management Service

9.3.2.1 Policy Administration Point

The Policy Administration Point (PAP) is responsible for authoring and management of access control policies. These access control policies are a formal set of rules that define what action, if any, a subject can take on a particular resource. For example, reading (action) the medical file of patient A (resource) is restricted only to the threating physician B (subject). Using these policies the authorisation service can make a decision (grant or deny access) on an incoming access request (see further).

In large frameworks like INTEGRATE, this authoring and management of policies can become very complex. That is why it is important to choose a suitable policy standard that supports the access requirements defined in the framework (whether with extensions or not). Several solutions are available for defining and enforcing complex

[11] SAML profiles, 2005, "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0", available from: http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf. [1 February 2013]
[12] Shibboleth, http://shibboleth.net/
[13] WS-Trust, 2007, "WS-Trust 1.3", available from: http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.pdf [1 February 2013]
[14] CXF STS, http://cxf.apache.org/docs/ws-trust.html

policies, see deliverable 2.1. After a thorough evaluation of INTEGRATE's access control requirements, it was decided to use XACML as the authorisation solution.

9.3.2.2 Policy Information Point

The Policy Information Point (PIP) is responsible for the resolution of possible missing attributes in an access control request coming from the Policy Enforcement Point (PEP). These attributes are needed by the authorisation service, i.e. the Policy Decision Point (PDP), to come to an accurate decision about the request. If this authorisation service needs further information to evaluate the incoming request from the PEP, the PIP is called, either directly by the authorisation service itself or by returning missing attributes to the PEP which have to invoke the PIP.

9.3.2.3 Access Manager

The access management provides a user interface to the access control administrators that enable them to manage and author policies and attribute information in a structured way. Advanced user interfaces will be required in order to display the complex structure of the policies.

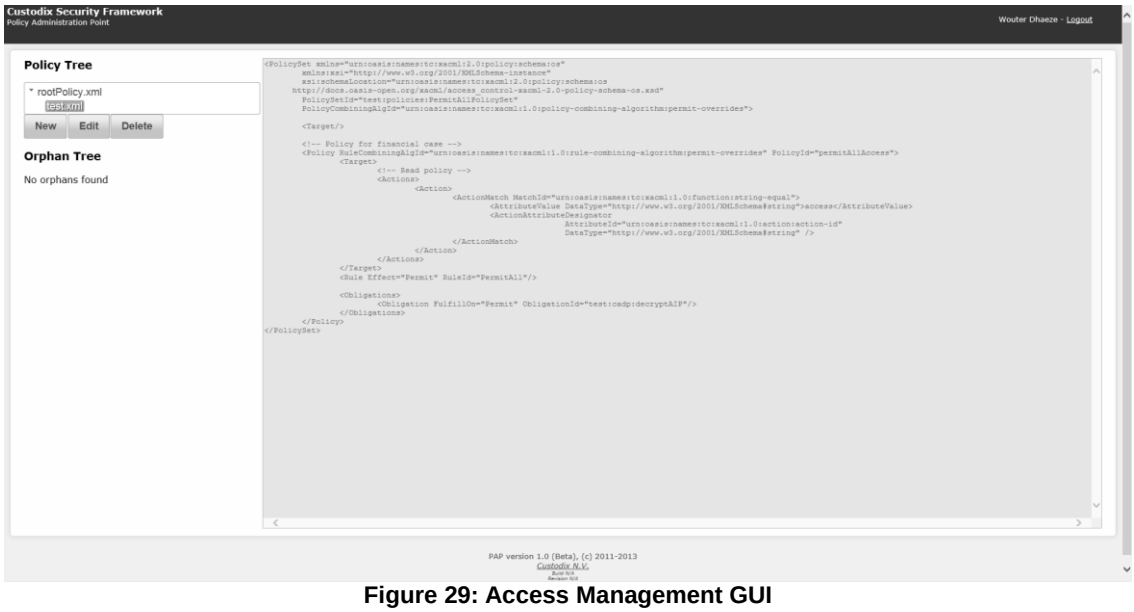


Figure 29: Access Management GUI

9.3.3 Authorisation Service

The authorisation service will augment and evaluate access requests coming from the different applications that restrict their resources by access control. The authorisation service evaluates these requests based on the (AC) policies that were defined in the access management and the requester security context.

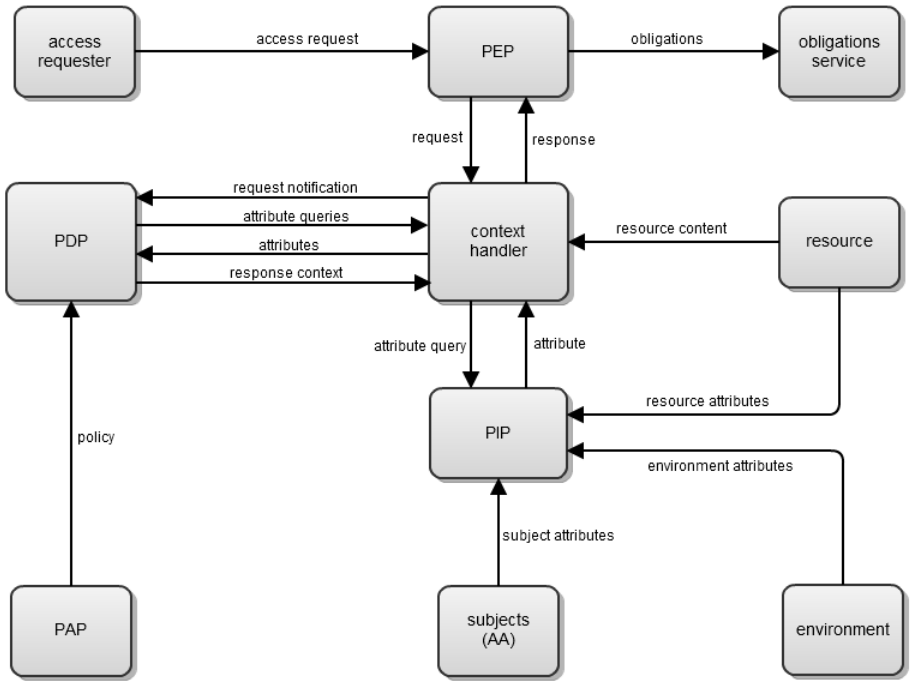


Figure 30: XACML Dataflow Diagram

As described in the previous section, it has been decided that the policies defined in INTEGRATE will be XACML based. This means that by using these XACML based policies, the authorisation service should be able to validate XACML decision request messages sent by Policy Enforcement Points (PEPs). The component responsible for the validation (making access decisions) is called the Policy Decision Point (PDP).

