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

D2.4 Initial System Architecture and Implementation Status
D2.3 Initial Report on the INTEGRATE Security Framework

Due date of deliverable: 01-31-2012
Actual submission date: 03-19-2012

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

Responsible WP: WP2 Architecture and integration

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
Brecht Claerhout	Custodix	brecht.claerhout@custodix.com
Kristof De Schepper	Custodix	kristof.deschepper@custodix.com
Jelle Van den Driessche	Custodix	jelle.vandendriessche@custodix.com
Sergio Paraíso	UPM	sparaiso@infomed.dia.fi.upm.es
George Manikis	FORTH	gmanikis@gmail.com
Ioannis Karatzanis	FORTH	karatzan@ics.forth.gr
Jasper Van Leeuwen	Philips	jasper.van.leeuwen@philips.com
Anca Bucur	Philips	anca.bucur@philips.com
Philippe Hennebert	IJB	philippe.hennebert@bordet.be

0.2 Documents history

Document version #	Date	Change
V0.1	10/06/2011	Initial draft
V0.2	10/20/2011	Revision
V0.3	12/05/2011	Revision
V0.4	12/21/2011	Parsing document to word template
V0.5	01/13/2012	Revision
V0.6	01/23/2012	Revision
V0.7	01/31/2012	Final internal review
V1.0	03/09/2012	Approved Version to be submitted to EU

0.3 Document data

Keywords	Architecture, implementation status, viewpoints, views
Editor Address data	Name: Kristof De Schepper Partner: Custodix Address: Korterijksesteenweg 214 b3 B-9830 Sint-Martens-Latem, Belgium Phone: +32 9 210 78 90 Fax: +32 9 211 09 99 E-mail: kristof.deschepper@custodix.com
Delivery date	01/31/2012

0.4 Distribution list

Date	Issue	E-mailer
January 2012		fp7-integrate@listas.fi.upm.es

1 Abstract

This document gives the initial description of the architecture, security framework and initial demonstrator of the INTEGRATE platform.

The first part of the document contains the initial architectural description of the INTEGRATE platform. It is structured based on the viewpoint model and conforms to the requirements of IEEE Std. 1471:2000 (also ISO/IEC 42010:2007). In the current state of the document it was decided to give a relative high level functional description of the different views. A more detailed description will be given in the following iterations of this document. Four views were identified: a functional view, information view, deployment view and data protection view. The functional view supplies an overview of each of the major components extracted from the use cases described in D1.4 (and scenarios from D2.1). The information view contains a description of how the data and meta-data is structured within the INTEGRATE platform. The deployment view gives insight to how the components of the INTEGRATE platform will be physically structured. The components protecting data access and patient privacy are described in the data protection view.

To comply with legal requirements concerning patient privacy and data protection defined in D1.3, the INTEGRATE platform requires an advanced security framework. In part two of this document, the requirements (authentication, de-identification, consent, ...) and S&T challenges (contextual attributes, vocabulary mapping, ...) of this security framework are described. It will include privacy enhancing, authentication, authorisation and audit mechanisms.

The last part gives a general overview of the demonstrator that will be shown at the first review of the INTEGRATE project. The aim is to demonstrate large parts of the INTEGRATE platform, be it with limited functionality. Two major blocks were selected from the architecture: the molecular testing and the analytical tools. The goals and approaches to implement these blocks were explicitly described.

Table of Contents

2 Introduction

This document is a merger of deliverable "D2.3 Initial report on the INTEGRATE security framework" and deliverable "D2.4 Initial system architecture and implementation status" (reason discussed further). It is divided into three main distinct parts:

- The INTEGRATE architecture description
- The security framework
- The implementation status

2.1 PART I – Architecture Description

As part of the software development design phase an architectural description is created that defines the main architectural components that will be used in the INTEGRATE platform.

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.

[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*

In this document three views were identified to be useful for the INTEGRATE architecture description: the functional, information and deployment view; also one cross-cutting view was identified: the data protection view. The definition of these views can be found in the corresponding underlying sections.
Finally, it should be noted that this document gives a snapshot of an evolving architecture as the project progresses (cf. deliverable iterations at month 18 and 24).

2.2 PART II – Security Framework

The second part of this document gives a first high-level description of the security framework. Originally this "Initial report on the INTEGRATE security framework" was planned to be the topic of a separate deliverable D2.3. However, because of the considerable overlap of the security framework with the architecture description, it was decided to merge D2.3 into D2.4 Initial system architecture and implementation status. While part one includes first parts of a solution (security view as part of the architecture description), the second part rather describes the requirements for the security framework and details the scientific and technological challenges to be dealt with.

2.3 PART III – Implementation Status

In the first year of the project, implementation is (as planned) limited. Hence instead of giving a (non-informative) implementation status of the various software components, the last part of the deliverable discusses the first year demonstrator that will be shown at the first annual project review.

2.4 The INTEGRATE Platform

From the description of work, the main goal of INTEGRATE platform is to offer solutions to clinical investigators, researchers and the pharmaceutical industry in order to improve collaboration, molecular testing for patient enrolment in trials, querying trial data, sharing of data and knowledge and building and sharing of predictive models for response to therapies. In deliverables D1.1, D1.2 and D1.3 this goal is formalized in the form of user requirements and scenarios. With these deliverables as main input, use cases were defined in D1.4, written from a technical point of view. As stated before, D1.4 only contains a snapshot of the design process of INTEGRATE, the final version will be specified in a later iteration (D1.5). The use cases are used as main input for this deliverable. In different sections of this document there is an explicit link to these use cases. Figure 1 gives an overview of the interactions between the different deliverables.

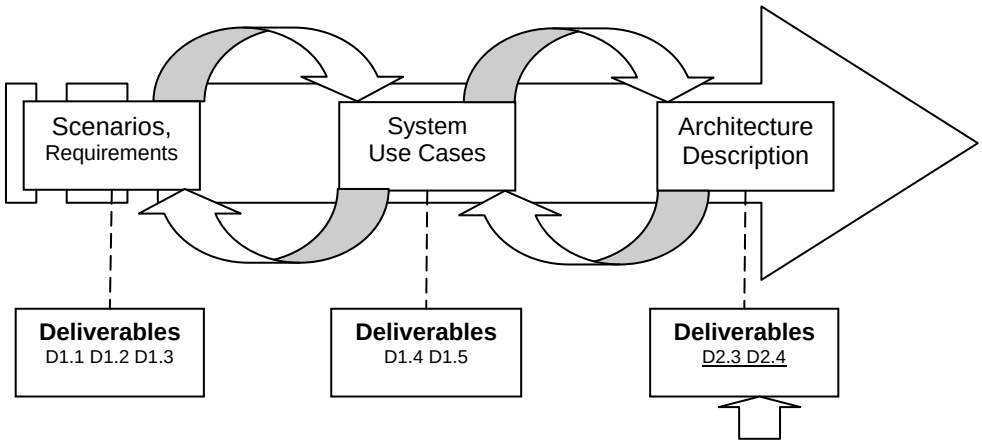


Figure 1: Deliverable interaction

3 (PART I) System Stakeholders and Concerns

3.1 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, Table 1 provides a short explanation. 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 D7.4[3]). Future exploitation (reflected in "usefulness", "scalability", "genericness" ...) is a concern of all system-owners and is not further discussed in this version of the document. If it becomes necessary (as the exploitation plan evolves) to describe the architecture from the exploitation perspective (refining existing stakeholders or introducing new stakeholders with their specific demands), this will be elaborated in a later iteration of the architecture description.

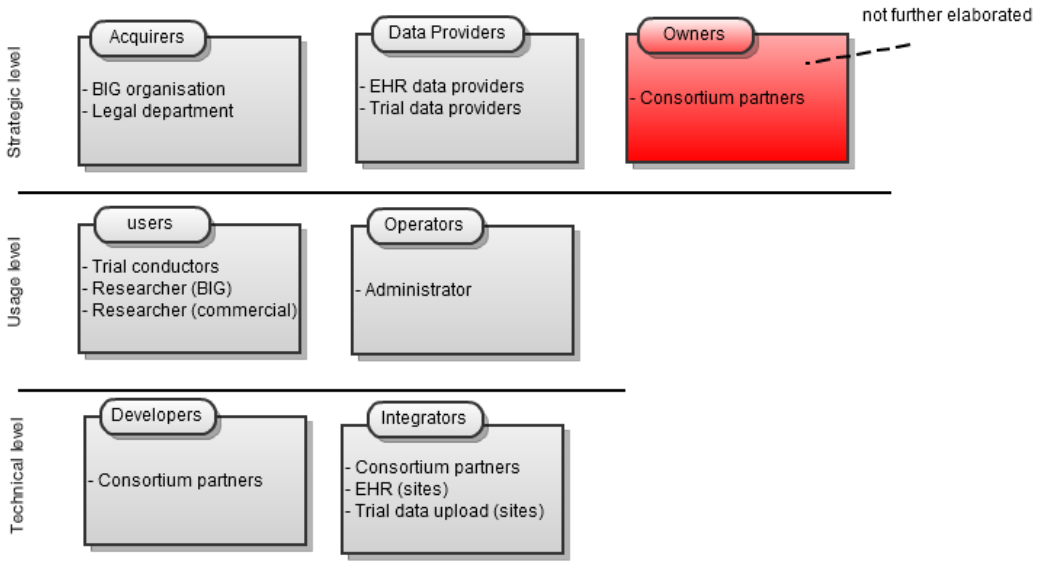


Figure 2: Overview of the identified stakeholders

[3] INTEGRATE Deliverable 7.4 - Initial exploitation plan

Role	Stakeholder	Remarks
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.
Acquirers	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.
Data Providers	Trial Data Providers	Site managers at the local BIG sites which plan to provide trial data in the INTEGRATE research environment.
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: an oncologist or a person that works under the responsibility of an oncologist - 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
Users	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.
Users	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.
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.
Integrators	EHR integrators	Technicians in charge of linking the EHR at the

[4] INTEGRATE Deliverable 1.2 - Definition of relevant user scenarios based on input from the users

	(@sites)	local sites with the INTEGRATE platform, i.e. making the EHR site compliant with the INTEGRATE interfaces.
Integrators	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.

Table 1 Overview of the identified stakeholders.

As illustrated on Figure 2, the stakeholders can be grouped according to their main concerns. This document, as WP2 deliverable, mainly focuses on viewpoints dealing with concerns at the technical level. In a later iteration these concerns can be extended with other types of concerns.

3.2 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.

ID	Stakeholder	Concern
CAC-001	BIG legal department	The INTEGRATE framework must comply with the legal, ethical and security requirements defined in INTEGRATE legal framework (deliverable D1.3).
CAC-002	BIG organisation	*Currently no direct concerns identified for BIG organisation.*

Table 2: Acquires concerns

ID	Stakeholder	Concern
CUS-001	Trial Conductors	All the defined requirements concerning the molecular screening and central pathology review functionality are available in the INTEGRATE platform.
CUS-002	Researcher (BIG)	All the defined requirements concerning the trial data querying and analytical tools functionality are available in the INTEGRATE platform.
CUS-003	Researcher (commercial)	All the defined requirements concerning the trial data querying and analytical tools functionality are available in the INTEGRATE platform.
CUS-004	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-005	Researcher (BIG)	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	Researcher (commercial)	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.

Table 3: User concerns

ID	Stakeholder	Concern
COP-001	Administrators	As an end-user, the administrator needs an idea of the functionality of the platform to assess the scope of the administration tools.

Table 4: Administrator concerns

ID	Stakeholder	Concern
CDE-001	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-002	Consortium partners	Security components of the INTEGRATE platform provide generic interfaces, so security can be integrated in the INTEGRATE services in a relative straightforward way.
CDE-003	Consortium partners	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.

Table 5: Developer concerns

ID	Stakeholder	Concern
CIN-001	Consortium partners	Connecting the separately developed software blocks of the INTEGRATE platform to one integrated system (complying interfaces)
CIN-002	EHR integrators (@sites)	Comply the EHR datawarehouses interfaces (situated on the sites) to the requirements of the INTEGRATE platform.
CIN-003	CDMS integrators (@sites)	Compliance with the trial data upload process (situated on the sites) to the interfaces of the INTEGRATE platform.

Table 6: Integrators concerns

ID	Stakeholder	Concern
CDP-001	EHR Data Providers	Offering EHR data access to the INTEGRATE platform that complies with the local regulations of the providing site.
CDP-002	Trial Data Providers	Providing trial data to the INTEGRATE platform that complies with the local regulations of the providing site.

Table 7: Data provider concerns

4 Architecture Viewpoints

Currently we are not working with formalised architecture viewpoints, but we may return to this later once we have well established what kind of description and models are appropriate for the different views.

5 Overview

5.1 10000 Feet View

The 10000 feet view of the architecture aims to give the reader a general idea about the technical setup of the INTEGRATE platform by offering a high-level view containing the main architectural building blocks and interactions between them.

This high level view is split up into different parts according to the defined scenarios (see D1.2). At the end of the section an overall (integrated) view is provided. This split has not only been made for clarity of explanation, but is guiding the overall research and development of the INTEGRATE system. While eventually all components will be integrated, the scenarios describe relatively distinct chunks of functionality, which can for a large part be dealt with separately. Furthermore, these scenarios have been prioritized and are not all elaborated at the same time. Consequently, not all requirements are fully specified in this first year iteration of the project (in any case, user needs do evolve over time).

At the same time, there is also a "natural" separation of scenarios according to the legal domain they belong. The INTEGRATE environment can be separated into two domains with respect to data protection: the "trial execution" and "research" domain. The former involves all scenarios in which users are directly involved with individual patients; the situation is similar to interactions in the care environment. There is a direct patient relationship (at recruitment), data access needs to be nominative (data of individuals needs to be evaluated). In the research domain, interest shifts from the individual level to the cohort level. There is no relation with the patient anymore, focus is on data analysis. Data can therefore be handled anonymously. Both domains are governed by a different (legal) rule set.

In order to ensure final integration, it is important to keep track of the overlap between the different views (basically the shared functional blocks). This is in fact the main responsibility of the architectural work, which is documented in this (iterated) architectural description deliverable.

5.1.1 Molecular Testing

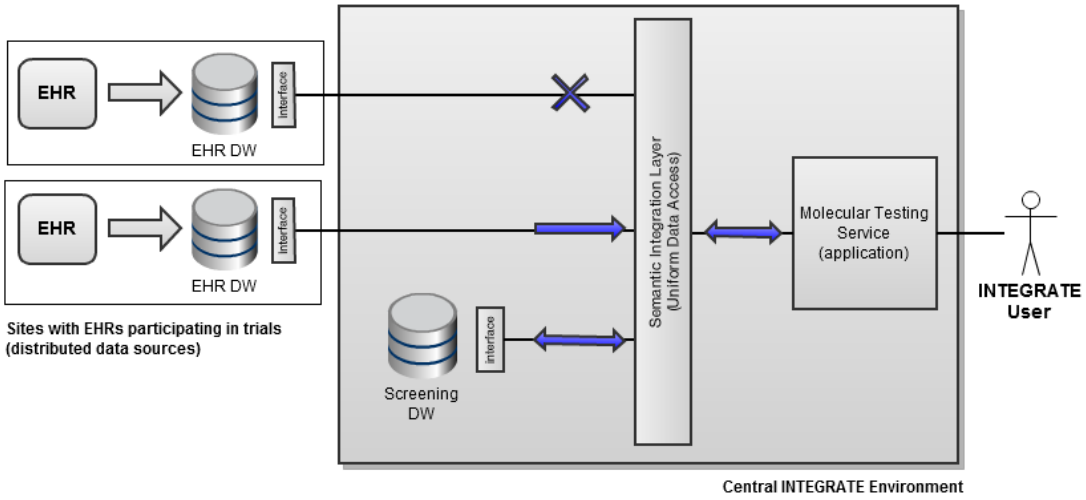


Figure 3: 10000 feet view molecular testing

The molecular testing scenario describes an interactive process for checking patient eligibility to a particular trial or set of trials. The automated screening (eligibility testing) application relies on two types of data sources for retrieving the necessary data about a patient. Those are:

- **Screening datawarehouse**
  The screening datawarehouse is a central datawarehouse (DW) (can be one or more instances) that stores all information about a patient which is specific to the screening process. It not only serves as a source of data for the screening application, it is also used for storing data entered in that application during the screening process.
- **EHR datawarehouses**
  The EHR datawarehouses are local exports of (part of) the EHR in trial participating sites. They are used in a typical "secondary use" setting: to retrieve parameters important in the screening process which are already present in the EHR. In the current approach, EHR data relevant for the screening process will eventually be copied into the screening DW. So that practically speaking, after a pre-filling step this DW can be used as the single local source for eligibility criteria matching.

A big difference between the screening DW and EHR DW's is that the latter are linked and local to the site in which a patient is screened. This is particularly important for access control, patient identity management and service discovery (of the correct EHR DW). The screening process will always involve the same screening DW, but a different EHR DW depending on the site to which the user of the application belongs (patient EHR data will logically reside in the site where they are screened).