Next to a PDP, a context handler component is part of the authorisation service. This component will augment the decision request messages coming from the PEPs with additional (probably missing) attributes obtained from Attribute Authorities (AAs) or Attribute Repositories. The AAs are accessed through a Policy Information Point (PIP) (see beneath). The context handler will also introduce supporting components complementary to a standard XACML PDP specification, which solve some of the limitations (like contextualisation) encountered in this specification.

The general flow is straightforward. The PEP intercepts access requests by a principal on one or more protected resources on a server. The PEP generates a decision request based on the attributes of the subject, the resource in question, the performed action and other information pertaining to this access request. This decision request is send to the Authorisation Service. Here, the request is possibly augmented by the context handler component which passes it to the PDP. The PDP will interpret the request and searches for policies (coming from the PAP) that apply to the request. Based on the rules defined in the found policies, the PDP will make an access decision and includes this decision in a decision response message. This message is sent back to the PEP. The PEP will use this response to decide if access is granted to the protected resource or not.

INTEGRATE will use the Balana[15] PDP engine for generating access decisions for incoming AC requests. Both XACML v2.0 as v3.0 request/responses and policies are supported in this engine. The problem of contextual attributes (described in deliverable D2.3/2.4) was worked out and written down in a paper that was presented at the HEALTHINF 2013 conference in Barcelona (Brecht Claerhout et al., 2013). To test the approach taken in this paper, a contextualisation handler extension (part of an extension mechanism placed between the PDP and context handler) was implemented in the XACML authorisation language for INTEGRATE.

9.3.4 Audit Service

Every security framework contains a (centralised) audit service. Each authentication attempt (both successful and failed), resource access/change, available issue (e.g. server exceptions), etc. needs to be logged by the audit service. Next to the logging functionality, the audit service needs to present the logs to the administrators in such a way that they can be easily consulted and interpreted.

INTEGRATE will use Could Auditing Data Federation[16] (CADF) data format and interface specification as event model and audit log record specification.

9.4 De-identification & Consent

The specification and status of the INTEGRATE de-identification and consent service(s) is part of Task 4.4 *Privacy Enhancing Processes and Services* and will be discussed in deliverable 4.6.

9.5 Security Integration

The integration of clients and service providers with the INTEGRATE security infrastructure was described in deliverable 2.5.

The LifeRay portal used for the analytical tools and central pathology review demonstrators uses the central IdP for authentication. For this a LifeRay plugin was written that integrates the IdP functionality with the internal LifeRay security. The different SOAP based INTEGRATE services/clients of the presented demonstrators in year 1 are integrated (using configuration or proxy) with WS-Trust and connected to the central STS. In the semantic layer, fine-grained access control was implemented. More information about this integration can be found in deliverable 2.5.

9.6 Summary

The aim of the INTEGRATE security framework was to create a framework that consists of modular and re-usable components, respectively dealing with authentication, authorisation, audit and privacy enhancing techniques. The services that were presented here fulfil to this aim. They are result of an iterative process, starting from the security requirements described in deliverable 2.4.

[15] Balana - http://xacmlinfo.org/category/balana/
[16] CADF - http://dmtf.org/sites/default/files/standards/documents/DSP0262_1.0.0b.pdf

10 REFERENCES

- Claerhout B., De Schepper K., Pérez del Rey D., and Bucur A., 2013, 'Contextualisation of ABAC Attributes through a Generic XACML Functionality Extension Mechanism', HEALTHINF 2013, pp. 52-57
- Brecht Claerhout, Kristof de Schepper, David Pérez-Rey, Raúl Alonso-Calvo, Jasper van Leeuwen, Anca I. D. Bucur, 'Implementing patient recruitment on EURECA semantic integration platform through a Groovy query engine', BIBE 2013, pp. 1-5

11 Table of Figures

12 Appendix A – trial metadata class definitions

<table>
<tr><th>Class</th><th>Description</th></tr>
<tr><td>Activity</td><td>DEFINITION:

Any action that can, in the context of a study or a post-marketing investigation, be defined, planned, scheduled or performed.

EXAMPLE(S): Administrative activities such as subject registration or informed consent Clinical activities such as surgical procedure, laboratory test, administration of a drug

OTHER NAME(S):

NOTE(S): This is similar in idea to HL7 RIM's Act class

<table>
<tr><th>Attribute name</th><th>Description</th></tr>
<tr><td>identifier</td><td>DEFINITION:

A unique symbol that establishes identity of an activity.

EXAMPLE(S): 12345 is the identifier for a substance administration

OTHER NAME(S):

NOTE(S):</td></tr>
</table>
</td></tr>
<tr><td>Arm</td><td>DEFINITION:

A path through the study which describes what activities the study subject or experimental unit will be involved in as they pass through the study.

EXAMPLE(S): A study could have 2 arms named IV-Oral and Oral-IV. The name IV-Oral reflects a path that passes through IV treatment, then Oral treatment.

OTHER NAME(S): Group [CTRR Observational</td></tr>
</table>

<table>
<tr><th>Class</th><th>Description</th></tr>
<tr><td></td><td>Studies]

NOTE(S): An Arm is typically equivalent to a treatment group in a parallel design study. Generally, each subject is assigned to an arm, and the design of the study is reflected in the number and composition of the individual arms. This intended path through which the subject progresses in a study is composed of time point events (study cell) for each epoch of the study. Each time point event, in turn, has a pattern of child time points through which the subject would pass. This planned path thus describes how subjects assigned to the arm will be treated.

<table>
<tr><th>Attribute name</th><th>Description</th></tr>
<tr><td>name</td><td>DEFINITION:

A non-unique textual identifier for the arm.

EXAMPLE(S): Treatment A

OTHER NAME(S):

NOTE(S):</td></tr>
<tr><td>typeCode</td><td>DEFINITION:

A coded value specifying the kind of arm.