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 CDS. 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. Similarly applications can be developed in a generic way, based solely on the common information model.

5.1.2 Trial Data Querying

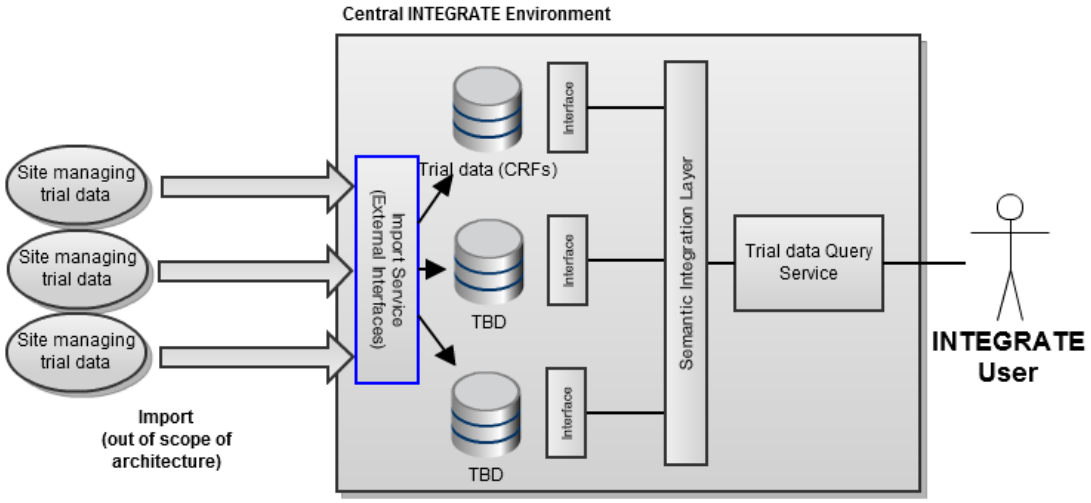


Figure 4: 10000 feet view trial data querying

The trial data query service will provide functionality for querying all data present in the research domain of the INTEGRATE platform in a uniform way. In the same way as explained in the section above, the semantic layer abstracts the data sources for the upper layer applications. The data warehouses are physically centralised and logically belong to the same security domain (which is not the case in the molecular screening scenario explained above, cf. EHR's).

Data residing in the research domain of the INTEGRATE platform is all de-identified. It is uploaded from the trial sites which act as data management sites (i.e. host the EDC environment) for the various trials (which can be a different site for different trials). A generic import service will guarantee that all data uploaded into the INTEGRATE research domain is de-identified conform the data protection framework. It is not only used for "external" import (from the trial sites), but also for import from the INTEGRATE trial domain.

5.1.3 Overall INTEGRATE Architecture

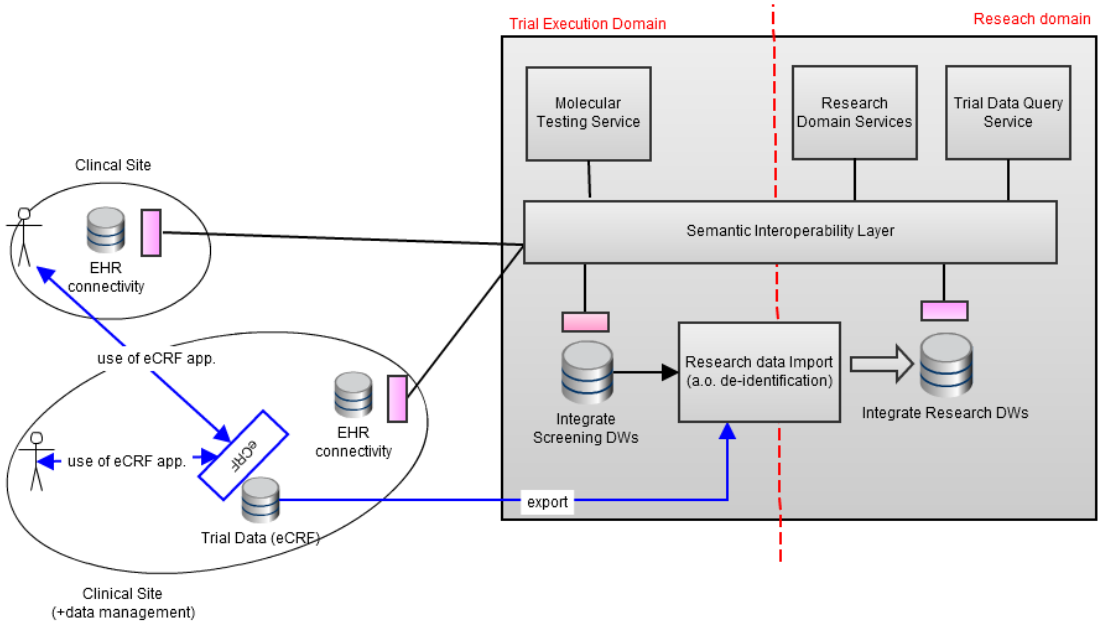


Figure 5: 10000 feet view overall

Figure 5 shows a first high level view of the whole INTEGRATE system. On the right the INTEGRATE environment is shown with its two logical domains: the trial execution domain and the research domain.

The logical separation between the two domains will be guaranteed by the security infrastructure (see below). Data is allowed to be imported from the trial conduct domain to the research domain only through an import service which includes de-identification.

The semantic interoperability layer abstracts the different data sources, presenting a common information model to the upper application layers. Data sources include the central INTEGRATE data warehouses as described earlier and the distributed EHR data exports at the participating trial sites.

6 Functional View

6.1 Molecular Testing

6.1.1 Screening Process

6.1.1.1 Introduction

Most of the functional related concerns that stakeholders have regarding the molecular testing scenario (see D1.2) are addressed in the screening process view. For this it offers a set of general architectural building blocks. Starting from the use cases (see D1.4), five 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.

In a later iteration of this document, the current presented functional components will be expanded to a higher level of detail, offering a more detailed description of the different components.

Concerns addressed (see paragraph 3.2)

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.

6.1.1.2 Diagram

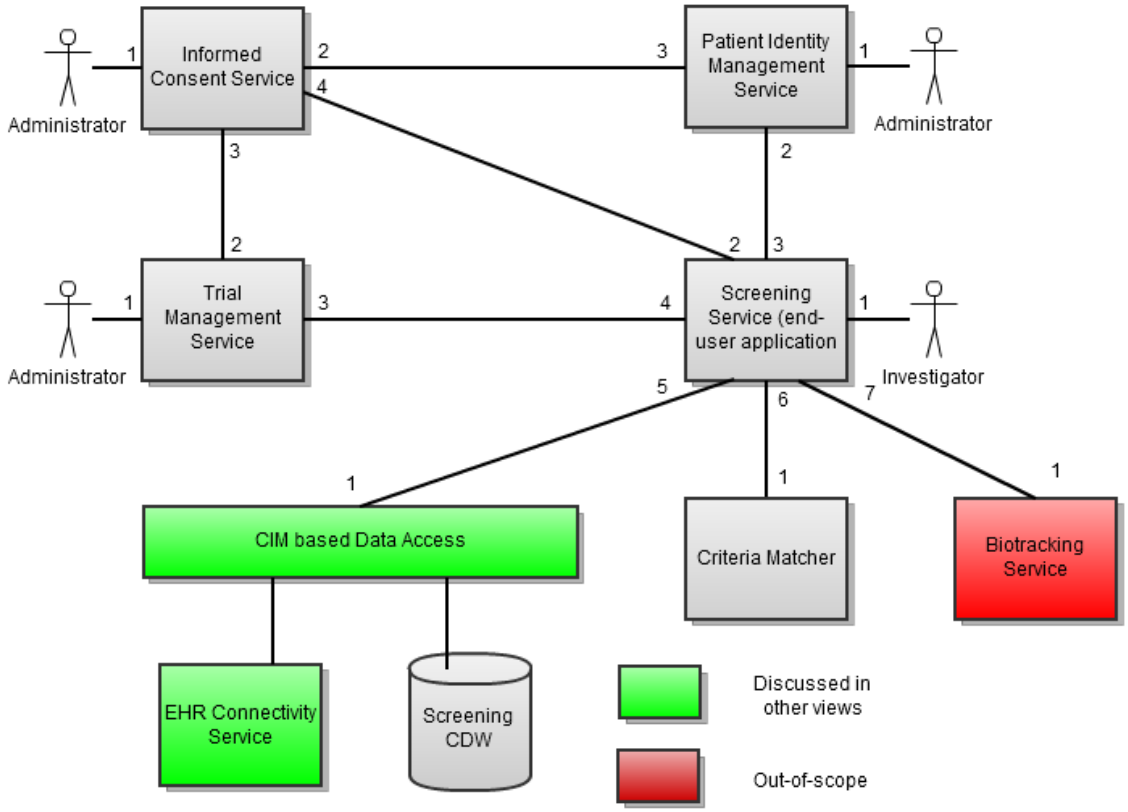


Figure 6: Screening process functional view

6.1.1.3 Components and Interfaces

Screening Service (end-user application)

The screening service is the main driver component in the screening process view. It is an end-user application that integrates and connects the different services that are needed to meet the specified requirements of the molecular testing scenario. An end-user (investigator) that wants to check a patient's eligibility for a trial will interact with the system through this component. This interaction is based on a provided advanced graphical interface that is part of the screening service component.

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 screening service. This front-end exposes following functionality: • A step by step process for checking the patient's eligibility to be enrolled in a 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 CIM-based data access*
6	*No interface exposed to the criteria matcher*

7	*No interface exposed to the biotracking service*

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	The informed consent service exposes following functionality to the trial management service: • Create, activate, list and edit informed consent configurations for a trial
4	The informed consent service exposes following functionality to the screening service: • Register a new signed informed consent for a particular patient and a selected trial • Verify if an informed consent is registered for a particular purpose for a patient and a selected trial • List the signed informed consent for a patient and a selected trial

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 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

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 • Create CRF for a trial • Register informed consent configurations 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 screening service: • List all the registered trials in the platform • Get detailed information about a selected trial

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 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 screening service:

	• Match criteria defined in a trial with provided screening data (coming from the datawarehouses)

CIM (Common Information Model) based Data Access

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 screening service needs a link with the semantic layer, by means of the CIM based data access component. This layer (worked out in paragraph 6.3) 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.***

Interface	Description
1	The CIM based data access exposes following functionality to the screening service: • Retrieval of screening data from the INTEGRATE datawarehouse(s) • Retrieval of EHR data from the site(s) datawarehouse(s)

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 screening service: • Register, track and analyse biological samples of a patient

6.1.2 EHR Connectivity

6.1.2.1 Introduction

The EHR connectivity focuses on a specific requirement of the molecular testing scenario, namely the semantic link between the EHR datawarehouse(s) and the INTEGRATE platform. Because this view has specific functionality, it was decided to separate it from the general screening process (see paragraph 6.1.1). The main intention of the EHR connectivity view is providing the link between the semantic layer on the INTEGRATE platform and the EHR datawarehouse interfaces on the sites. This enables the investigator in the molecular testing scenario to find information about a selected patient. From the use cases (D1.4) we currently differentiate one component in the first iteration. A more detailed description will be provided in a later iteration as soon as there is a better view on the specific requirements of this section.

Concerns addressed (see paragraph 3.2)

CUS-001, **CAC-001** *(1)*, **CIN-001**, **CIN-002**, **CDP-001**, **CDE-001**

(1) The link between the EHR datawarehouses and the INTEGRATE platform must comply the INTEGRATE legal framework.

6.1.2.2 Diagram

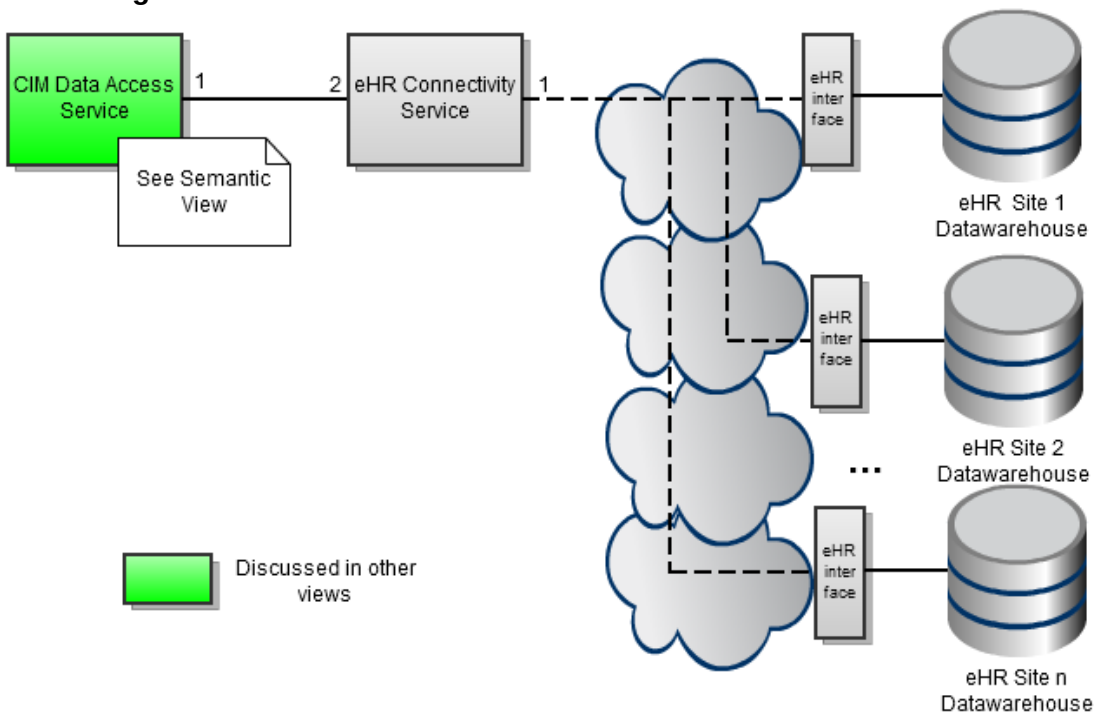


Figure 7: EHR connectivity functional view

6.1.2.3 Components and Interfaces

EHR Connectivity Service

The molecular testing scenario demands that there is a semantic link to (part of) the EHR data of a patient. This data can be found in an EHR datawarehouse that is provided by a site. Because of the distributed nature of the EHR data, a service should be provided that can locate EHR data of a particular patient. This together with patient querying functionality is provided by the EHR connectivity service.

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

Interface	Description
1	*No interface exposed to the EHR datawarehouse*(s)
2	The EHR connectivity service exposes following functionality to the CIM data access service: • Provide the EHR data from a given patient

CIM Data Access Service

The component that triggers the EHR data look-up is the CIM data access service. This service (which is worked out in paragraph 6.3) provides functionality to query the datasets of the EHR 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.***

Interface	Description
1	*No interface exposed to EHR connectivity service*

6.2 Trial Data Querying

6.2.1 Introduction

The trial data querying scenario (see D1.2) demands that research should be able to generate and execute queries on the INTEGRATE datawarehouses in the research domain in order to retrieve datasets of research information. The functional concerns of this scenario are addressed in the trial data querying view. As described before, this querying of the datawarehouses happens in collaboration with the semantic layer (see paragraph 6.3). In the current iteration of this document we only differentiate one main component from the use cases (D1.4[5]). A more detailed description will be provided as soon as there is a better view on the specific requirements of this section.

Concerns addressed (see paragraph 3.2)

CUS-002, CUS-003, CDE-001

[5] INTEGRATE Deliverable 1.4 - Consolidation of the user needs, use-case development and requirements analysis

6.2.2 Diagram

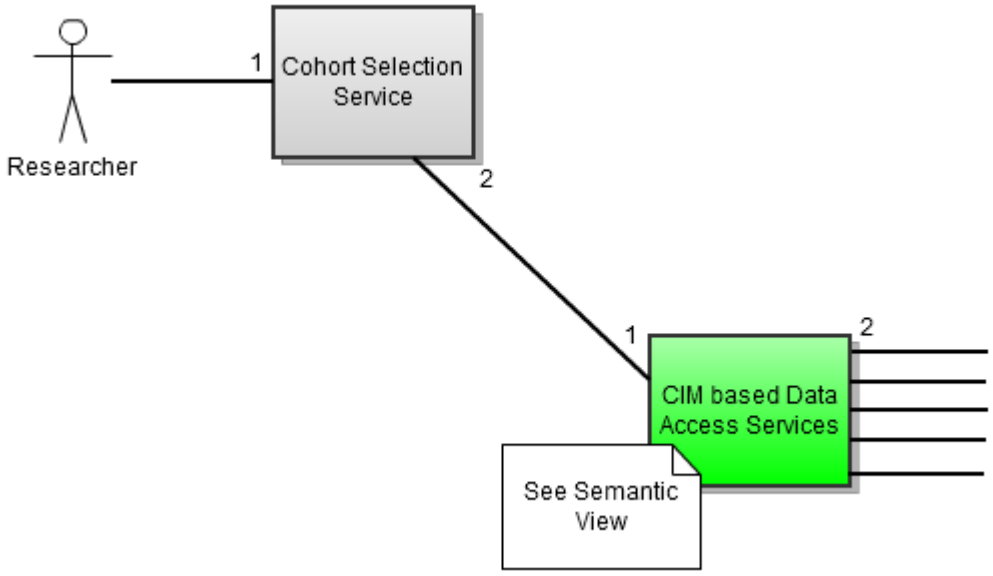


Figure 8: Trial data querying functional view

6.2.3 Components Interfaces

6.2.3.1 Cohort Selection Service

The cohort selection service provides functionality for researchers to generate queries in a flexible way and execute these on a particular dataset. This dataset can be either the whole dataset available in the INTEGRATE datawarehouses or subsets coming from previously executed query(s). The querying always happens through the semantic layer. Generated queries are stored and can be re-used by the researcher at a later point in time.