EXAMPLE(S): Experimental, Active Comparator, Placebo Comparator, Sham Comparator, No intervention, Other

OTHER NAME(S):

NOTE(S):</td></tr>
</table></td></tr>
<tr><td>CriterionMap</td><td>CriterionMap provides additional information for the visualization of the related widget definition
<table>
<tr><th>Attribute name</th><th>Description</th></tr>
<tr><td>value</td><td>textual representation of the</td></tr>
</table></td></tr>
</table>

Class	Description
	widgetdefinition DEFINITION: classification: Code representing the class of the widget (e.g. lab test, demographics)
Date	
DefinedActivity	DEFINITION: An activity that frequently occurs in studies (e.g. more than one time in more than one arm) and therefore is called out as a reusable template in a global library of activities outside the context of any particular study, and may be used in the composition of a defined subject activity group. A defined activity is a "kind of" activity rather than an "instance of" an activity. EXAMPLE(S): Standard blood chemistries are frequently included in studies - also activities that are study-specific and recur more than one time in more than one arm may be defined, such as a substance administration activity involving X amount of drug Y. OTHER NAME(S): NOTE(S): A defined activity is represented here as a subtype of Activity, but could also be thought of as an activity at a particular stage in the business process in which the activities occur, i.e., in the "defined" stage rather than the "planned" stage, the "scheduled" stage or the "performed" stage.
	Attribute name: **description** Description: DEFINITION: The textual representation of the activity. EXAMPLE(S): OTHER NAME(S): NOTE(S): This may contain more

<table>
<tr><th>Class</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>detail than the description present in the text part of a coded concept.</td></tr>
<tr><td>DefinedAdministrativeActivity</td><td colspan="2">DEFINITION:

An activity defined at a global library level that is not directly related to hypothesis evaluation or testing, but is typically essential to the efficient and/or effective coordination and execution of a study.

EXAMPLE(S): assignment to a treatment arm, registration to a study, start of on-study period, end of on-study period, obtain informed consent, verify eligibility criteria, enroll, randomize, complete study visits, exit trial, break treatment blind, protocol violation, premature withdrawal

OTHER NAME(S):

NOTE(S):

<table>
<tr><th>Attribute name</th><th>Description</th></tr>
<tr><td>description</td><td>DEFINITION:

The textual representation of the activity.

EXAMPLE(S):

OTHER NAME(S):

NOTE(S): This may contain more detail than the description present in the text part of a coded concept.</td></tr>
</table>
</td></tr>
<tr><td>DefinedCompositionRelationship</td><td colspan="2">DEFINITION:

A relationship between a composite activity and a component activity that comprises it, i.e. parent and child activities, where all these activities are part of a global library of activities.

EXAMPLE(S): A battery of tests may be composed of multiple routine labs that are always ordered</td></tr>
</table>

<table>
<tr><th>Class</th><th>Description</th></tr>
<tr><td></td><td>together as a group.

A glucose tolerance test which is comprised of administering glucose and taking multiple timed blood samples which are then tested for glucose.

OTHER NAME(S):

NOTE(S): This class helps represent an AND relationship between siblings with the same parent activity.

<table><tr><th>Attribute name</th><th>Description</th></tr><tr><td>sequenceNumber</td><td>DEFINITION:

An integer specifying the relative sequential or temporal ordering of this relationship among other similar relationships having the same source.</td></tr></table></td></tr>
<tr><td>DefinedCriterionGroup</td><td>DEFINITION:

A collection of conditions joined together via composition (ANDed) and/or optionality (ORed) to form a logical expression upon which the execution of an activity is based or upon which the cessation of a repeated activity is based, where components of the group may include other activities, observation results and/or other criterion groups, and where both the criterion group and it's components are defined as part of the global library.

EXAMPLE(S): (A and (B or C)), where A might be an activity, B and C might be 2 different observation results, and the two sets of parentheses are 2 criterion groups, one inside (a component of) the other.

OTHER NAME(S):

NOTE(S): A criterion group represents the parentheses around a set of criteria in a logical expression.</td></tr>
<tr><td>DefinedCriterionGroupCompositionRelat</td><td>DEFINITION:</td></tr>
</table>

<table>
<tr><th>Class</th><th>Description</th></tr>
<tr><td>ionship</td><td>A relationship between a criterion group and an activity, observation result or other criterion group that is a component of the group, i.e. a relationship between a logical set of parenthesis and one of the items inside the parentheses, where the criterion group and its components are both part of a global library of activities.

EXAMPLE(S): A battery of tests may be composed of multiple routine labs that are always ordered together as a group.

A glucose tolerance test which is comprised of administering glucose and taking multiple timed blood samples which are then tested for glucose.

OTHER NAME(S):

NOTE(S): This class helps represent an AND relationship between siblings in the same criterion group.</td></tr>
<tr><td>DefinedEligibilityCriterion</td><td>DEFINITION:

An activity defined at a global library level that identifies one of a set of conditions that a subject must meet in order to participate in a study, or that a study subject must meet into order to participate in a certain part of the study.

EXAMPLE(S):

OTHER NAME(S):

NOTE(S):

<table>
<tr><th>Attribute name</th><th>Description</th></tr>
<tr><td>description</td><td>DEFINITION:

The textual representation of the activity.

EXAMPLE(S):</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Class</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>OTHER NAME(S):

NOTE(S): This may contain more detail than the description present in the text part of a coded concept.</td></tr>
<tr><td>DefinedExclusionCriterion</td><td colspan="2">DEFINITION:

An activity defined at a global library level that identifies a characteristic or requirement intended to be applied to a potential study subject to determine whether they may not participate in a study.

EXAMPLE(S): Must be over the age of 18.

OTHER NAME(S):

NOTE(S):

<table><tr><th>Attribute name</th><th>Description</th></tr><tr><td>description</td><td>DEFINITION:

The textual representation of the activity.

EXAMPLE(S):

OTHER NAME(S):

NOTE(S): This may contain more detail than the description present in the text part of a coded concept.</td></tr></table></td></tr>
<tr><td>DefinedExperimentalUnitAllocation</td><td colspan="2">DEFINITION:

An administrative activity defined at a global library level that is the assignment of an experimental unit to a portion of the study, such as an arm or a portion of an arm (when secondary allocations may occur).

EXAMPLE(S): randomization, direct assignment based on eligibility criteria, etc.

"Escalating dose cohort studies" enroll subjects in successive arms, i.e., one arm is completely filled before any subjects are enrolled in the next arm. In</td></tr>
</table>

<table>
<tr><th>Class</th><th>Description</th></tr>
<tr><td></td><td>such a study, allocation depends on which arms have been fully enrolled and which are currently open for enrollment. Note that this example assumes that the experimental unit is the subject (rather than a part of a subject or a group of subjects).

OTHER NAME(S):

NOTE(S):

<table><tr><th>Attribute name</th><th>Description</th></tr><tr><td>description</td><td>DEFINITION:

The textual representation of the activity.