Related to use cases: **UC.3**, **UC.4**, **UC.5**, **UC.6**, **UC.24**, **UC.25**

Component	Interface
1	A researcher interacts with the front-end part (GUI based) of the cohort selection service. This front-end exposes following functionality: • Define new queries on a data set • Store a defined query • Retrieve stored queries • Send a query to the semantic layer for execution • Store result sets of a query on a data set • Retrieve stored result sets • Download the data linked with the result sets
2	*No interface exposed to the CIM based data access service*

6.2.3.2 CIM based Access Services

In the trial data querying scenario, a researcher needs to be able to query data stored in the INTEGRATE datawarehouses situated in the research domain. For this the screening service needs a link with the semantic layer, by means of the CIM based data access component. This layer (which is worked out in paragraph 6.3) provides functionality to query the datasets of the INTEGRATE datawarehouses. It abstracts the underlying data sources for the upper cohort selection service and presents data to applications according to a single integrated data model.

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

Component	Interface
1	The CIM based data access services exposes following functionality to the cohort selection service: • Execute queries on the INTEGRATE datawarehouses and return the matching result sets
2	*No interface exposed to the datawarehouses*

6.3 Semantic Layer

6.3.1 Introduction

The semantic interoperability layer section is focused on the common information model (CIM) interactions with data sources, retrieving data and related components.

These processes involve the following components:

- Common Data Model (CDM): Common schema of patient information stored at the different data warehouses of the INTEGRATE platform.
- Core Dataset: Medical vocabulary used within the INTEGRATE platform. This vocabulary standardizes the concepts used within the INTEGRATE platform, including relationships to perform semantically-aware queries.
- Reasoner: Responsible for inferring knowledge about the core dataset and the CDM
- Terminology binding: Provide information related to the location of concepts of the core dataset within the CDM.
- Query engine: Generate and executes queries on data warehouses. It is responsible of interacting with the data warehouse to retrieve semantically-aware information.

The following components have being also considered as part of the semantic layer, although physically they could be located at each participating institution.

- Data warehouse: Physical storage of the INTEGRATE platform, where EHR and data compliant with the CDM is stored.
- ETL: Tools that extract information from data sources, transform this information to CDM structure and load it into data warehouses.

Concerns addressed (see paragraph 3.2)

CUS-002, CUS-003, CDE-001

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

6.3.2 Diagram

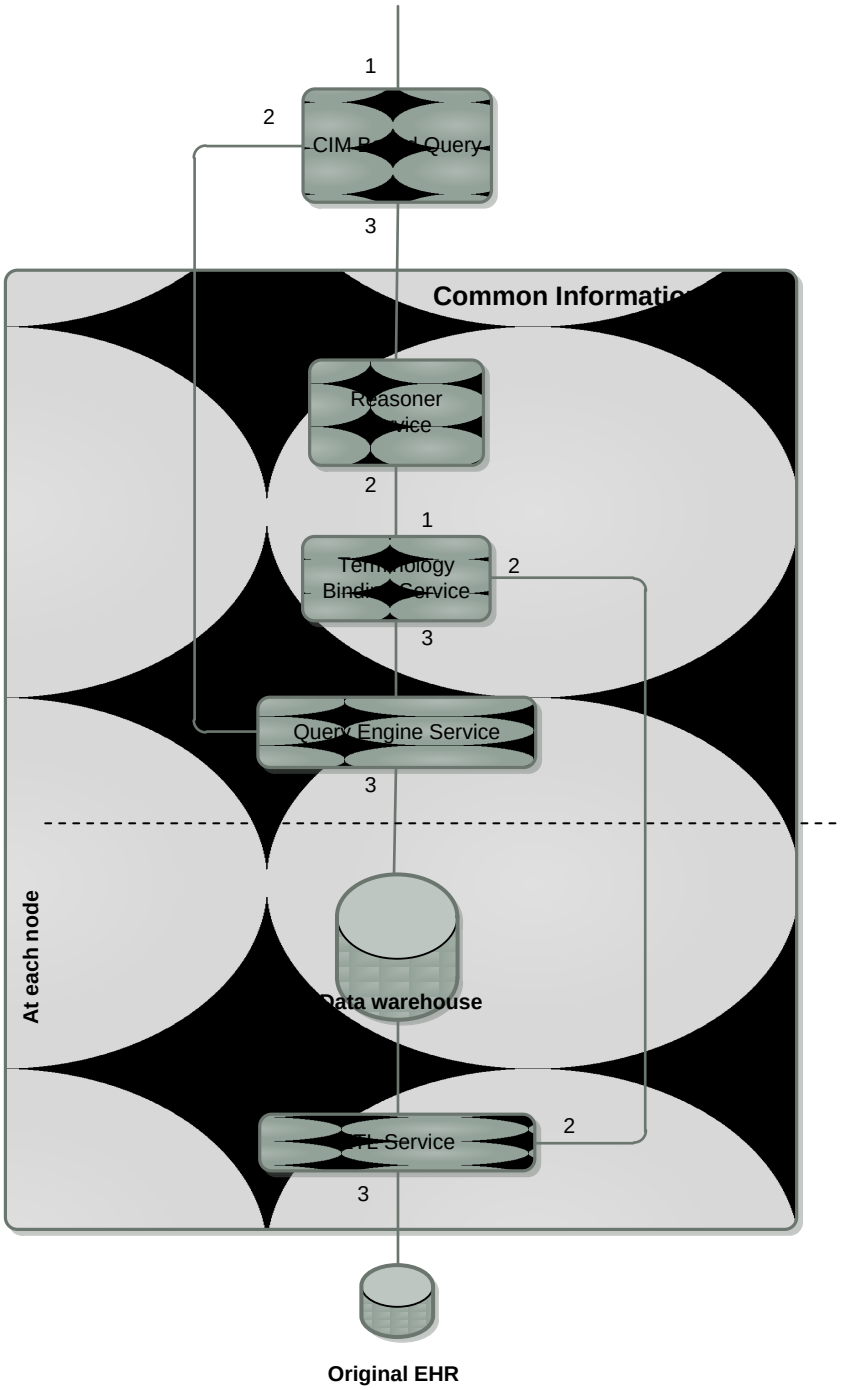


Figure 9: Common Information Model

The diagram shows the different services that interact in the functional section of the semantic layer. The next sections describe in detail each service and relations among components.

6.3.3 Components and Interfaces

In this section all the components of Figure 9 are described.

6.3.3.1 CIM Data Access Service

The CIM data access service exposes the functionality of the semantic layer to other components. It sends queries to the reasoner service. Once these queries are transformed and executed, the CIM data access service is responsible for returning the query result to the requesting service.

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

Interface	Description
1	The CIM data access service exposes the following functionality to the requesting INTEGRATE services: • Execute queries on the INTEGRATE data warehouses and return the matching result sets
2	The CIM data access service exposes the following functionality to the query engine service: • Receive results of the launched query, these results will be sent back to the requesting service
3	*No interface exposed to the reasoner service*

6.3.3.2 Reasoner Service

The reasoner service is applied to infer knowledge about the Core Dataset and the CDM. It should classify the vocabulary in order to retrieve information about relationships of the medical terminology (from concepts at the CIM data access service) and how it is represented in the data model used for the data warehouse.

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

Interface	Description
1	The reasoner service exposes the following functionality to the CIM data access service: • The reasoner infers knowledge about concepts contained in the query.
2	*No interface exposed to the terminology binding service*

6.3.3.3 Terminology binding service

The terminology binding service is responsible for indicating elements of the data warehouse schema where core dataset concepts are stored. For that purpose, the service receives a list of concepts given by the core dataset reasoned and returns the location of the concepts at the data warehouse.

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

Interface	Description
1	*No interface exposed to the reasoner service*
2	*No interface exposed to the ETL service*
3	*No interface exposed to the query engine service*

6.3.3.4 Query Engine Service

The Data Warehouse stores the information of the different data sources using the CDM structure. This information is loaded by the ETL mapping service and is coded with concepts of the Core Dataset.

The query engine service is responsible for modifying and executing the queries against the Data Warehouse. The modification employs the list of concepts given by the reasoner service and terminology binding service.

Once the query is executed, the query engine service returns the results to the CIM data access service.

Related to uses cases: **UC.SEM.5**

Interface	Description
1	*No interface exposed to the terminology binding service*
2	*No interface exposed to the CIM data access service*
3	*No interface exposed to the data warehouse*

6.3.3.5 ETL Service

The ETL service is responsible for the transformation of the original data into the data warehouse. Three other components are related to this:

- Data sources
- Terminology binding
- CDM

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

Interface	Description
1	Data Warehouse - In this component, the patient data related to the clinical trial (CT) are stored.
2	Terminology binding - This component transforms and maps the

	information with the common data model to a standardized common vocabulary.
3	Data sources - In this process, the ETL service extracts the information contained in the EHR data sources, they are stored in the Data Warehouse.

6.4 Central Pathology Review

6.4.1 Introduction

The functional concerns that the stakeholders have regarding the central pathology review scenario (see D1.2) are addressed in this view. Here a central review platform is designed (using the UC.CR.* use cases from D1.4 as input) that provides the necessary functionality to review images by multiple reviewers and manage (and log) this procedure. Next to this core functionality it will also offer extra collaborative services between the reviewers such as messaging and scheduling. The components that are part of and interact with the central review platform are described in the next subsections.

Concerns addressed (see paragraph 3.2)

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.

6.4.2 Diagram

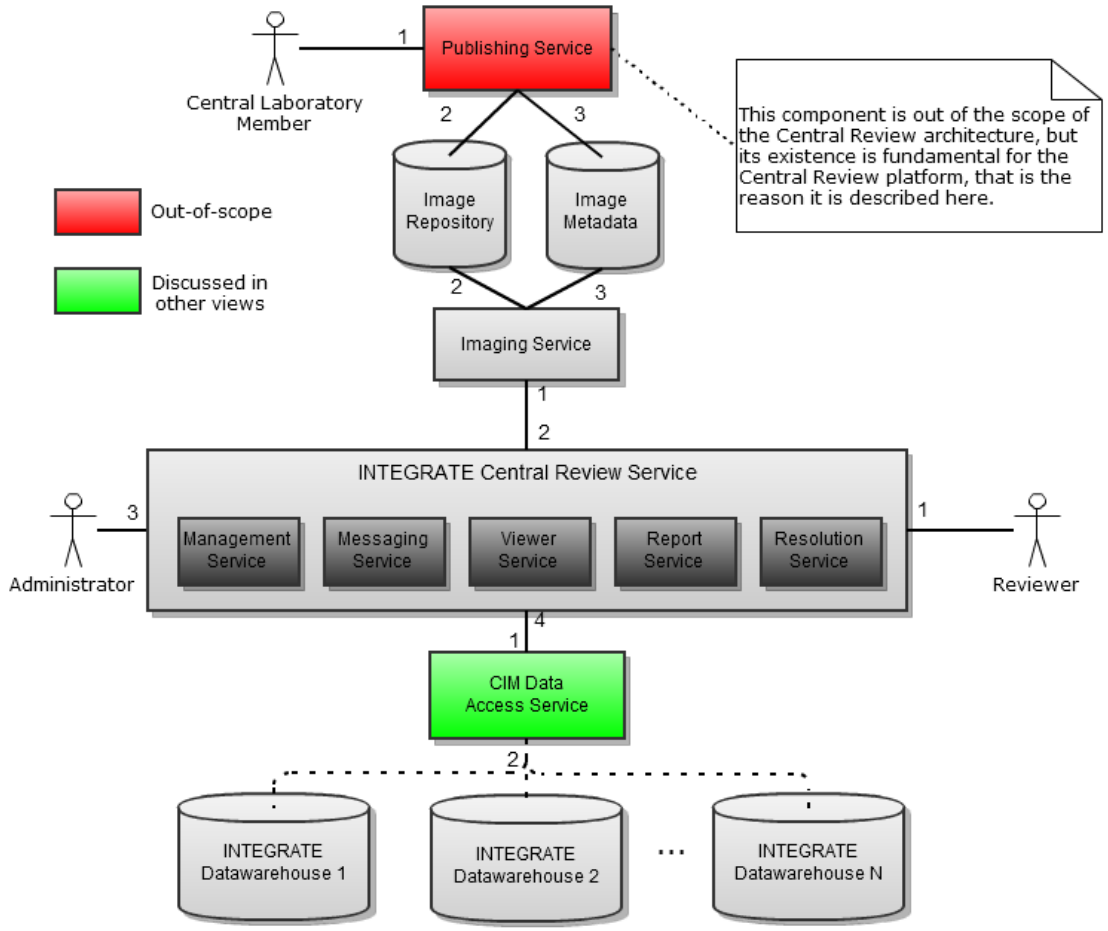


Figure 10: Central Pathology Review

6.4.3 Components and Interfaces

6.4.3.1 Publishing Service

This component is a GUI-based or portal framework with directions on how a reviewer or a developer can store a new medical image to the images repository.

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 datawarehouse of INTEGRATE, which will be the input for the central review platform. The publishing service is described in another scenario -and not in the central review scenario- and therefore will be defined and implemented by the architecture defined there. Moreover, the "image repository" and the corresponding" Image meta-data" could also be a part of the "INTEGRATE data repository" and the "INTEGRATE meta-data repository". The separation has been made for reasons of better understanding.*

Related use cases: -

Interface	Description
1	The reviewer/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.
2	The publishing service exposes following functionality to the images repository: • Uploading a new image (i.e. pathology raw image, DICOM,…) to the repository.
3	The publishing service exposes following functionality to the image meta-data: • Uploads meta-data information related to the image. (1) Type of image (i.e. pathology, DICOM, …). (2) Preview thumbnail of the image. (3) Annotations (i.e. an XML structure). (4) Etc.

6.4.3.2 Imaging Service

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 requesting an image and the meta-information of it and returns the data to the 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 in the image repository (read only) • Providing access to the metadata in the image meta-data repository
2	*No interface exposed to the image repository*
3	*No interface exposed to the image meta-data repository*

6.4.3.3 Management Service

The management service provides all the necessary functionality and tools in order to define collaboration groups, in order to create and schedule new central 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 which are pending to be reviewed. • View the history of the reviewed images. • View incoming messages and compose new ones. • View notifications from the platform. • Etc.
2	*No interface exposed to the imaging service*
3	The management service exposes to the administrators a GUI from which they can: • Create/edit a collaboration group • Create/edit a task • View a listing of images which are pending to be reviewed. • View the history of the reviewed images. • View incoming messages and compose new ones. • View notifications from the platform. • Etc.
4	*No interface exposed to the data/meta-data access service*

6.4.3.4 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 send and receive messages, to receive notifications or to communicate through an iterative process with other reviewers in order to 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 data/meta-data access service*

6.4.3.5 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, annotate and probably analyze 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	The viewer service pushes the annotation information (if any) of the image which is under review, to the "data/meta-data access service".

6.4.3.6 Report Service

The report service is just 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.

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.
2	*No interface exposed to the imaging service*
3	The administrators can see the reports which are filled by the reviewers.
4	The report service pushes the data entered by the user into the "data/meta-data access service", from where they are then stored to the appropriate meta repositories.

6.4.3.7 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 which are under review (the annotations and the data of the corresponding report). The resolution service exposes to the reviewer the following functionality: 1. It merges information from all reviews in a common portal page, side by side, in such a way that the reviewers can easily compare the measurements. Next to each

	measurement there will be a verification switch. 2. If the reviewer reading the form has not been able to find any inconsistent measurement, then he/she marks the image as "Accepted". If the image is marked as "Accepted" by all of the reviewers, the resolution service sets the corresponding status flag and stores it to the database. 3. If there is a disagreement from the part of the reviewer who is reading the form, then he/she marks the measurement/value which is in question and automatically the image is marked to be "For further investigation" (a status flag is set and stored to the database). In this situation where there is not an agreement between all the reviewers, the resolution process tries to address the issue using a message exchange mechanism which starts a conversation among the reviewers until they reach to an agreement and then marks the image as "Accepted".
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 image being reviewed (how it is characterized by each reviewer) • See an overview regarding the image, which merges the information from all the reviews in a simple and common page (as described in functionality 1 of the interface 1 of the resolution service).
4	The resolution service pushes the status of the image under review to the data/meta-data access service (= the status flag, described in Interface 1).