EXAMPLE(S):

OTHER NAME(S):

NOTE(S): This may contain more detail than the description present in the text part of a coded concept.</td></tr></table></td></tr>
<tr><td>DefinedInclusionCriterion</td><td>DEFINITION:

An activity defined at a global library level that identifies a characteristic or requirement intended to be applied to a potential study subject to determine whether they may participate in a study.

EXAMPLE(S): pregnancy

OTHER NAME(S):

NOTE(S):

<table><tr><th>Attribute name</th><th>Description</th></tr><tr><td>description</td><td>DEFINITION:

The textual representation of the activity.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Class</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>EXAMPLE(S):

OTHER NAME(S):

NOTE(S): This may contain more detail than the description present in the text part of a coded concept.</td></tr>
<tr><td>DefinedInformedConsentActivity</td><td>Attribute name</td><td>Description</td></tr>
<tr><td></td><td>description</td><td>DEFINITION:

The textual representation of the activity.

EXAMPLE(S):

OTHER NAME(S):

NOTE(S): This may contain more detail than the description present in the text part of a coded concept.</td></tr>
<tr><td>DefinedObservation</td><td colspan="2">DEFINITION:

An activity defined at a global library level whose intention is to obtain a result by observing, monitoring, measuring or otherwise qualitatively or quantitatively gathering data or information about one or more aspects of a subject's, study subject's or experimental unit's physiologic or psychologic state.

EXAMPLE(S): blood chemistry panel, body mass index calculation, blood pressure measurement

OTHER NAME(S):

NOTE(S):

Attribute name | Description
description | DEFINITION:

The textual representation of the activity.</td></tr>
</table>

<table>
<tr><th>Class</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>EXAMPLE(S):

OTHER NAME(S):

NOTE(S): This may contain more detail than the description present in the text part of a coded concept.</td></tr>
<tr><td>DefinedObservationResult</td><td colspan="2">DEFINITION:

A reusable, "template" description of possible findings of an observation.

EXAMPLE(S): A blood pressure measurement may result in a diastolic number and a systolic number.

OTHER NAME(S):

NOTE(S): The DefinedObservationResult class can be used to represent defined ranges for contingencies by constraining the result attribute from ANY to IVL<PQ>, for instance, or any other range value. Such DefinedObservationResults may be used as criteria for conditional activities or repeated activities.</td></tr>
<tr><td>DemographicsMap</td><td colspan="2">DemographicsMap provides additional information for the visualization of the related widget definition (demographics).
<table>
<tr><th>Attribute name</th><th>Description</th></tr>
<tr><td>upperLimit</td><td>Indicative upper limit for the value (used for displaying purposes). This is NOT a hard upper limit.</td></tr>
<tr><td>lowerLimit</td><td>Indicative lower limit for the value (used for displaying purposes). This is NOT a hard upper limit.</td></tr>
<tr><td>gender</td><td>Required gender</td></tr>
<tr><td>value</td><td>textual representation of the widgetdefinition</td></tr>
<tr><td>classification</td><td>DEFINITION:

Code representing the class of the widget (e.g. lab test, demographics)</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Class</th><th>Description</th></tr>
<tr><td>DocumentAuthor</td><td>DEFINITION:

The individual who is responsible for the content of a document.

EXAMPLE(S): A healthcare provider could be the author of a study protocol document

OTHER NAME(S):

NOTE(S):</td></tr>
<tr><td>DocumentVersion</td><td>DEFINITION:

A representation of a particular edition or snapshot of a document as it exists at a particular point in time.

EXAMPLE(S): Version 3 of a case report form (CRF) for a physical exam, version 2 of an informed consent form.

OTHER NAME(S):

NOTE(S):

<table>
<tr><th>Attribute name</th><th>Description</th></tr>
<tr><td>text</td><td>DEFINITION:

A textual or media-based representation that is the full or comprehensive narrative or content of the document.

EXAMPLE(S):

OTHER NAME(S):

NOTE(S):</td></tr>
<tr><td>officialTitle</td><td>DEFINITION:

The formal title of the document.

EXAMPLE(S):</td></tr>
</table>
</td></tr>
</table>

Class		Description
	date	OTHER NAME(S): NOTE(S): If there is only one title, use this attribute. DEFINITION: The date (and time) on which the document is versioned. EXAMPLE(S): OTHER NAME(S): NOTE(S): DEFINITION:
	uniformResource Locator	A complete or local reference to a website, ftp, file path or other location from which the document contents can be retrieved. EXAMPLE(S): OTHER NAME(S): NOTE(S): Derived from DocumentVersion.text.ED.reference. Local references should only be used when communicating between systems capable of resolving the local reference. In a Regulatory Product Submission (RPS) message, this identifies the file (with a Uniform Resource Identifier (URI)), which is part of the documentation. A URI is a compact string of characters used to identify or name a resource. The main purpose

Class	Description	
		of this identification is to enable interaction with representations of the resource over a network, typically the World Wide Web, using specific protocols. URIs are defined in schemes defining a specific syntax and associated protocols.
double		
Epoch	DEFINITION: One of a set of ordered partitions of a subject's, study subject's or experimental unit's participation in a study. An Epoch represents a state within a study such that subjects in separate arms within that state are comparable. Each epoch serves a purpose in the study as a whole, typically exposing the subject to a treatment or preparing them for a treatment, or gathering post-treatment data. Activities and activity results control the subject's movement from one epoch to another. EXAMPLE(S): A study designed to assess the effects of treatments might have 3 epochs. A Screening Epoch in which subjects' eligibility is determined and baseline measurements are made. A Treatment Epoch during which treatments are given and effects of treatment are assessed. A Follow-up Epoch during which post-treatment assessments are conducted. OTHER NAME(S): NOTE(S): A subject moves from one epoch to another and can only be in one epoch at a time. The subject can only move to an epoch with a greater sequenceNumber. Activities in the same epoch but a different arm need not be similar in time and pattern. Subjects in different arms will not necessarily pass through the same epochs. Attribute name | Description	

Class		Description
	name	DEFINITION: A non-unique textual identifier for the epoch. EXAMPLE(S): first treatment epoch, second treatment epoch, first wash-out epoch, second wash-out epoch OTHER NAME(S): NOTE(S): When multiple Epochs have the same purpose (e.g., treatment), then the titles will probably include order numbers to distinguish them.
	typeCode	DEFINITION: A coded value specifying the kind of epoch. EXAMPLE(S): screening, treatment, follow-up OTHER NAME(S): NOTE(S):
	sequenceNumber	DEFINITION: An integer specifying the relative sequential or temporal ordering of this epoch among other similar epochs in a study. EXAMPLE(S): In a Study that has Screening, Treatment and Follow-Up epochs, the sequence number indicates which Epoch precedes the other.

<table>
<tr><th>Class</th><th>Description</th></tr>
<tr><td></td><td>OTHER NAME(S):

NOTE(S):</td></tr>
<tr><td>GeneralMap</td><td>GeneralMap provides additional information for the visualization of the related widget definition
<table><tr><th>Attribute name</th><th>Description</th></tr><tr><td>value</td><td>textual representation of the widgetdefinition</td></tr><tr><td>classification</td><td>DEFINITION:

Code representing the class of the widget (e.g. lab test, demographics)</td></tr></table></td></tr>
<tr><td>HealthcareFacility</td><td>DEFINITION:

An organization that devotes some or all of its resources (people, places, things) to the delivery of healthcare services (including the financial and administrative management of those resources).

EXAMPLE(S): Northwestern Memorial Hospital

OTHER NAME(S):

NOTE(S): A healthcare facility may be manifest as a single physical location (e.g. building), or, alternatively, as a distributed collection of physical spaces.</td></tr>
<tr><td>LabValueMap</td><td>LabValueMap provides additional information for the visualization of the related widget definition (a labvalue)
<table><tr><th>Attribute name</th><th>Description</th></tr><tr><td>upperlimit</td><td>Indicative upper limit for the value (used for displaying purposes). This is NOT a hard upper limit.</td></tr><tr><td>lowerLimit</td><td>Indicative lower limit for the value (used for displaying purposes). This is NOT a hard upper limit.</td></tr><tr><td>value</td><td>textual representation of the widgetdefinition</td></tr><tr><td>classification</td><td>DEFINITION:</td></tr></table></td></tr>
</table>

Class	Description	
		Code representing the class of the widget (e.g. lab test, demographics)
MatchingScript	**Attribute name Description** value type	
Organization		
Organization	DEFINITION: A formalized group of persons or other organizations collected together for a common purpose (such as administrative, legal, political) and the infrastructure to carry out that purpose. EXAMPLE(S): US National Cancer Institute (NCI); CDISC; HL7, ACME Corporation OTHER NAME(S): NOTE(S):	

Attribute name	Description
name	DEFINITION: A non-unique textual identifier for the organization. EXAMPLE(S): OTHER NAME(S): NOTE(S):
postalAddress	DEFINITION: A contact point used to send physical forms of communication to the organization. EXAMPLE(S): OTHER NAME(S):

Class	Description	
		NOTE(S):
Person	DEFINITION: A human being. EXAMPLE(S): OTHER NAME(S): NOTE(S):	
PlannedActivity	DEFINITION: An activity that is intended to occur or start at some point in the context of a particular study. EXAMPLE(S): Pregnancy tests are planned for study subjects who are females of childbearing potential. OTHER NAME(S): NOTE(S): A PlannedActivity may be a container of other activities and have a complex structure involving components, options and contingencies using the associated relationship classes. This structure allows the representation of concepts in previous versions of BRIDG such as StudyCells, StudySegments and StudySubjectEncounters. A PlannedActivity could also be thought of as an activity at a particular stage in the business process in which the activities occur, i.e., in the "planned" stage rather than the "scheduled" stage or the "performed" stage. An instance of a PlannedActivity is not assigned to a particular Subject, StudySubject, or ExperimentalUnit, but to a "kind of" Subject, StudySubject, or ExperimentalUnit.	
	Attribute name	Description
	studyDayRange	DEFINITION: The relative timing for a planned activity expressed as the number of days offset from the study-defined reference activity (e.g., date of

Class		Description
		registration, start of treatment) for a particular subject, study subject or experimental unit. EXAMPLE(S): Day 1, Days 10-20 OTHER NAME(S): Visit Day NOTE(S): Derived from all the pauseQuantity values of the composite activity structures that this activity is a part of minus the offset of the reference activity. The study-defined reference activity can be different from study to study. The study day for a date after this reference activity is a positive integer calculated as the difference in the two dates + 1. The study day for dates before the reference activity is a negative integer calculated as the difference between the two dates. Note that this means there is no "Day 0."
	identifier	DEFINITION: A unique symbol that establishes identity of an activity. EXAMPLE(S): 12345 is the identifier for a substance administration OTHER NAME(S): NOTE(S):
PlannedRandomizationBookAllocation	Attribute name	Description
	studyDayRange	DEFINITION:

Class		Description
	identifier	The relative timing for a planned activity expressed as the number of days offset from the study-defined reference activity (e.g., date of registration, start of treatment) for a particular subject, study subject or experimental unit. EXAMPLE(S): Day 1, Days 10-20 OTHER NAME(S): Visit Day NOTE(S): Derived from all the pauseQuantity values of the composite activity structures that this activity is a part of minus the offset of the reference activity. The study-defined reference activity can be different from study to study. The study day for a date after this reference activity is a positive integer calculated as the difference in the two dates + 1. The study day for dates before the reference activity is a negative integer calculated as the difference between the two dates. Note that this means there is no "Day 0." DEFINITION: A unique symbol that establishes identity of an activity. EXAMPLE(S): 12345 is the identifier for a substance

Class	Description
	administration OTHER NAME(S): NOTE(S):
PlannedStudySite	DEFINITION: A facility in which study activities are intended to be conducted. EXAMPLE(S): The site where the study subject encounter is intended to occur, or the site of the Investigator. OTHER NAME(S): NOTE(S):
RandomizationBookEntry	
ResearchStaff	DEFINITION: Individual who is employed and/or involved in any aspect of conduct of protocol driven research. EXAMPLE(S): administrators, clinical and data managers, clinical research pharmacists, clinical research associates, clinical trials compliance coordinators, clinical trials specialists, laboratory technologists, nurses, research services consultants, study coordinators and others OTHER NAME(S): NOTE(S):
StudyAccrualStatistics	Attribute name / Description **expectedEnrollmentCompetitiveTrialPercentage** **expectedDropoutPercentage** **expectedNoEnrollmentPercentage**
StudyActivity	DEFINITION: The intention to use a defined activity in the design of a study. EXAMPLE(S): If a study's design includes the

<table>
<tr><th>Class</th><th colspan="2">Description</th></tr>
<tr><td></td><td colspan="2">activity of taking blood pressure, the DefinedActivity for blood pressure is linked to the study via this class.