6.4.3.8 Data/Meta-Data Access Service

The data/metadata access service 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 data/meta-data access service exposes to the INTEGRATE central review service the functionality to download and upload data to the data and meta-data repositories of INTEGRATE.
2	*No interface exposed to the INTEGRATE meta-data repository*
3	*No interface exposed to the INTEGRATE data repository*

6.5 Analytical Tools

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 any type of clinical, genomic and imaging data, stored in the central INTEGRATE datawarehouse(s). Further it exposes functionality for connecting to the tools & model 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 sharing of predictive models is 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 (see paragraph 3.2)

CUS-002, CUS-003, CDE-001, CIN-003, CDP-002

6.5.2 Diagram

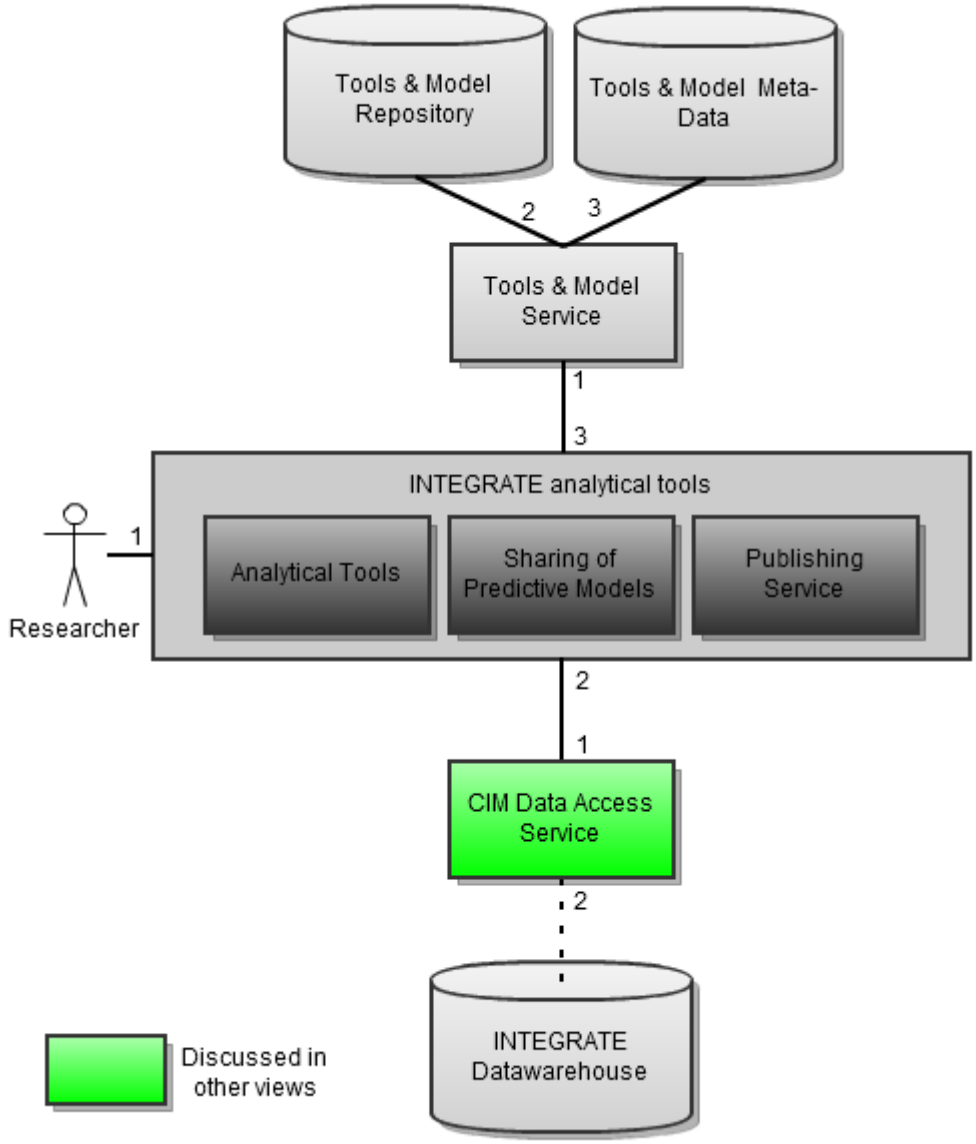


Figure 11: Analytical Tools

6.5.3 Components and Interfaces

6.5.3.1 Analytical Tools

The analytical tools communicate via the analysis platform with the INTEGRATE datawarehouses, the INTEGRATE meta-data repository, the tools & model repository and the tools & model meta-data repository for the selection, retrieval and management of the data and tools. Moreover, it provides a user-friendly framework for the visualization, 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 visualization of an analysis report. • All the analysis reports are available for download via the platform.
2	*No interface exposed to the data access service*
3	*No interface exposed to the tools & model service*

6.5.3.2 Sharing of Predictive models

As a part of the INTEGRATE analysis platform, the sharing of predictive models 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 visualization of an analysis report. • All the analysis reports are available for download via the platform.
2	*No interface exposed to the data access service*
3	*No interface exposed to the tools & model service*

6.5.3.3 Publishing Service

This component is a GUI based or portal framework with directions on how a Researcher or Developer can store a new tool to the tools & model repository.

Related use cases: **UC.IAT_PM.5**

Interface	Description
1	The researcher interacts with the publishing service (a GUI based front-end) of the INTEGRATE analysis platform. The front-end exposes following functionality: • Detailed directions for uploading a new tool or model.
2	*No interface exposed to the data access service*
3	The publishing service exposes following functionality to the tools & model service: • Uploading the tool or model to the repository. • Providing metadata information related to the functionality of the tool or model. (1) Type of analysis (i.e. survival analysis). (2) Type of data used (i.e. 2 clinical + *.txt files). (3) Memory constraints (i.e. high computational cost). (4) File format (i.e. R script, Matlab executable, etc.). (5) Model authorship (6) Etc.

6.5.3.4 Data Access Service

The data access service provides functionality for querying data from the INTEGRATE datawarehouse. It interacts with the INTEGRATE datawarehouse and the meta-data repository and retrieves the desired data for statistical analysis or predictive modeling.

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

Interface	Description
1	The data access 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. • Storage of an analysis report given by either a statistical tool or a predictive model.
2	The data access service exposes following functionality to the INTEGRATE datawarehouse: • Executing a query and requesting access to retrieve and store data.

6.5.3.5 Tools and Model Service

The INTEGRATE analysis platform, on behalf of the analytical tools and the sharing of predictive models, 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	The tools and model service exposes following functionality to the tools & model repository: • Giving access to the data repository, enabling a tool to be executed in a batch mode and retrieving the analysis report.
3	The tools and model service exposes following functionality to the tools & model meta-data: • Executing a query using the metadata information of an implemented analysis. In case of a statistical tool, this includes: (1) Type of analysis (i.e. survival analysis). (2) Type of data used (i.e. 2 clinical + *.txt files). (3) Output files (i.e. 2 *.eps + 1 *.csv, etc.). (4) Memory constraints (i.e. high computational cost). (5) File format (i.e. R script, Matlab executable, etc.). (6) Model authorship (7) Etc. • Executing a query using the metadata information of an implemented analysis. In case of a predictive model, this includes: (1) Comprises precise information about the predictor and predicted variables. (2) The mathematical model. (3) The training and validation data sets. (4) The estimated model accuracy metrics (e.g. AUROC) (5) Type of data used (i.e. 2 clinical + *.txt files). (6) Output files (e.g. 2 *.eps + 1 *.csv, etc.). (7) Memory constraints (e.g. high computational cost). (8) File format (e.g. R script, Matlab executable, etc.). (9) Model authorship (10) Etc.

7 Information View

7.1 Meta-data Models of the INTEGRATE Repository

7.1.1 Introduction

In this chapter the content of the model, meta-data and annotation repositories will be specified. Input for this chapter comprises D1.2, D1.4 (Use Cases) and the preliminary modelling performed in D4.2[6]. In specifying the meta-data models, we aim at leveraging BRIDG[7] 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[8]) and clinical practice standards (such as HL7v3[9]).

7.1.2 Diagram

For modelling the metadata, we leverage BRIDG by reusing classes and relationships (from version 3.0.3). 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 <<demonstrator>> stereotype indicates that the defined classes are relevant for the 1st INTEGRATE demonstrator. The BRIDG definitions are used for the BRIDG classes in the class descriptions section.
The meta-data model is described on a per-topic (informed consent, molecular testing, meta analysis) basis after the commonly used classes have been introduced.

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

7.1.2.1 Static Data Structure Model

Common classes overview

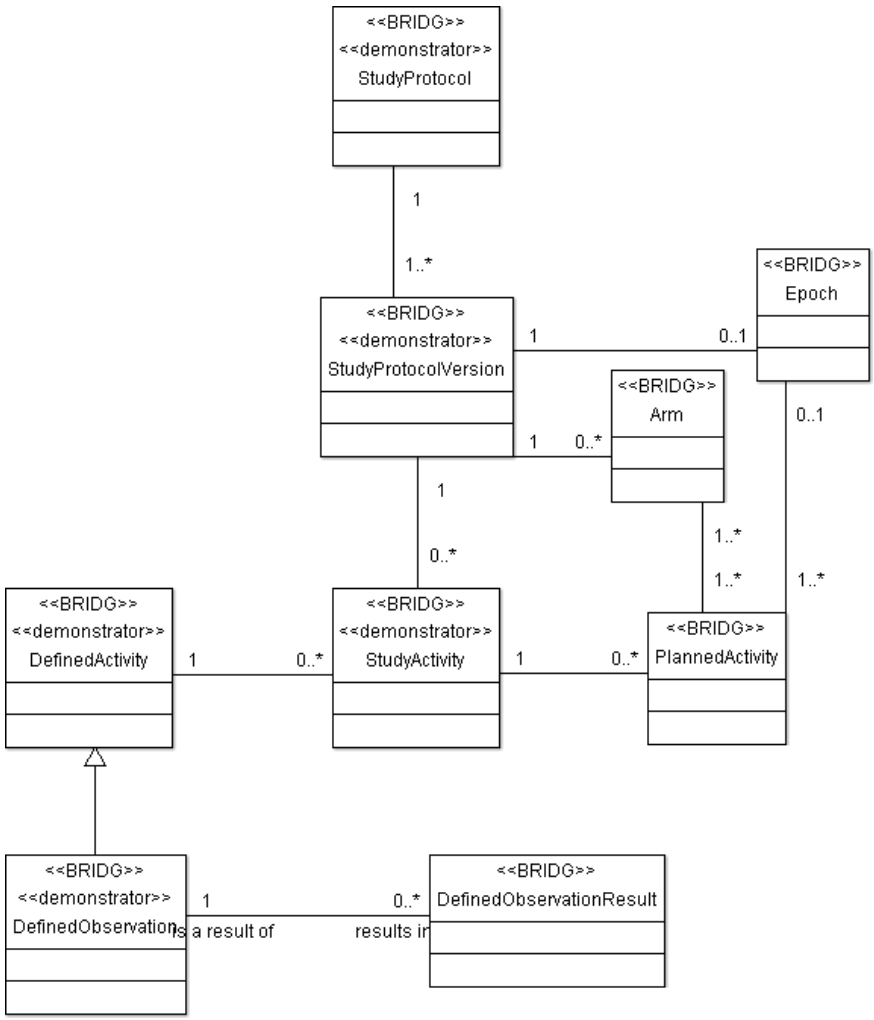


Figure 12: Common classes static model

Figure 12 shows the classes that are often referred to in the more topic specific diagrams below. StudyProtocol represents (the protocol of) the clinical trial (or study) at hand. The StudyProtocol is versioned by the StudyProtocolVersion (different versions can be active at different sites, changes to the study protocol can be recorded). The StudyActivity denotes the intention to use a defined activity in the trial. DefinedActivities defines the “kind” of activities that are possible (not necessarily directly connected to the StudyProtocol at hand). The PlannedActivity shows all activities which are actually planned in (the version of) the StudyProtocol.

DefinedObservation are the data/information gathering activities about one or more aspects of a study subject's or experimental unit's physiologic or psychologic state, the possible findings are described in the DefinedObservationResults.

The arm denotes 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 and is typically equivalent to a treatment group in a parallel design trial. 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. The intended path through which the subject progresses in a trial 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. The epoch represents a state within a study such that subjects in separate arms within that state are comparable (e.g. the different phases). Each epoch serves a purpose in the trial as a whole, typically exposing the subject to a treatment or preparing them for a treatment, or gathering post-treatment data.

Informed Consent

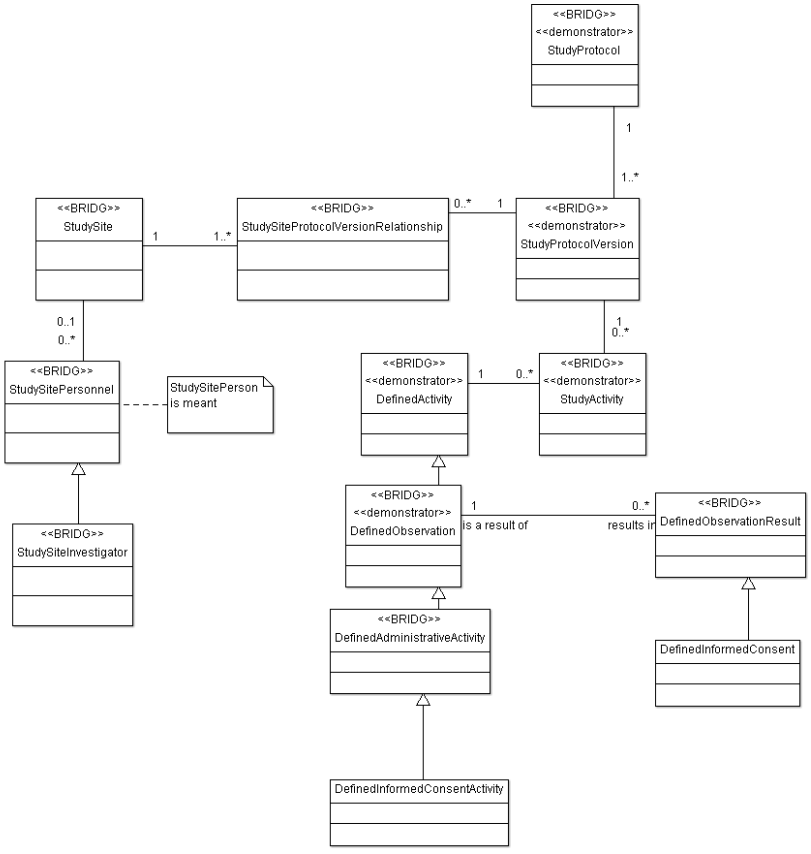


Figure 13: Informed Consent static model

The informed consent model describes the relevant meta-data related to informed consent (hence the name). Any actual informed consent details (as signed by a patient) will be captured in a (clinical) data warehouse, containing a reference to the informed consent template in the meta-data repository.

Use Case ID	UC.IC.1
Use case name	Registration of Patient Informed Consent
Brief description	Informed consent by a patient is registered on the INTEGRATE system by an investigator
Relevant steps	

3. A selectable list, containing all active informed consent configurations (in which the investigator has sufficient access rights), is displayed. The investigator selects the configuration that contains the informed consent form that the patient will sign or has signed. This page provides two options “download IC form” and “register IC”.	
Description	
Use case excerpt	**Meta-data model**
active informed consent configurations	Each clinical trial (StudyProtocol) may have different versions (StudyProtocolVersion).For each StudySite, the StudySiteProtocolVersionRelationship specifies what StudyProtocolVersion is active. Each StudyProtocolVersion may have DefinedActivities . DefinedObservations describe the possible observations (much like a template or a type) and is associated with the possible results of that observation (definedObservationResult) . The DefinedInformedConcentActivity results in DefinedInformedConsent. This implies that the actual informed consent templates for a (version of a) clinical trial can be found in the DefinedInformedConsent.
the investigator has sufficient access rights	Each StudySiteInvestigator can access at least and at most all StudyProtocolVersions that are available for use at his/her StudySite

Use Case ID	UC.IC.3
Use case name	List overview of informed consents (IC) registered for a patient
Brief description	Shows a list of all the signed informed consents belonging to a particular patient.
Relevant steps	
5. A new list is shown with all the signed informed consents of the selected patient that the investigator is allowed to see according to his access rights. 6. The investigator can now see details of each of the signed informed consents, like upload date, signed digital document, ... These details depend on the informed consent data model.	
Description	
Use case excerpt	**Meta-data model**
all the signed informed consents of the selected patient that the investigator is allowed to see according to his access rights	Each StudySiteInvestigator can access at least and at most all StudyProtocolVersions that are available for use at his/her StudySite
details of each of the signed informed consents, like upload date, signed digital document, ...	Generic details are captured in the informed consent template (DefinedInformedConsent). Patient specific details will be captured elsewhere. The DefinedInformedConsent will however specify what patient specific details are required.