OTHER NAME(S):

NOTE(S): The number of times this activity occurs during the study and the relative timing for those occurrences is represented by PlannedActivity.</td></tr>
<tr><td>StudyExecution</td><td>Attribute name</td><td>Description</td></tr>
<tr><td></td><td>effectiveDateRange</td><td>DEFINITION:

Specifies the period of time over which the study was executed

EXAMPLE(S):

OTHER NAME(S):

NOTE(S): May specify the start and/or end or duration. This can be derived by looking at the StudyOverallStatus where low.value corresponds to the activation date and high.value corresponds to the completion date.</td></tr>
<tr><td></td><td>enrolledStudySubjectNumber</td><td>DEFINITION:

An integer specifying the quantity of study subjects enrolled in the study at the current time.

EXAMPLE(S):</td></tr>
</table>

Class	Description	
		OTHER NAME(S): NOTE(S): This can be derived by counting the number of enrolled subjects.
StudyObjective	DEFINITION: A goal that the study is aiming to achieve in terms of a scientific question to be answered by the analysis of data collected during the study. EXAMPLE(S): To extend the life of study participants by at least 3 years. To determine efficacy of Drug X dose 1, dose 2, and dose 3 as measured by the percentage of subjects experiencing headache relief. To compare overall survival in subjects with [type of cancer] who have received [prior treatment] and who are randomized to treatment with either Combination A+B or single-agent B. To select a Drug X dose for further evaluation based on comparison of the short-term antiviral activity, safety, and tolerability of different oral doses of Drug X in combination with Drug Y in HIV-1 infected therapy-naïve subjects. To compare the proportion of subjects developing a rash in subjects administered dermatological precautions (DP) versus subjects administered usual care precautions (UC) during 12 weeks of treatment of Drug X in [disease description] subjects. To obtain exploratory descriptive information on the relationship of tobacco use, alcohol use and dietary patterns on toxicity and outcomes in males and females. OTHER NAME(S): NOTE(S): StudyProtocolVersion.purposeStatement, StudyProtocolVersion.primaryPurposeTypeCode and StudyObjective may sound similar in meaning but are distinct concepts in BRIDG. StudyProtocolVersion.purposeStatement, which is an broad explanation of why a study is being conducted (e.g. determine efficacy of a drug or procedure), differs from	

<table>
<tr><th>Class</th><th>Description</th></tr>
<tr><td></td><td>StudyProtocolVersion.primaryPurposeTypeCode which is a classification of the purpose or intent of the study (e.g. Prevention, Treatment, Quality of Life), and that differs from StudyObjective, which describes in a specific and measurable way what the study hopes to accomplish (e.g. extend life of subjects at least 3 years, reduce frequency of symptoms).

<table>
<tr><th>Attribute name</th><th>Description</th></tr>
<tr><td>description</td><td>DEFINITION:

The textual representation of the study objective.

EXAMPLE(S): The objective of the analysis is to evaluate the efficacy of study treatment versus placebo. The Alzheimer's Disease Assessment Scale - Cognitive Subscale, total of 11 items [ADAS-Cog (11)] and the Video-referenced Clinician's Interview-based Impression of Change (CIBIC+) will serve as the primary efficacy instruments. Efficacy will be determined by testing for a statistically significant relationship between the change in both the ADAS-Cog (11) and CIBIC+ scores, and drug dose (0, low dose [54 mg], and high dose [81 mg]).

OTHER NAME(S):

NOTE(S): The example above makes reference to content from more than just the StudyObjective class, rather also from classes closely related to it. However, in prose presentation, this is common practice. See BRIDG GForge Tracker #30649 and #31401.</td></tr>
<tr><td>typeCode</td><td>DEFINITION:</td></tr>
</table>
</td></tr>
</table>

Class		Description
		A coded value specifying the kind of study objective. EXAMPLE(S): primary, secondary, tertiary, exploratory OTHER NAME(S): NOTE(S): The examples given are the only allowable concepts for this attribute.
StudyOverallStatus		
StudyProtocol	DEFINITION: A discrete, structured plan (that persists over time) of a formal investigation to assess the utility, impact, pharmacological, physiological, and/or psychological effects of a particular treatment, procedure, drug, device, biologic, food product, cosmetic, care plan, or subject characteristic. EXAMPLE(S): OTHER NAME(S): NOTE(S): The term "protocol" is somewhat overloaded and must be qualified to provide semantic context. Therefore the term "study protocol" was chosen to disambiguate it from other protocols. The notion of a study protocol includes (but is not limited to) the design, statistical considerations, activities to test a particular hypothesis or answer a particular question that is the basis of the study, characteristics, specifications, objective(s), background, pre-study/study/post-study portions of the plan (including the design, methodology, statistical considerations, organization). The study may be of any type that involves subjects, including prevention, therapeutic, interventional or observational. Subjects involved in the study protocol may be biological entities (human, animal, specimen, tissue, organ, etc.) or products. The study protocol is related to other supporting documents, including (but not limited to)	

Class	Description
	informed consent documents, case report forms (CRFs), regulatory and approval documentation, correlative studies, etc. (via the inherited association to DocumentVersionRelationship). The complete notion of the study protocol is represented in BRIDG by the classes StudyProtocol, StudyProtocolVersion, StudyProtocolDocument, StudyProtocolDocumentVersion, StudyExecution and all their associations. - The StudyProtocol class represents the content of the study protocol which includes characteristics and plan of the study which can be distilled into or abstracted from a version of the study protocol document and can exist even before the information is put into document form. - The StudyProtocolVersion class represents the details of the study protocol that may change over time. - The StudyProtocolDocument class represents the document form of the study protocol and is a grouping of the various study protocol document versions. - The StudyProtocolDocumentVersion class represents the document form of the study protocol version and is the details of the study protocol document that may change over time. - The StudyExecution class represents the conduct of a study based on a study protocol definition which includes the scheduled and performed activities that are subject-specific as well as study-level and site-level activities.
StudyProtocolDocumentVersion	DEFINITION: A representation of a particular edition or snapshot of a document containing a study protocol as it exists at a particular point in time. EXAMPLE(S): Version 3 of a breast cancer protocol. OTHER NAME(S): Amendment NOTE(S): The term "protocol" is somewhat overloaded and must be qualified to provide semantic context. Therefore the term "study protocol" was chosen to disambiguate it from other protocols. The notion of a study protocol includes (but is not limited to) the design, statistical