Use Case ID	UC.IC.5

Use case name	Create an Informed Consent Configuration
Brief description	A new informed consent configuration needs to be added to the INTEGRATE platform.
Relevant steps	
2. On this portal page the administrator selects “create informed consent configuration”. 3. The portal redirects the administrator to a new page, displaying a form page that needs to be filled in by the administrator to successfully add a new informed consent configuration. a. The exact content of the form will be determined by the model of the informed consent that will be worked out . 5. The administrator presses a “save” button on the form page, this creates a new informed consent configuration. Each informed consent will get a unique ID in the platform. The new informed consent is NOT activated. This means that it is not possible to register patients who have signed it yet. The IC must first be activated	
Description	
Use case excerpt	**Meta-data model**
Displaying a form page that needs to be filled in by the administrator to successfully add a new informed consent configuration. The exact content of the form will be determined by the model of the informed consent that will be worked out	A new informed consent configuration will result in a new DefinedInformedConsent description.
The administrator presses a “save” button on the form page, this creates a new informed consent configuration. Each informed consent will get a unique ID in the platform. The new informed consent is NOT activated. This means that it is not possible to register patients who have signed it yet. The IC must first be activated	DefinedInformedConsents can be linked to a StudyProtocolVersion via the DefinedInformedConsentActivity. Addition of DefinedInformedConsents will result in a new StudyProtocolVersion (which is initially not active e.g. used in a StudySiteProtocolVersionRelationship).

Molecular Testing

Figure 14 shows the clinical trial meta-data required to determine the eligibility of a patient for selected trials.

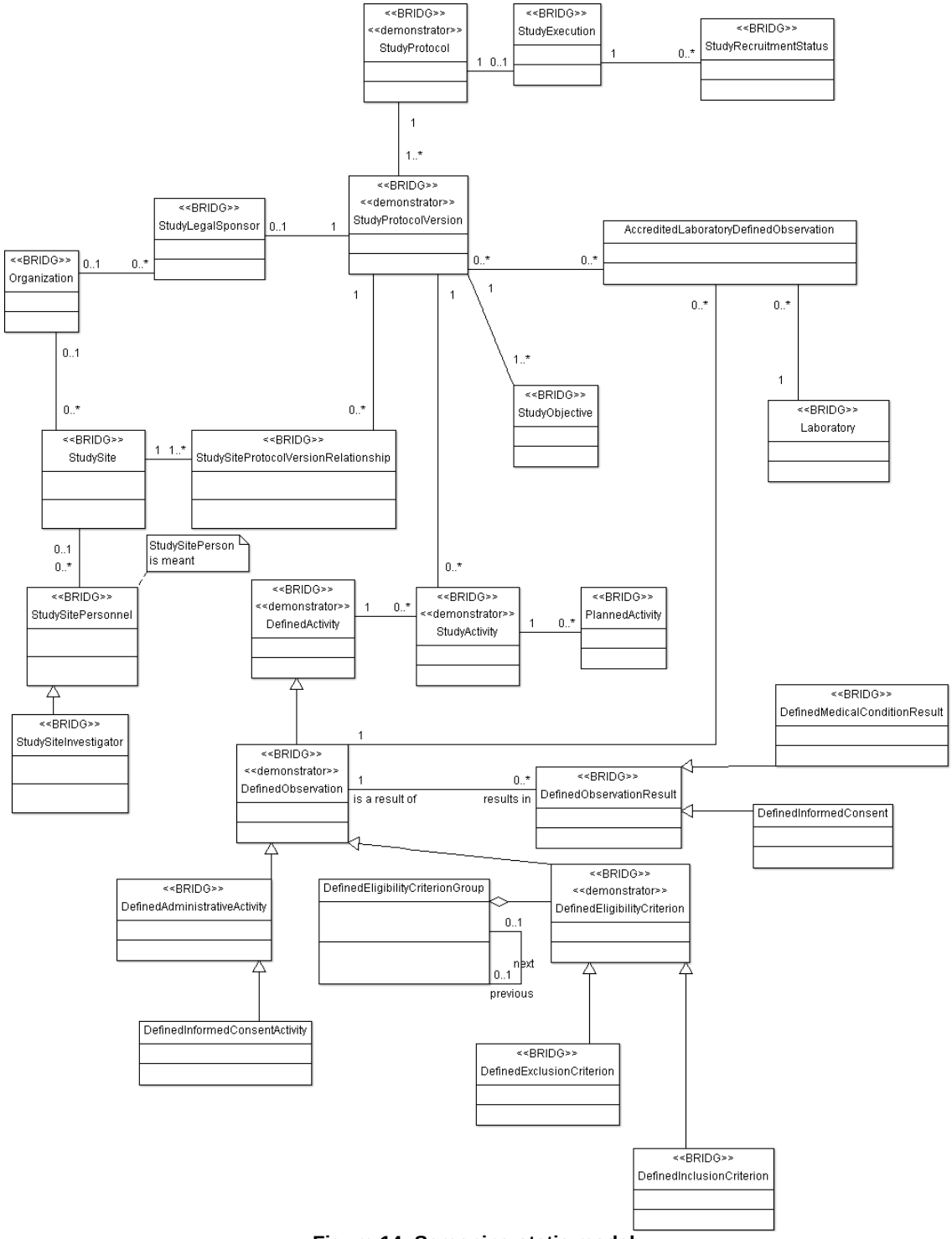


Figure 14: Screening static model

Use Case ID	UC.1
Use case name	Patient Trial Screening
Brief description	A patient is screened for inclusion in a registered trial
Relevant steps	

4. After a patient is selected, a new page is displayed where the investigator sees which Informed consents are required to continue to the next screening steps.
- The informed consents that are already entered in the INTEGRATE platform, are marked as fulfilled.
- The informed consents that are missing, need to be registered on the INTEGRATE platform (UC.IC.1, UC.IC.3)
- As long as the necessary informed consents are not present (UC.IC.2), the next screening steps cannot be executed.

5. After all necessary informed consents for the patient are submitted to the platform, the investigator selects, on a new page, one or more preferred trial(s) from the received list of trials (UC.TRIALMGT.3) he is allowed to access.
- The investigator only can view and select the trial(s) in which he participates (investigator is linked with a hospital that is linked with trials)
- Which trial(s) are selected is the responsibility of the investigator.

7. The investigator is shown a CRF page containing fields of the required trial eligibility criteria for the selected trial (received from UC.TRIALMGT.4)
- Some of these fields can be already automatically filled in (but still editable for the investigator) by:
  - Using the available patient information subsets of the EHR link with local hospital
  - Using general eligibility criteria that have been stored on the platform for the patient in a previous session.

8. The missing criteria are filled in by the investigator (UC.22).

11. The investigator is presented a page containing information about the required biological samples and molecular testing for the chosen trial.
- Some of this information are molecular criteria that should be fulfilled for the selected trial

Description	
Use case excerpt	**Meta-data model**
the investigator sees which Informed consents are required	The StudySitePersonel has access to all (active) StudyProtocolVersions which are available at the StudyPersonel's StudySite. The required DefinedInformedConsents for these StudyProtocolVersions can subsequently be accessed.
the investigator selects on a new page one or more preferred trial(s) from the received list of trials (UC.TRIALMGT.3), he is allowed to access	The StudySiteInvestigator has access to all (active) StudyProtocolVersions which are available at the StudyPersonel's StudySite
shown a CRF page containing fields of the required trial eligibility criteria for the selected trial	The set of DefinedEligibilityCriterion represent the eligibility criteria for the selected trial
Some of these fields can be already automatically filled in (but still editable for the investigator) by: • Using the available	The DefinedEligibilityCriterion will encapsulate the INTEGRATE specific mechanism of how to obtain the required data from the INTEGRATE data warehouse / EHR.

patient information subsets of the EHR link with local hospital • Using eligibility criteria that have been stored on the screening platform for the same patient in a previous session.	
The missing criteria are filled in by the investigator	The DefinedEligibilityCriterion will encapsulate the INTEGRATE specific mechanism of how to obtain the required data by manual data entry.
The investigator is presented a page containing information about the required biological samples and molecular testing for the chosen trial. • Some of this information are molecular criteria that should be fulfilled for the selected trial	Currently, the molecular tests are treated as regular DefinedEligibilityCriterion. This might be changed once the available data w.r.t. biotracking is available. (Note that molecular test results can come from accredited and unaccredited labs. Accreditation is on a per StudyProtocolVersion basis)
…investigator is shown a CRF page… The investigator is presented a page containing…	DefinedEligibilityCriterionGroup groups DefinedEligibilityCriterion into ordered groups.

The following aspect has recently been brought up during a technical meeting. It should still be incorporated in the scenario's and/or use cases.

Description	For a clinical trial, a selected number of laboratories are accredited to perform specific tests. When assessing the eligibility of a patient, it is important to know whether relevant tests have been performed by laboratories accredited for the test.
Description	
Use case excerpt	**Meta-data model**
Accredited laboratories	The StudyProtocolVersion can have multiple AccreditedLabDefinedObservations. These are the DefinedObservations for which a PerformingLaboratory is accredited.

Meta analysis

Figure 15 shows the clinical trial meta-data required to perform a meta-analysis over the available clinical trial data.

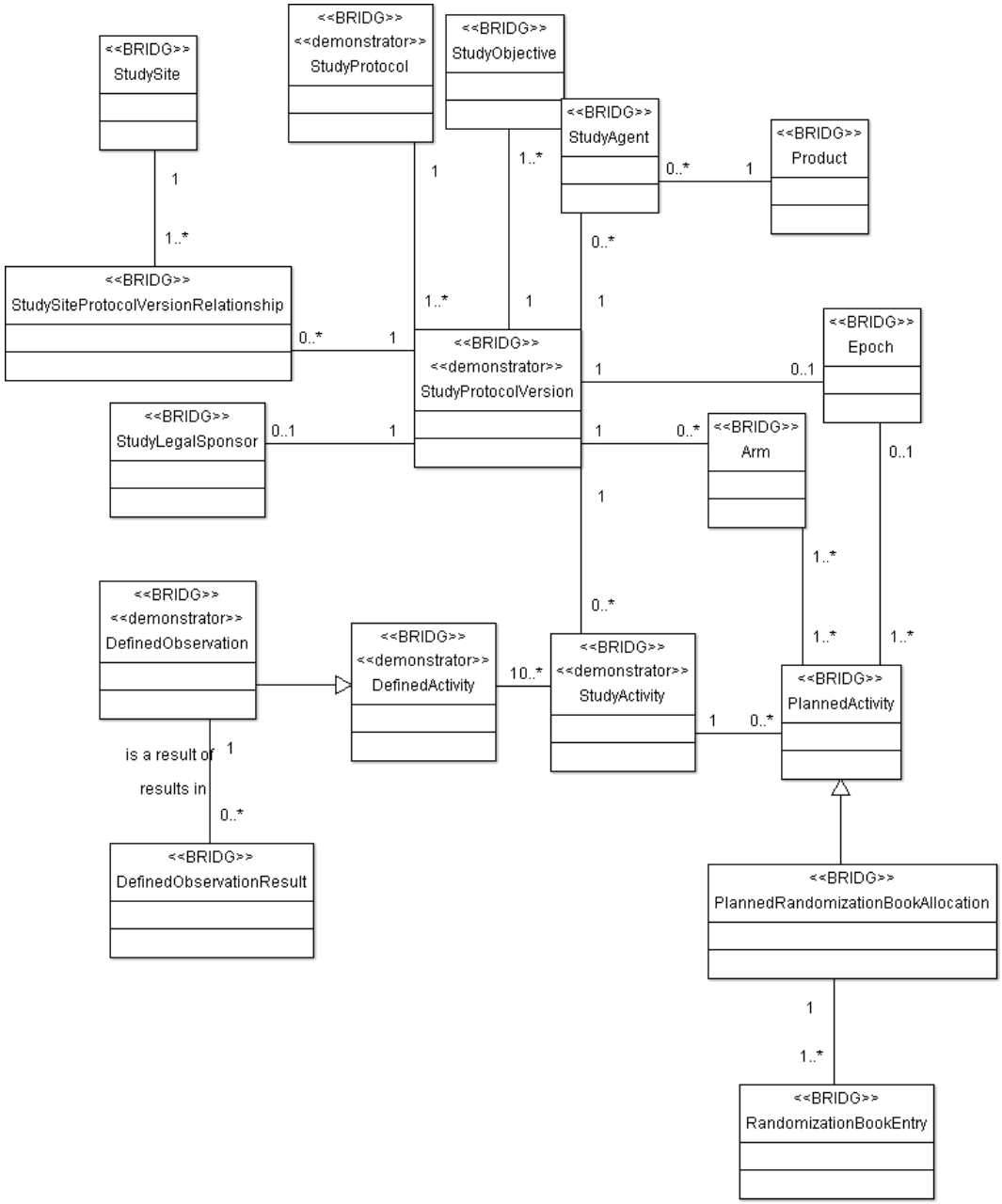


Figure 15: Meta analysis static model

Use Case ID	UC.4
Use case name	Define a query
Brief description	The researcher defines a query specifying the criteria used to create a result set.
Relevant steps	
3. The researcher specifies the selection criteria that define a subset of the patient population (i.e. the “query”).	
Description	
Use case excerpt / notes	**Meta-data model**

the selection criteria that define a subset of the patient population. Notes: querying can be performed on collected clinical trial data (of patients) and on trial meta-data. Some examples from the following rows are taken from D1.2 [10].	
• Trial meta –data ○ Specific arm of a trial	Each PlannedActivity is assigned to (at least) one arm. The RandomizationBookEntry's contain the mapping between patient number and the assigned arm.

• Trial meta –data
  ○ Particular drug
  ○ Drug that targets protein HER2
	Each StudyProtocolVersion can have StudyAgent's (related to a actual Product).
	(The StudyAgent can be used for instance in combination with an external drug – target database to find out molecular targets).

• Trial meta –data
  ○ Sponsor of a trial
  ○ Participating sites
	• StudyProtocolVersion can have a StudyLegalSponsor
	• StudySites are related to the StudyProtocolVersion by StudySiteProtocolVersionRelationship

This query is not (yet) described in the scenarios and use cases • Trial meta –data ○ Compare observations between different epochs (phases) of a trial, or the same epochs of different trials.	A PlannedActivity can be related an Epoch. Two PlannedActivities in different epochs can refer to the same DefinedActivity (e.g. tumour size assessment by MRI). The Epochs refer to different phases in a clinical trial, such as screening, or treatment.

Class descriptions

This section describes the classes[11].

Class name	Description
AccreditedLaboratoryDefinedObservation	Specifies that the lab is accredited to perform the DefinedObservation for this StudyProtocolVersion.
Arm	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 and is typically equivalent to a treatment group in a parallel design trial. 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 trial is composed of time point events (study cell) for each epoch of

[10] INTEGRATE Deliverable 1.2 – Definition of relevant user scenarios based on input from the users. A. Irrthum et al.
[11] Where applicable, the BRIDG definitions are used.

	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.
DefinedActivity	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 study subject activity group. A defined activity is a "kind of" activity rather than an "instance of" an activity.
DefinedAdministrativeActivity	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.
DefinedEligibilityCriterion	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.
DefinedEligibilityCriterionGroup	Construct used to group DefinedEligibilityCriterion and to order the groups.
DefinedExclusionCriterion	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.
DefinedInclusionCriterion	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.
DefinedInformedConsent	A reusable informed consent template.
DefinedInformedConsentActivity	The activity of obtaining a (signed) informed consent
DefinedMedicalConditionResult	A reusable template description of a sign, symptom, disease, or other medical occurrence.
DefinedObservation	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 study subject's or

	experimental unit's physiologic or psychologic state
DefinedObservationResult	A reusable, "template" description of possible findings of an observation.
Epoch	One of a set of ordered partitions of an 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 trial 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.
Laboratory	An organization with the capability and competency to perform scientific research, experiments and measurements.
Organization	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.
PlannedActivity	An activity that is intended to occur or start at some point in the context of a particular study.
PlannedRandomizationBookAllocation	An activity that is intended to occur at some point in the context of a particular study and 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) based on a randomization book.
Product	A pharmaceutical form of an active substance or placebo being tested or used as a reference in a clinical trial, including products already with a marketing authorization but used or assembled (formulated or packaged) in a way different from the authorised form, or when used for an unauthorised indication, or when used to gain further information about the authorised form
RandomizationBookEntry	An item/element of a randomization book that can be used to assign a subject to a planned arm or portion of an arm in a study.
StudyActivity	The intention to use a defined activity in the design of a study
StudyAgent	A product that is being used or tested as part of a study.

StudyExecution	An ongoing and/or past performance of a formal investigation as specified in a study protocol.
StudyLegalSponsor	A sponsor that initiates the investigation and is legally responsible for the study.
StudyObjective	The reason for performing a study in terms of the scientific questions to be answered by the analysis of data collected during the study.
StudyProtocol	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
StudyProtocolVersion	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
StudyReqruitmentStatus	Status of finding and enrolling appropriate study subjects (those selected on the basis of the protocol's inclusion/exclusion criteria) into a study, specifying the phase in the lifecycle of recruitment for the study (e.g. Not yet recruiting; recruiting; enrolling by invitation; active, not recruiting; completed; suspended; terminated; withdrawn).
StudySite	A facility in which study activities are conducted.
StudySiteInvestigator	A researcher at a study site who oversees multiple aspects of the study at a site, including protocol submission for Institutional Review Board (IRB) approval, participant recruitment, informed consent, data collection and analysis.
StudySitePersonel	A person who performs a particular role within the context of a specific study site. (this should have been named StudySitePerson)
StudySiteProtocolVersionRelationship	Specifies the link between a study site and a version of the study protocol used or available for use at that site.

7.2 Semantic Layer

7.2.1 Introduction

The semantic interoperability layer covers the common information model (CIM) and how it interacts with new data sources and the required information. From the information viewpoint, it is necessary to define the different data structures of the semantic layer and also the flow model of the information entities.

In next sections is presented the general diagram of the semantic layer, the different data structures and the flow model of the information.

7.2.2 Diagram

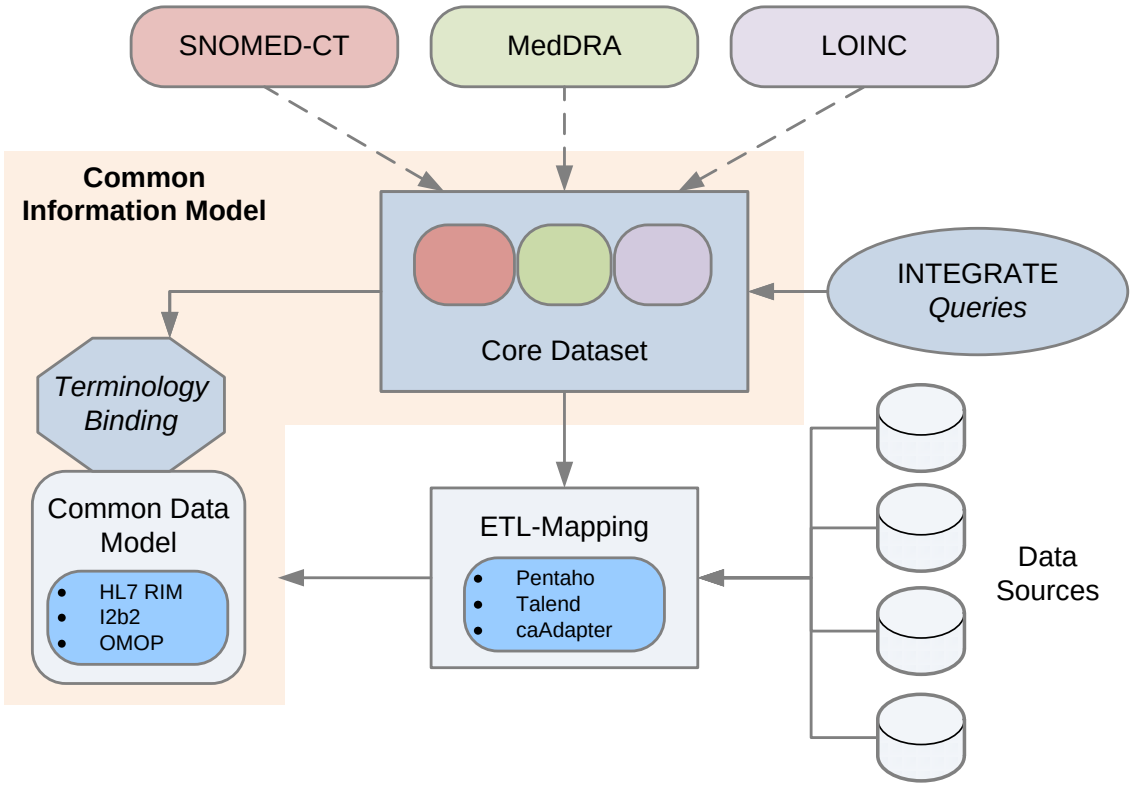


Figure 16: Semantic Layer

7.2.2.1 Static Data Structure Model

In this section, the different data structures used in the model are described, together with their relation to the use cases. It is focused on the common information model, specifically on the core dataset and the common data model.

Core Dataset

The core dataset component will act as the medical vocabulary of the INTEGRATE platform. For that purpose, it is necessary to use a subset of ontologies that represent

the knowledge with a set of relevant concepts and relationships between those concepts.

The Figure 17 shows an example display of a medical concept (in this case *Anthracycline*) in a medical ontology (SNOMED-CT).

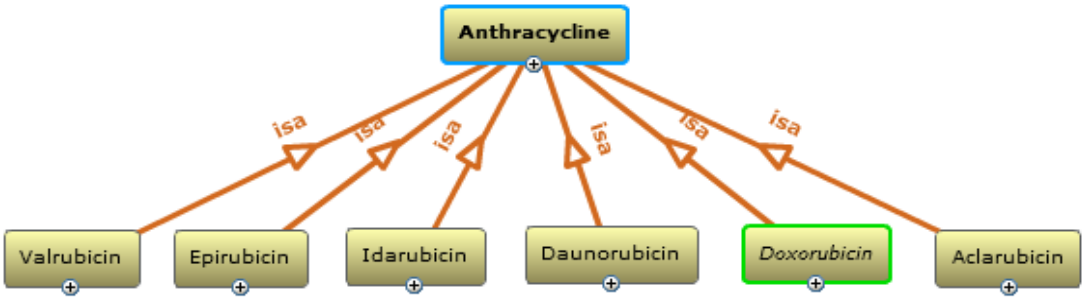


Figure 17: Example of SNOMED-CT visualization

The core dataset will include concepts of a medical vocabulary, such as SNOMED, used within the CDM. Additional concepts related to those used within the CDM will be included as well.

Common Data Model

The common data model is the schema of the data warehouse for the semantic interoperability layer. It is the homogeneous storage of patient-based information from the different sources. The CDM will receive a query including with core dataset concepts and temporal restrictions. The query is then executed to retrieve data from the common data model. Core dataset concepts within the query will be modified by reasoning over the core dataset. The modifications will be performed over relations presented in the ontology such as is-a.

Finally semantically-aware information will be returned to the user by the semantic layer.

7.2.2.2 Information Flow Model

This section requires to detail two different *"information entities"*. These entities are related to new data sources and the INTEGRATE query engine (**data source entity** and **query entity**). The flow models of the two entities are described below.

Data Source Entity

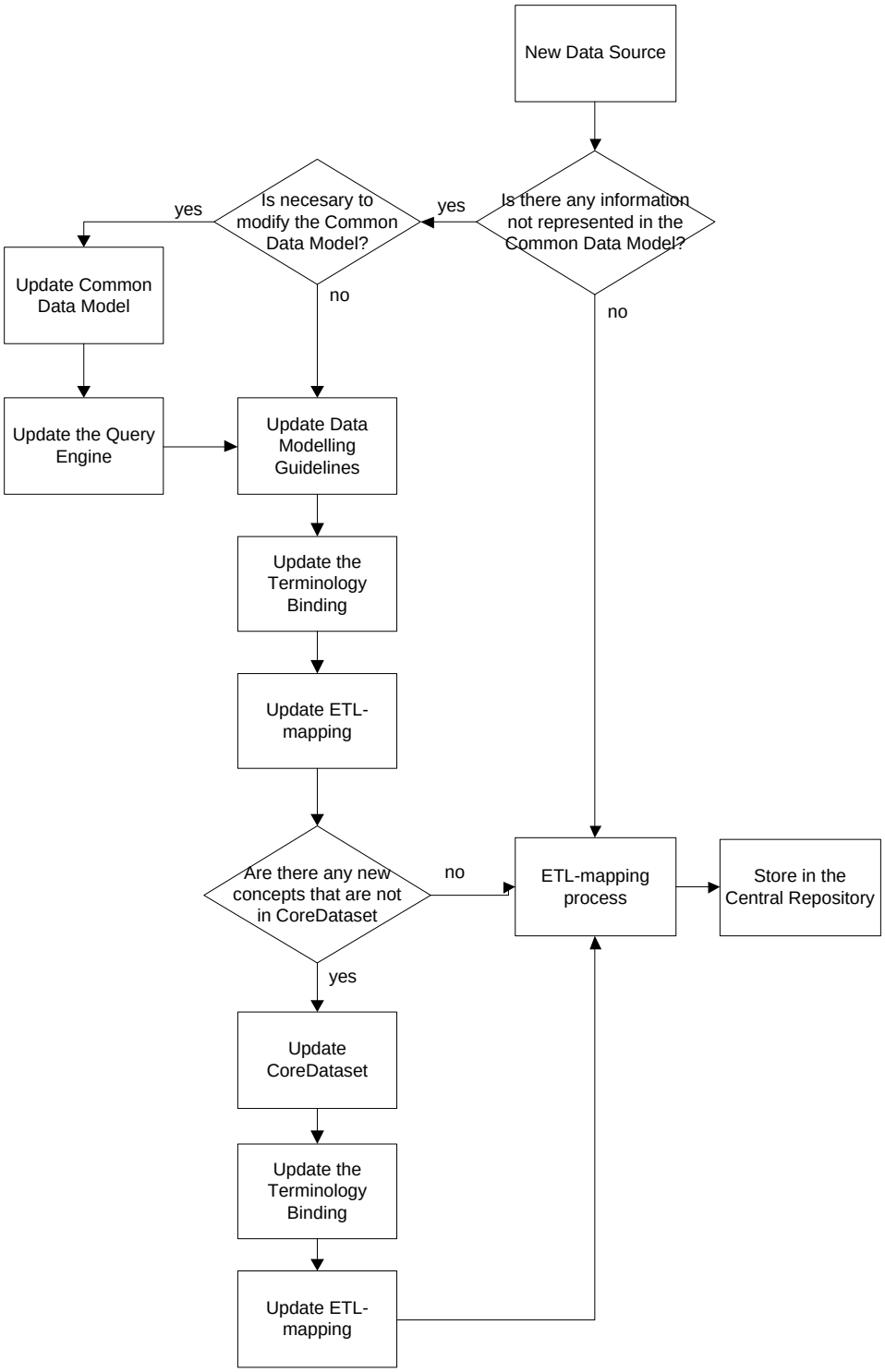


Figure 18: Flow Model of the Data Source Entity

"New Data Source" entity is defined by relevant information for the INTEGRATE platform. This information should be stored in the central repository.

As shown in Figure 18, the entity starts with the data source information that should be added to the central repository, this information is given to the semantic layer. In next steps, if the new information is not represented in the model (this happens rarely) then it is necessary to change the schema of the data warehouse (CDM) and the other components. If the new information is represented in the model, then ETL mapping process extracts information from the data source, transforms and loads it into the data warehouse.

INTEGRATE Query Entity

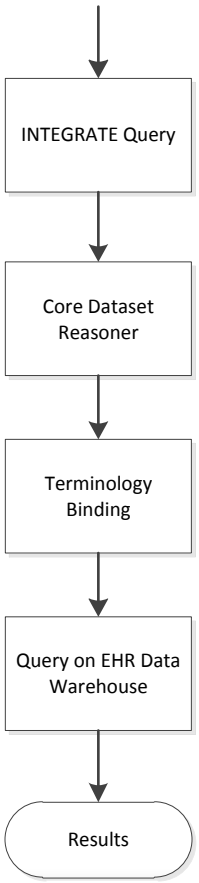


Figure 19: Flow Model of the INTEGRATE Query Entity

The INTEGRATE query entity is defined by the query sent to the semantic interoperability layer of the platform. This query will be executed against the data warehouse.

As shown in Figure 19, the entity starts with the query launched to the platform using Core Dataset concepts.

In second step the Core Dataset concepts are expanded with the knowledge inferred in the reasoner. In next step, the terminology binding indicate in what place of the schema of the Data Warehouse are placed these core dataset concepts. Finally, the translated query is performed with the list of the Core Dataset concepts and temporal restrictions defined by the original query.

8 Deployment View

8.1 Deployment View

8.1.1 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 (D1.2). In the first iteration of this document, only a high level view is given of the component deployment in INTEGRATE. This means that some components/nodes of the current presented deployment diagram will be refined/moved/split up in a later iteration. For example some components will be moved from server side to a client, if it is decided to use a thick client to access the INTEGRATE services (not decided at this point). A more detailed deployment view (diagram) will be given as soon as we get a better view on the components.

It was decided that most of the components of the INTEGRATE framework will be developed in Java. All the main server side components are java-based and will run on servers with a Java virtual machine installed. The client side however will be mainly C# based because this language provides better support for designing advanced graphical user interfaces. The communication between the different modular components is SOAP based (uniform for both Java and C#). Choosing for SOAP has some advantages compared to alterative protocols like REST. First SOAP is still the most commonly used protocol for exchanging information between two services. Because it is used so frequently, the partners (developer stakeholders) of INTEGRATE have experience with it, which is not the case with protocols like REST (requires time and effort to learn). Another reason for choosing SOAP is the presence of the security extensions defined in the WS-* stack. More information about these security extensions can be found in the state-of-the-art document[12].

Concerns addressed (see paragraph 3.2)

CUS-001 *(1)*, **CUS-004, CUS-005, CUS-006, CIN-001, CIN-002, CIN-003**

(1) Requirements concerning the reviewers

[12] See deliverable D2.1 State-of-the-Art Report on Standards

8.1.2 Diagram

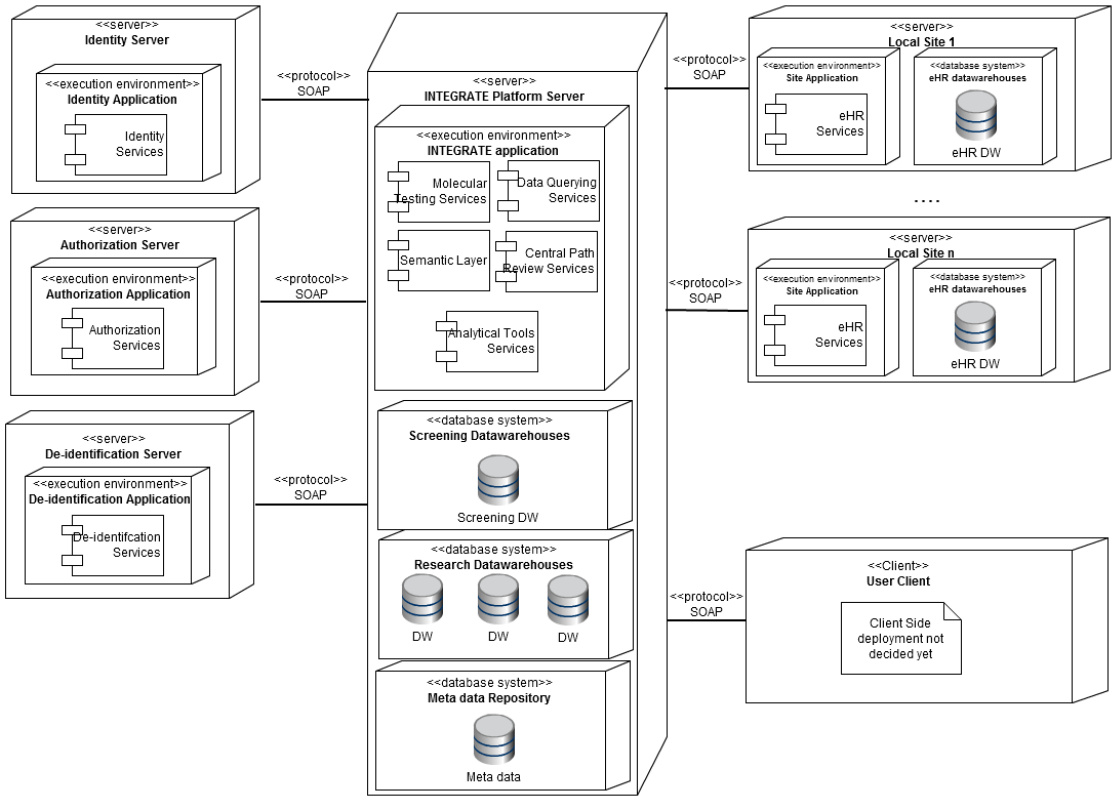


Figure 20: INTEGRATE deployment view

8.1.3 Nodes

Node	Type	Sub-node	Sub-node Type	Components	Process Speed	Memory
Identity Server	*Server*	Identity application	*execution environment*	Authentication Services	To be determined	To be determined
Authorization Server	*Server*	Authorization application	*execution environment*	Authorization Services	To be determined	To be determined
De-Identification Server	*Server*	De-identification application	*execution environment*	De-identification Services	To be determined	To be determined
INTEGRATE Server	*Server*	INTEGRATE application	*execution environment*	Molecular testing services Data querying services Semantic layer Central pathology review services Analytical tools services	To be determined	To be determined
		Screening datawarehouses	*execution environment*	Semantic Layer Trial Data Querying	To be determined	To be determined

		Research Datawarehouses	*database system*	Screening Datawarehouse	To be determined	To be determined
		Meta-data repository	*database system*	Datawarehouses	To be determined	To be determined
Local Site 1,...	*Server*	Site application	*execution environment*	EHR services	To be determined	To be determined
		EHR Datawarehouses	*database system*	EHR Datawarehouse	To be determined	To be determined
User Client	Client	*Unknown*	*Unknown*	*Unknown*	To be determined	To be determined

9 Data Protection View

9.1 Authentication

9.1.1 Introduction

Authentication is an important security concept in the INTEGRATE legal framework. Users interacting with INTEGRATE services and resources need to be authenticated (and authorised) before they can access these services and resources. These concerns are addressed in the authentication view. A set of the main general architectural building blocks is listed in this section, meeting the authentication requirements.