Class	Description
	considerations, activities to test a particular hypothesis or answer a particular question that is the basis of the study, characteristics, specifications, objective(s), background, pre-study/study/post-study portions of the plan (including the design, methodology, statistical considerations, organization). The study may be of any type that involves subjects, including prevention, therapeutic, interventional or observational. Subjects involved in the study protocol may be biological entities (human, animal, specimen, tissue, organ, etc.) or products. The study protocol is related to other supporting documents, including (but not limited to) informed consent documents, case report forms (CRFs), regulatory and approval documentation, correlative studies, etc. (via the inherited association to DocumentVersionRelationship). The complete notion of the study protocol is represented in BRIDG by the classes StudyProtocol, StudyProtocolVersion, StudyProtocolDocument, StudyProtocolDocumentVersion, StudyExecution and all their associations. - The StudyProtocol class represents the content of the study protocol which includes characteristics and plan of the study which can be distilled into or abstracted from a version of the study protocol document and can exist even before the information is put into document form. - The StudyProtocolVersion class represents the details of the study protocol that may change over time. - The StudyProtocolDocument class represents the document form of the study protocol and is a grouping of the various study protocol document versions. - The StudyProtocolDocumentVersion class represents the document form of the study protocol version and is the details of the study protocol document that may change over time. - The StudyExecution class represents the conduct of a study based on a study protocol definition which includes the scheduled and performed activities that are subject-specific as well as study-level and site-level activities. The term "Amendment" needs to be disambiguated since it sometimes refers to the amended version of the protocol (StudyProtocolDocumentVersion) and other times refers to the summary of changes

<table>
<tr><th>Class</th><th>Description</th></tr>
<tr><td></td><td>(AmendmentChangeSummaryVersion).

<table>
<tr><th>Attribute name</th><th>Description</th></tr>
<tr><td>publicTitle</td><td>DEFINITION:

The title of the document intended for the general population.

EXAMPLE(S):

OTHER NAME(S):

NOTE(S):</td></tr>
<tr><td>publicDescription</td><td>DEFINITION:

The textual summary of a document intended for the general population.

EXAMPLE(S):

OTHER NAME(S):

NOTE(S):</td></tr>
<tr><td>scientificDescription</td><td>DEFINITION:

The textual summary of a document that includes extended scientific or technical information.

EXAMPLE(S):

OTHER NAME(S):

NOTE(S):</td></tr>
<tr><td>text</td><td>DEFINITION:

A textual or media-based representation that is the full or comprehensive narrative or content of the document.</td></tr>
</table></td></tr>
</table>

Class	Description	
	officialTitle	EXAMPLE(S): OTHER NAME(S): NOTE(S): DEFINITION: The formal title of the document. EXAMPLE(S): OTHER NAME(S): NOTE(S): If there is only one title, use this attribute.
	date	DEFINITION: The date (and time) on which the document is versioned. EXAMPLE(S): OTHER NAME(S): NOTE(S):
	uniformResourceLocator	DEFINITION: A complete or local reference to a website, ftp, file path or other location from which the document contents can be retrieved. EXAMPLE(S): OTHER NAME(S): NOTE(S): Derived from DocumentVersion.text.ED.reference.

Class	Description	
		Local references should only be used when communicating between systems capable of resolving the local reference. In a Regulatory Product Submission (RPS) message, this identifies the file (with a Uniform Resource Identifier (URI)), which is part of the documentation. A URI is a compact string of characters used to identify or name a resource. The main purpose of this identification is to enable interaction with representations of the resource over a network, typically the World Wide Web, using specific protocols. URIs are defined in schemes defining a specific syntax and associated protocols.
StudyProtocolVersion	DEFINITION: A plan at a particular point in time for a formal investigation to assess the utility, impact, pharmacological, physiological, and/or psychological effects of a particular treatment, procedure, drug, device, biologic, food product, cosmetic, care plan, or subject characteristic. EXAMPLE(S): OTHER NAME(S): NOTE(S): The term "protocol" is somewhat overloaded and must be qualified to provide semantic context. Therefore the term "study	

Class	Description
	protocol" was chosen to disambiguate it from other protocols. The notion of a study protocol includes (but is not limited to) the design, statistical considerations, activities to test a particular hypothesis or answer a particular question that is the basis of the study, characteristics, specifications, objective(s), background, pre-study/study/post-study portions of the plan (including the design, methodology, statistical considerations, organization). The study may be of any type that involves subjects, including prevention, therapeutic, interventional or observational. Subjects involved in the study protocol may be biological entities (human, animal, specimen, tissue, organ, etc.) or products. The study protocol is related to other supporting documents, including (but not limited to) informed consent documents, case report forms (CRFs), regulatory and approval documentation, correlative studies, etc. (via the inherited association to DocumentVersionRelationship). The complete notion of the study protocol is represented in BRIDG by the classes StudyProtocol, StudyProtocolVersion, StudyProtocolDocument, StudyProtocolDocumentVersion, StudyExecution and all their associations. - The StudyProtocol class represents the content of the study protocol which includes characteristics and plan of the study which can be distilled into or abstracted from a version of the study protocol document and can exist even before the information is put into document form. - The StudyProtocolVersion class represents the details of the study protocol that may change over time. - The StudyProtocolDocument class represents the document form of the study protocol and is a grouping of the various study protocol document versions. - The StudyProtocolDocumentVersion class represents the document form of the study protocol version and is the details of the study protocol document that may change over time. - The StudyExecution class represents the conduct of a study based on a study protocol definition which includes the scheduled and performed activities that are subject-specific as well as study-level and site-level activities.

<table>
<tr><th>Class</th><th colspan="2">Description</th></tr>
<tr><td></td><td>Attribute name</td><td>Description</td></tr>
<tr><td></td><td>targetAccrualNumberRange</td><td>DEFINITION:

An integer falling within minimum and maximum bounds that specifies how many study subjects are to be accrued for the study.

EXAMPLE(S):

OTHER NAME(S):

NOTE(S): A typical target accrual number (always assumed to be a minimum target) would be targetAccrualNumberRange.IVL<INT>.low, a maximum target accrual would be targetAccrualNumberRange.IVL<INT>.high.</td></tr>
<tr><td></td><td>acronym</td><td>DEFINITION:

The non-unique initials or abbreviated name used for identification of the study.

EXAMPLE(S): WHI for Women's Health Initiative

OTHER NAME(S):

NOTE(S):</td></tr>
<tr><td>StudyRecruitmentStatus</td><td colspan="2">DEFINITION:

Status of finding and enrolling appropriate study subjects (those selected on the basis of the protocol's inclusion/exclusion criteria) into a study.