Concerns addressed (see paragraph 3.2)

CAC-001 *(1)*, **CDE-001** *(2)*

(1) The link between the user databases and the INTEGRATE platform must comply with the INTEGRATE legal framework

(2) The functionality of the identity management is important for the stakeholders responsible for developing the security system on the INTEGRATE platform.

9.1.2 Diagram

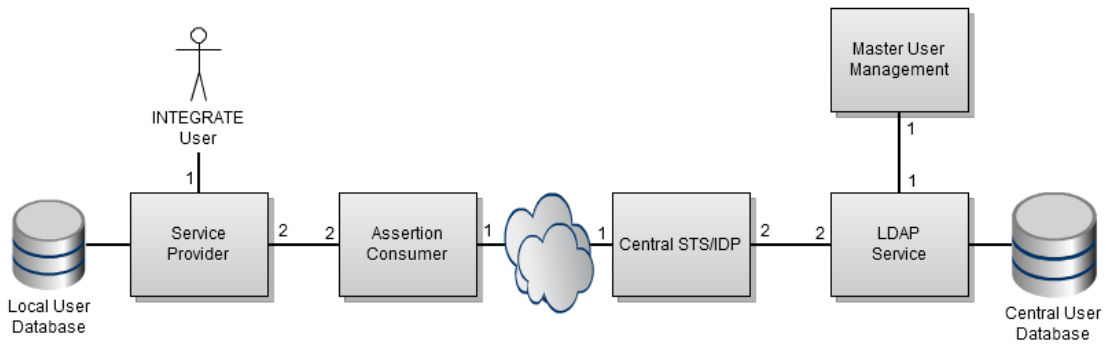


Figure 21: Authentication security view

9.1.3 Interfaces and Components

9.1.3.1 Service Provider

The service provider represents the service that can be accessed once the INTEGRATE user is authenticated. A service provider can have a local storage for storing service specific user data..

Related to uses cases: -

Interface	Description
1	An INTEGRATE user interacts with the front-end part of an INTEGRATE service provider. This front-end requires that the user is authenticated.
2	*No interface exposed to the assertion consumer*

9.1.3.2 Assertion Consumer

The assertion consumer will check whether a user is authenticated to the service provider. If not, the user is redirected to the IdP where he must enter his credentials. The returned authentication token of the IdP is validated by the assertion consumer.

Related to uses cases: **UC.SEC.1, US.SEC.2**

Interface	Description
1	*No interface exposed to the central STS/IdP*
2	The assertion consumer exposes following functionality to the service provider: • Return user identity information of an authenticated user • Enforce authentication of a user

9.1.3.3 Central STS/IdP

The security token service (STS) or identity provider (IdP) is responsible for the actual authentication. Together with the security framework, it provides the necessary functionality to successfully authenticate users. It requests credentials, provides the identity assertion, etc.

Related to uses cases: **UC.SEC.1, UC.SEC.2**

Interface	Description
1	The central STS/IdP provides following functionality to the assertion consumer: • Provide identity assertion based on given credentials
2	*No interface exposed to LDAP service*

9.1.3.4 Master User Management

The master user management component handles user management. The component is a front-end to the central user database, allowing for easier management of the different users within the INTEGRATE platform.

Related to uses cases: **UC.SEC.3**

Interface	Description
1	*No interface exposed to LDAP*

9.1.3.5 LDAP Service

The LDAP service provides enhanced functionality for user management. It serves as an additional layer on top of the central user database. LDAP allows for the definition of fine-grained password policies, better management of users, etc.

Related to uses cases: -

Interface	Description
1	The LDAP service exposes following functionality to the master user management: • Modification of user data stored in the central user database
2	The LDAP service exposes following functionality to the central STS/IDP: • Retrieval of user information from the central user database

9.2 Authorisation

9.2.1 Introduction

One of the major requirements of the INTEGRATE legal framework (see D1.3) is that access, to a particular service or resource, should only be provided to users that are allowed to see them. The authorisation view addresses these specific security concerns by providing an overview of the main components that are needed for authorisation and the connections between these components. In the current iteration only the most important components are defined, these will be refined in the next iteration of the document.

Concerns addressed (see paragraph 3.2)

CAC-001 *(1)*, **CDE-001** *(2)*

(1) The link between the attribute storage, the policy storage and the INTEGRATE platform must conform to the INTEGRATE legal framework

(2) The functionality of the identity management is important for the stakeholders responsible for developing the security system on the INTEGRATE platform.

9.2.2 Diagram

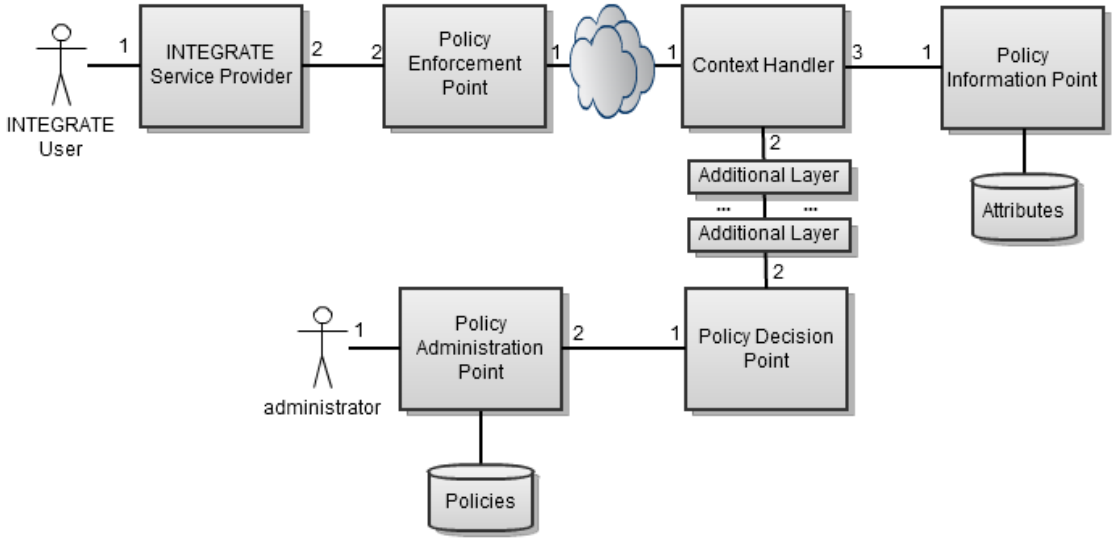


Figure 22: Authorisation security view

9.2.3 Interfaces and Components

9.2.3.1 INTEGRATE Service Provider

This represents each service (offering some specific functionality) of the INTEGRATE platform that is protected by access control.

Interface	Description
1	An INTEGRATE user interacts with the front-end part of an INTEGRATE service. This front-end offers functionality that is protected by access control
2	*No interface exposed to the policy enforcement point*

9.2.3.2 Policy Enforcement Point

When a subject performs an action on a resource in a INTEGRATE service protected by access control, the PEP will intercept this access request. It will trigger the context handler and provides access decision information. After the PDP has made a decision, the PEP will allow/deny access to the resource depending on the response content.

Interface	Description
1	*No interface exposed to the context handler*
2	The policy enforcement point exposes following functionality to the INTEGRATE service provider: • Enforces policy decision point decisions

9.2.3.3 Context Handler

The context handler is triggered by the PEP and will generate a decision request containing access information. This request is sent to the PDP where a decision is made and the response(s) is send back to the context handler. The context handler finally pushes this response(s) back to the PEP.

Interface	Description
1	The context handler exposes following functionality to the policy enforcement point: • Generating requests using the access information coming from the PEP
2	*No interface exposed to the policy information point*
3	*No interface exposed to the policy decision point*

9.2.3.4 Policy Decision Point

The PDP interprets the requests coming (and generated) from the context handler and evaluates them using the policies that were registered in the PAP of the INTEGRATE platform. When a decision(s) is made the PDP generates a decision response and sends it back to the context handler.

Interface	Description
1	No Interface exposed to the policy administration point
2	The policy decision point exposes following functionality to the context handler: • Make an access decision for a given authorisation request using the defined policies.

9.2.3.5 Policy Information Point

If a request to the PDP contains insufficient information, the PDP can request extra external information for making an access control decision to the PIP (over the context handler). The PIP is connected to attribute stores (of INTEGRATE) where the missing attributes can be found.

Interface	Description
1	The policy information point exposes following functionality to the context handler: • Provides missing attributes to the context handler

9.2.3.6 Policy Administration Point

The PAP is responsible for managing the policies of the INTEGRATE authentication framework. For this it offers functionality to generate, maintain, remove and secure these policies that will be used for access control decisions. It is important that this component will provide these functionalities in a user-friendly and intuitive way.

Interface	Description
1	An administrator interacts with the front-end part of the policy administration point. This front-end offers functionality to the administrator to generate new policies.
2	The policy administration point exposes following functionality to the policy decision point: • Provides all policies that are stored in the policy database

9.2.3.7 Additional Layers

Additional layers can be placed between the context handler and the PDP in order to extend the basic implementation of XACML without changing the decision engine of the PDP (see part II Security Framework of this document). The added components manipulate the requests and pass them to the next layer, this process repeats until the request is sent to the PDP. These layers need to have strictly defined interfaces in order to provide a generic solution where new additional layers can be easily fit in.

9.3 De-identification

9.3.1 Introduction

In the introductory chapters (see 5.1.3) of this document, it became clear that there is a separation between the scenarios according to the legal domain they belong (the "trial execution" and "research" domain). All data and patient information entering the research domain must be de-identified in order to meet the legal requirements, defined in D1.3 (a researcher is legally not allowed to see identifying information). The concerns regarding de-identification are addressed in this view, offering a security solution for de-identification requirements of the INTEGRATE platform. The focus is not to give a detailed view yet, but identify the general process that is needed in INTEGRATE to separate the two domains using de-identification.

Concerns addressed (see paragraph 3.2)

CAC-001 *(1)*, **CDE-001** *(2)*

(1) The link between the datawarehouses and the INTEGRATE platform must comply with the INTEGRATE legal framework
(2) The functionality of the identity management is important for the stakeholders responsible for developing the security system on the INTEGRATE platform.

9.3.2 Diagram

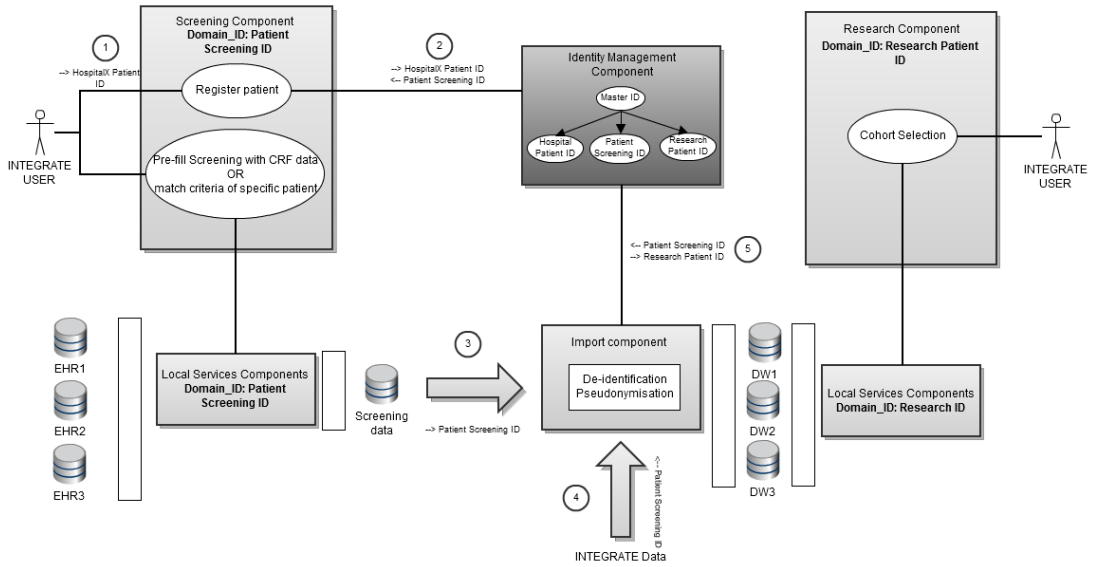


Figure 23: De-identification process

Figure 23 describes the process of de-identifying patient information and data in INTEGRATE. In the "trial conduct" domain, an INTEGRATE user can register a new patient (1) for screening purposes (assuming that the patient is not registered yet). When registering this patient the user includes the local hospital ID of this patient. This hospital ID can be used to refer to the patient in the hospitals. During the registration, the hospital ID of the patient is sent to the central identity management service of INTEGRATE (2). This service will create a screening ID for the given patient and link it with hospital ID by using a parent master ID. This master ID is only visible within the identity management component itself. The screening ID of the patient is presented to the INTEGRATE user and will be used as reference for the remainder of the screening process.

During the screening process patient information and data is generated/extracted and stored in a central screening datawarehouse (see 6.1). When this process ends, the data of this screening datawarehouse will be made available for research purposes. Because this data contains identifying information, it should be de-identified before it can be used in the "research" domain (3). This also applies for data that enters the "research" domain from outside the platform (4). To attain anonymity, the data (screening data and data coming from outside) is processed by an import component before it is stored in the research datawarehouses. This import component will remove all identifying information present in the data. To retain traceability of the patient, a new research ID is generated (5) for the patient in the identity management service and is connected with the master ID. By using this research ID in the research domain, a patient can be referred to without giving away his identity (as the identity management service will never give away the screening/hospital/master ID that is linked with the research ID).

10 Relationships between Views

10.1 Functional - Deployment View Consistency

Here the components discussed in the functional view are linked with the nodes where these components are deployed. The following table provides an overview of this relationship. This table will be extended in later iterations of this document as the deployment and functional view will be extended with new components and nodes.

Deployment Viewpoint	Functional Viewpoint
Identity services	Central STS/IDP Master user management LDAP services Assertion consumer
Authorization services	Policy enforcement point Policy decision point Policy administration point Policy information point Context handler
De-identification services	Identity management component Import component
Molecular testing services	Informed consent service Trial management service Screening service Criteria matching service Patient identity management service Biotracking service
Data Querying services	Cohort selection service
Semantic layer	CIM data access service Core dataset reasoner service Terminology binding service Query engine service ETL service
Central pathology review services	Publishing service Imaging service Management service Messaging service Viewer service Report service Resolution service
Analytical tools services	Analytical tools Sharing of predictive models Publishing service Tools and model service
EHR services	EHR connectivity service

10.2 Functional - Information View Consistency

The following table describes the relation between the functional and the information view. It provides more detail on how the functional components maintain the data stores of the application.

Data Stores	Functional Components with access
User database	LDAP services
Policy storage	Policy administration point
Attribute storage	Policy information point Identity provider Identity management service
Screening datawarehouse	Semantic layer
Common data model	Query engine service Semantic ETL service
Image repository	Imaging service Publishing service
Image metadata	Imaging service Publishing service
Research datawarehouse	Semantic layer
Tools and model repository	Tools and model service
EHR datawarehouse	Semantic layer
Trial meta-data repository	Trial management service

10.3 Deployment - Information View Consistency

Each of the major components requires access to specific data. The following table describes the data sources used by the different major components.

Deployment Viewpoint	Data Stores
Identity services	User database
Authorization services	Policy storage
De-identification services	Screening datawarehouse Research datawarehouse
Molecular testing services	Trial meta-data repository Patient database
Data Querying services	Research datawarehouse
Semantic layer	Screening datawarehouse Research datawarehouse EHR datawarehouses Common data model
Central pathology review services	Image repository Image metadata repository
Analytical tools services	Tools & model repository

11 (PART II) Security Framework

11.1 Introduction

The goal of the 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 D1.1, D1.2 and D1.3).

The INTEGRATE security framework 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 as much as possible independent. This means that 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.

One example of the referred coupling of components forming a security framework relates to the relation between user management and access right management. From an administrator's point of view these go hand in hand and should thus be managed together. However, in a distributed system, identity and access rights are typically spread over different components. This particular problem can for example be dealt with by introducing a single management component that oversees configuration in the different security components (see also Figure 24)[13].

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). Note that such issues often only occur in very specific cases which make them harder to detect. In that respect, automatic policy testing can be of help[14].