<table>
<tr><td>Attribute name</td><td>Description</td></tr>
<tr><td>code</td><td>DEFINITION:

A coded value specifying the phase in</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Class</th><th colspan="2">Description</th></tr>
<tr><td></td><td>date</td><td>the lifecycle of recruitment for the study.

EXAMPLE(S): Not yet recruiting; recruiting; enrolling by invitation; active, not recruiting; completed; suspended; terminated; withdrawn.
DEFINITION:

The date (and time) on which the recruitment status is assigned.

EXAMPLE(S):

OTHER NAME(S):

NOTE(S):</td></tr>
<tr><td>StudySite</td><td colspan="2">DEFINITION:

A facility in which study activities are conducted.

EXAMPLE(S): The site where the study subject encounter occurs, or the site of the Investigator.

OTHER NAME(S):

NOTE(S):</td></tr>
<tr><td>StudySiteInvestigator</td><td colspan="2">DEFINITION:

A researcher at a study site who oversees multiple aspects of the study at a site, including protocol submission for IRB approval, participant recruitment, informed consent, data collection, and analysis.

EXAMPLE(S):

OTHER NAME(S):

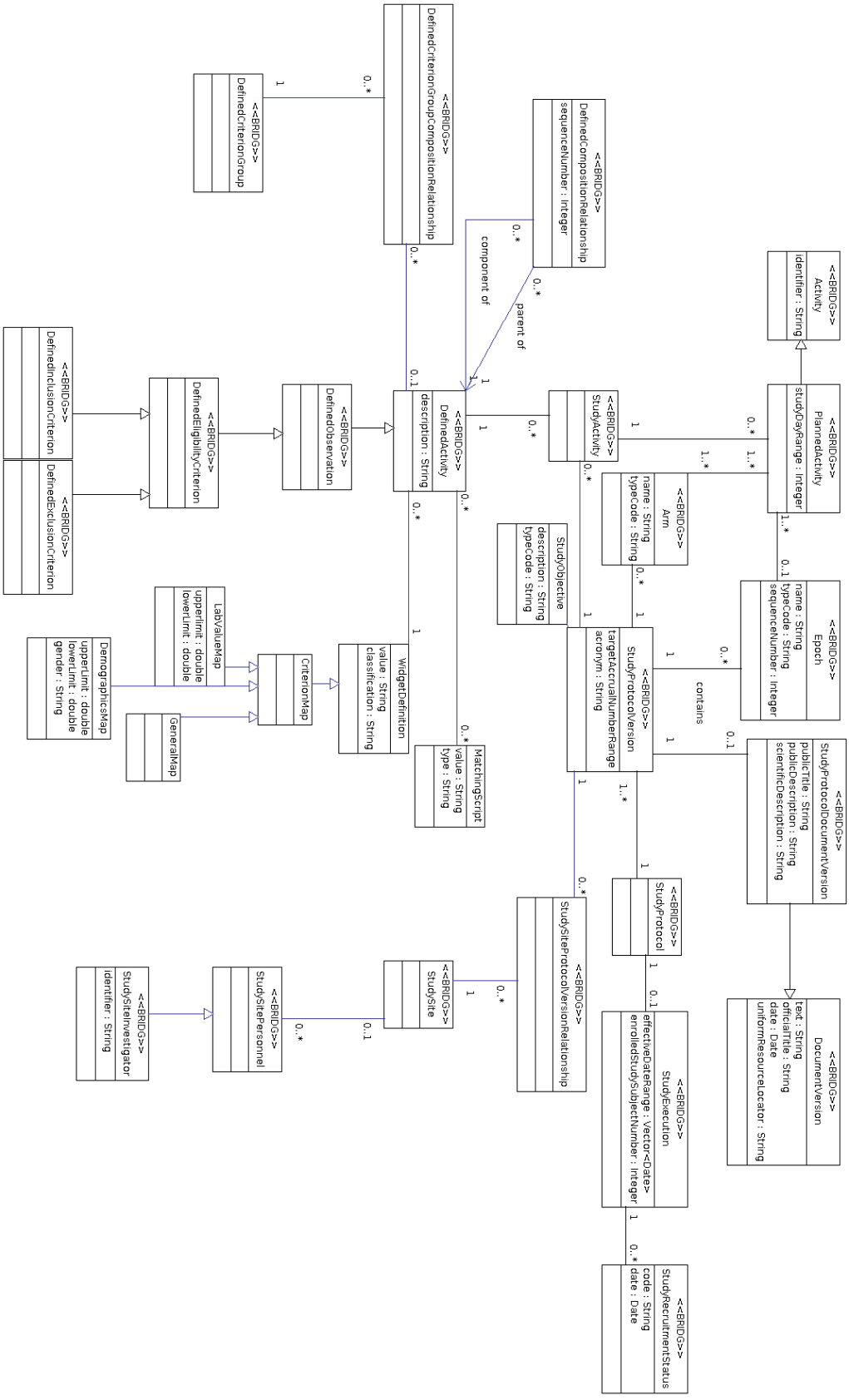
NOTE(S):

<table>
<tr><th>Attribute name</th><th>Description</th></tr>
<tr><td>identifier</td><td>DEFINITION:</td></tr>
</table>
</td></tr>
</table>

Class	Description
	A unique symbol that establishes identity of the study site investigator.
StudySitePersonnel	EFINITION: A person who performs a particular role within the context of a specific study site. EXAMPLE(S): Study Site Investigator, Study Site Research Coordinator OTHER NAME(S):
StudySitePersonnel	DEFINITION: A person who performs a particular role within the context of a specific study site. EXAMPLE(S): Study Site Investigator, Study Site Research Coordinator OTHER NAME(S):
StudySiteProtocolVersionRelationship	DEFINITION: Specifies the link between a study site and a version of the study protocol used or available for use at that site. EXAMPLE(S): OTHER NAME(S): NOTE(S): Even if a study site's IRB has not reviewed the study protocol version, if there is a new version for the study protocol, then there is the potential for a relationship between the site and the version. The dateRange is specified only if the version is approved for this site by the IRB and activated at the site. Retroactive approval means that the dateRange does not have to be on or after the IRB approval date.
WidgetDefinition	WidgetDefinition contains widget type information such that the demonstrator can display the correct widget for the criterion. Attribute name | Description

Class		Description
	value **classification**	textual representation of the widgetdefinition DEFINITION: Code representing the class of the widget (e.g. lab test, demographics)

13 Appendix B – Trial Meta-data Model



14 Appendix C – Common Data Model