11.2 Overview

The figure below (Figure 24), shows a generic high level component view of the INTEGRATE security framework. A short description of the major components in the framework is given in the following paragraphs.

[13] In an Attribute Based Access Control (ABAC) model based implementation (see further), this means that access management changes need attribute and policy changes to be reflected in different attribute repositories.

[14] G. De Angelis, T. Kirkham and S. Winfield, “Access Policy Compliance Testing in a User Centric Trust Service Infrastructure”, in Proc. of the 1th International Workshop on Quality Assurance for Service-based applications (QASBA 2011), Lugano, Switzerland, Sep. 2011

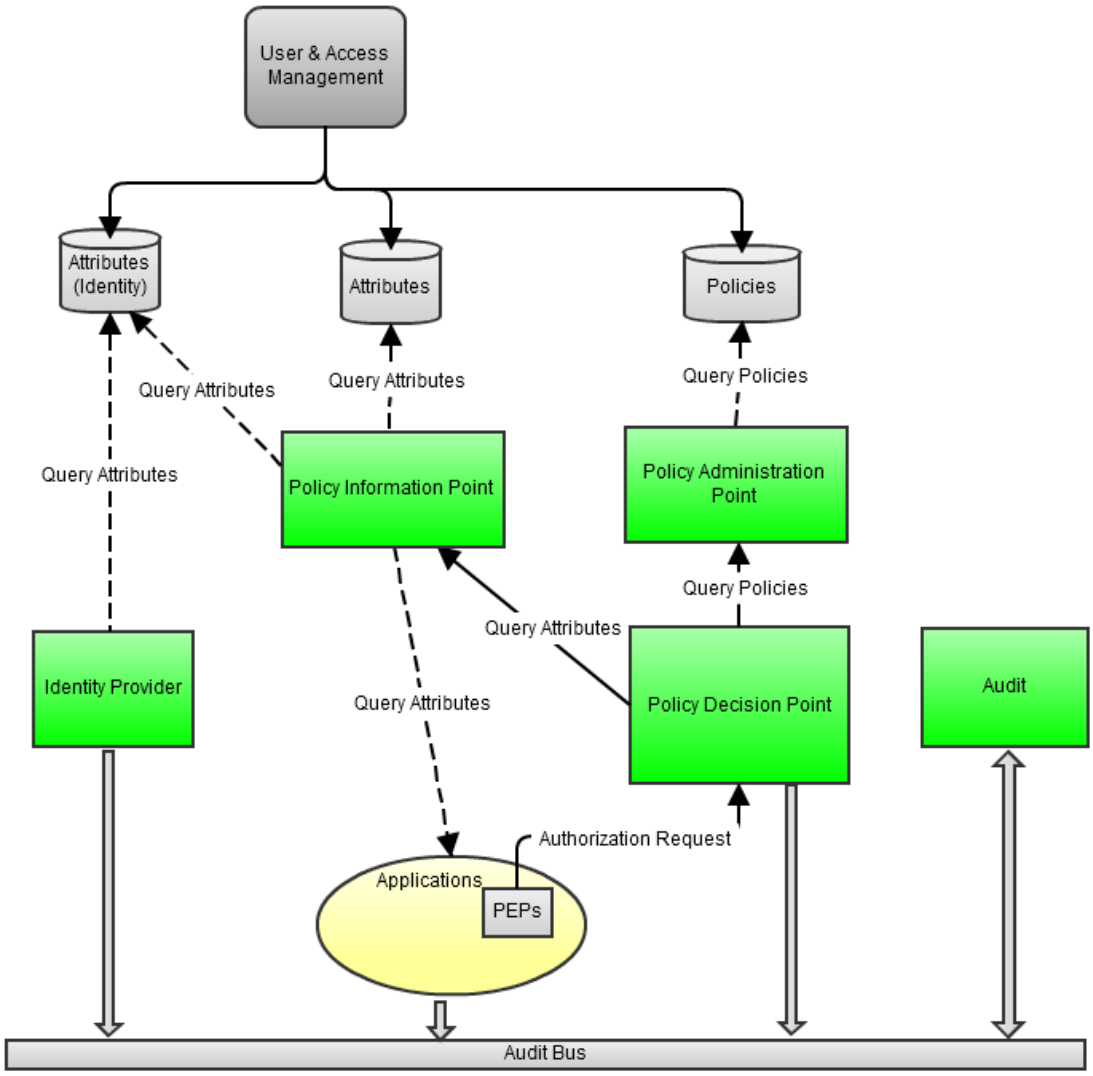


Figure 24: High level component of a generic security framework

As mentioned in the project proposal, INTEGRATE aims to technically enforce and govern access control throughout the collaborative environment by relying on policy-based authorization services. INTEGRATE builds upon the extensible access control modeling language (XACML[15]) policy language which is an OASIS XML-based standard for authorization and access control. It is the most prominent policy language standard and (to our knowledge) the only one for which multiple, open source and commercial, implementations exist.

XACML implements the attribute-based access control (ABAC) model. In ABAC, attributes that are associated with a user, action or resource serve as inputs to the decision of whether a given user may access a given resource in a particular way. ABAC presents an access control model inherently capable of meeting many of the

[15] More information on XACML can be found in deliverable D2.1 (State of the art report on standards)

"modern" access control demands (e.g. data dependent access policies, environment dependent policies, …).

11.2.1 Access Control Decisions (and PDP)

In this document, the XACML access control model is used as reference for descriptions about the authorization components. This model is equivalent to the ISO 10181-3 model[16] (but uses different terms for the different components) and can be found on Figure 25.

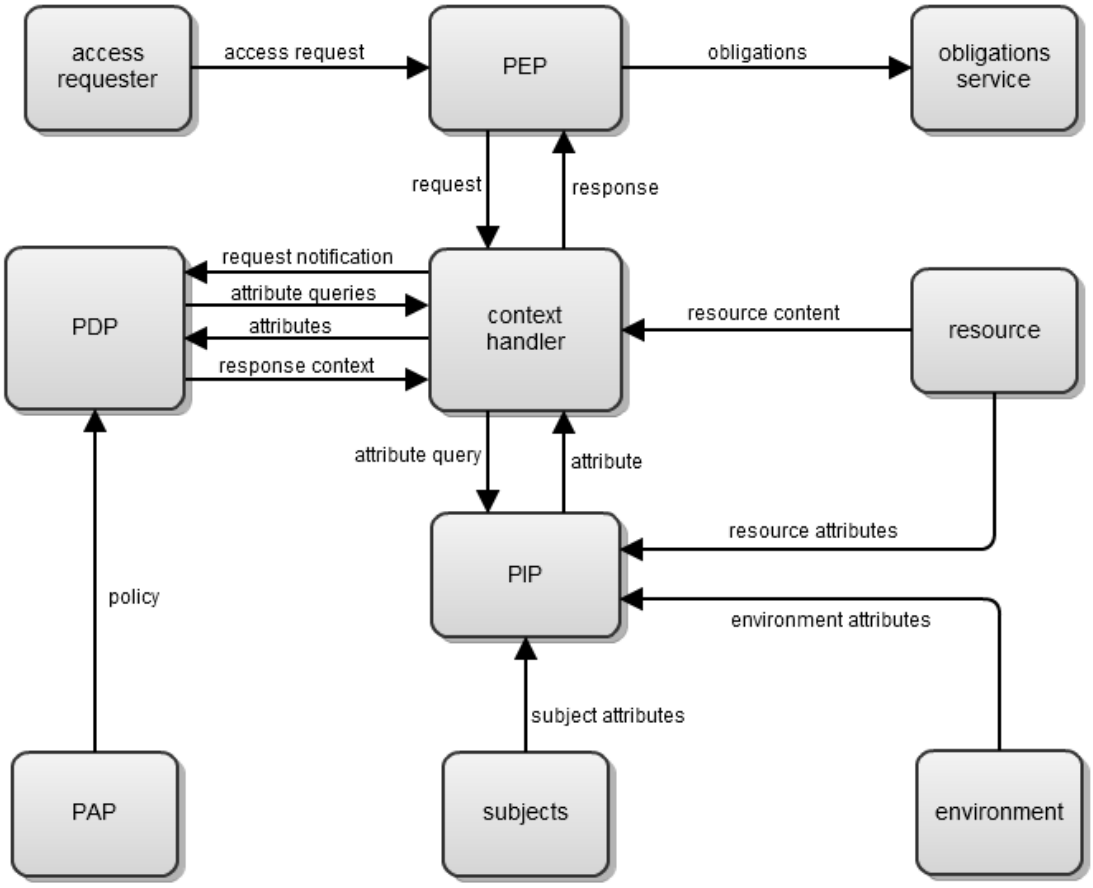


Figure 25: XACML Access Control Model

In XACML, an access request is modelled as a "subject" who wants to perform "action" on a "resource" (subject/action/resource triplet). A policy enforcement point (PEP) intercepts this request and queries the policy decision point (PDP) with the question whether the user is allowed to perform this action on the resource. The PDP makes a decision based on the request parameters and the available policies[17]. These policies (in the form of XML files) are created and managed by the policy administration point

[16] ISO/IEC 10181-3 Information technology – Open Systems Interconnection – Security frameworks for open systems: Access control framework

[17] When the PDP encounters undefined attributes within the policies, the PDP queries the policy information point (PIP), which is responsible for gathering additional required information.

(PAP). The PDP decision goes back to the PEP which is responsible for enforcing it (allowing or denying access).

Although XACML is the most prominent policy language standard, it has its share of limitations. For example there are no mechanisms for expressing links (e.g. hierarchy) between different attributes in a convenient way in a XACML policy. Approaches for solving such problems include e.g. use of a strict structural profile in authored policies (e.g. RBAC profile for XACML)[18] or the use of semantic reasoners as policy decision engines with linked to that the use of a security ontology as policy language[19].

The INTEGRATE approach to solve the limitations of ABAC and in particular the XACML implementation is not proposing language alternatives. Instead, INTEGRATE aims to introduce supporting components complementary to a standard XACML access control decision engine. Using this strategy has several advantages. First off all, the final solution remains for a large part standards based and allows drop-in replacement of core components. This offers the obvious advantages towards further exploitation. Secondly, this separates the concern of maintaining the policy decision logic (and other support components) from developing more advanced features. The former is not to be underestimated for a mission critical component (cf. effort required for validation). This approach has been previously[20] successfully tested in the context of attribute translation between XACML policy decision points (PDP's) in different security domains (with a different attribute vocabulary).

11.2.2 Identity Provider (IdP)

The IdP component is a service provider responsible for the identity management within INTEGRATE. Among other functionality, the IdP provides Single Sign On (SSO). As already described in the state-of-the-art deliverable[21], SSO works generally spoken as follows: An INTEGRATE user tries to access an INTEGRATE service provide (SP) for which he/she does not have a local active authenticated session. The client of the user will be requested to pass an INTEGRATE identity assertion. The client will request this assertion from the identity provider (IdP). If the user is already authenticated, the IdP will provide the identity assertion (Single Sign-on), if not, the user will first have to authenticate him/herself. The client will then pass this assertion to the SP that the user originally wanted to access. The SP will verify the assertion and give the client access if the identity assertion is evaluated as valid.

A commonly used implementation method for identity provision including SSO is the SAML protocol (SAML - Security Assertion Markup Language[22]). It is an XML-based protocol, making it possible to exchange authentication and authorisation data between one or more security domains. SAML will be used for identity providing in INTEGRATE.

[18] http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-rbac-profile1-spec-os.pdf
[19] Rodolfo Ferrini and Elisa Bertino. 2009. Supporting RBAC with XACML+OWL. In Proceedings of the 14th ACM symposium on Access control models and technologies (SACMAT '09). ACM, New York, NY, USA, 145-154.
[20] I. Ciuci, B. Claerhout, L. Schilders, R. Meersman. 2011. Ontology-Based Matching of Security Attributes for Personal Data Access in e-Health
[21] See Deliverable D2.1
[22] More information about SAML and other identity provision methods can be found in deliverable D2.1 (state-of-the-art)

11.2.3 Audit

A (centralised) audit service is an integral part of any security framework; especially on platforms like INTEGRATE containing sensitive data that can only can be accessed by users who have sufficient access rights. Each authentication attempt (both successful and failed), resource access/change, available issue (e.g. server exceptions), etc. needs to be logged on 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.

11.3 (Specific) Requirements from the Scenarios

Requirements for the security framework have been identified based on the user scenarios and the legal analysis, available in deliverable D1.2 and deliverable D1.3 respectively.
This section lists those requirements which have a considerable or specific impact on the scientific and technical work in INTEGRATE relating to security.

11.3.1 Centralised Governance Framework for Managing Security (OVERALL)

INTEGRATE aims to provide a platform in which different loosely coupled applications can interoperate in a transparent way. With respect to security this implies that one can guarantee that all data and applications/services are governed by a (uniform) well defined legal framework. This is a difficult task in a distributed environment. Often in such environments, each individual administrator (managing one or more of the different applications) implements (or configures) the security policies for his applications in a proprietary way. Changes to the overall security policy (even as simple as adding a user) can thus require some time to take effect, i.e. until each application (configuration) has been updated. Furthermore, this approach is error-prone (with a lot of human action) and in practice requires additional auditing steps.

Clearly, within INTEGRATE the objective is to introduce central management which allows security policy changes to be configured for the whole platform at once (cf. managing users [**UC.SEC.3**] and access policies [**UC.SEC.4**]). Also on a lower level, measures are taken to ensure uniform enforcement of security policies. One example is the introduction of a centralised PDP. This PDP makes most of the access control decisions for the entire framework. Next to simplifying administration, this approach also offloads some of the burden of security implementation from the independent applications. Note that “centralised PDP” should be interpreted in a broad way, meaning that it includes the setup in which a number of different instances of the same PDP implementation get fed by the same policies from a central point.

11.3.2 Authentication & Identity Provision Related Requirements

A user must be able to authenticate himself on the INTEGRATE platform [**UC.SEC.1**]. This authentication is achieved with a Single Sign-On (SSO) session. Once a user is authenticated within a SSO session, he can use every component within the INTEGRATE platform. SSO is implemented using the Security Assertion Markup

Language(SAML[23]). Single Logout (SLO) [**UC.SEC.2**] will not be implemented (see also for a discussion on SLO[24]). More information on SAML can be found in the state-of-the-art report on standards[25]. As an addition to the usual authentication, LDAP will be used to enable fine-grained password policies and to facilitate the user management.

Based on the analysis of the use cases and scenarios, specific authentication & identity provision requirements like federation and delegation will probably not be necessary.

11.3.3 Consent

Consent is an important concept in the medical domain. Roughly speaking, consent is required for each action that can damage the integrity of a person, be it a medical act, usage of an experimental drug or the processing of personal data (for a less simplified definition see e.g. CONTRACT[26], an European project dedicated to the subject of consent in care and clinical trials).

The absence or presence of consent for a specific matter translates into an authorization decision. For example, if a patient has consented to engaging in the screening process, the physician can be given access to screening specific data of this patient. Consent can be considered to be a personalised security policy (defined by the data subject, governing only his data). The integration of consent into the authorisation framework can lead to improved assurance of compliance to consent directives (and thus to law) because of the automated enforcing.

Requirements relating to consent for INTEGRATE are mentioned in [**UC.IC.1**]. The link between consent and authorization will be researched in more detail later in the project.

11.3.4 De-identification & Pseudonymisation

Within INTEGRATE a legal framework has been set up for enabling re-use of trial related patient data by the BIG researchers. This framework requires data to be de-identified. Because new incoming data on a person must still be linkable to the same de-identified person, the patient ID's are pseudonymized.

Different components within the framework use different pseudo-ID's for the same real ID. To enable this, all of the pseudo-IDs are linked to a master ID. This way data belonging to one patient that are used by different components can still be linked back to that one patient. Even if it is not allowed to link the data back to a specific person, it should still be possible to find out which local ID's match the same person. The links between the local pseudo-ID's and the master ID's are protected by a trusted third party (TTP). A more detailed description of the INTEGRATE approach can be found in 9.3 .

[23] http://wiki.oasis-open.org/security
[24] https://wiki.shibboleth.net/confluence/display/SHIB2/SLOIssues
[25] D2.1 State-of-the-art report on standards
[26] http://www.contract-fp7.eu/site/

11.4 S&T Challenges

In this part, the primary scientific and technological (S&T) challenges with respect to the security framework will be described.

11.4.1 Contextual attributes

From the scenarios (D1.2) it was clear that an INTEGRATE user can have different roles within different trials, i.e. an investigator in one trial and lab member in another trial. These role differences need to be taken into account when making an access control decision in INTEGRATE (the role, and therefore the access rights, depends on the context – trial in this case - in which the person resides). The standard XACML implementation does not provide such functionality; the standard RBAC profile of XACML can handle only one context a time. In INTEGRATE one of the challenges is to make the PDP aware of these role differences in order to make the correct access control decision.

Before a solution was worked out, a literature study was conducted on the subject of context awareness. In many papers the limitations of the XACML RBAC profile are acknowledged. The problems these papers encountered vary greatly; the main problems that were encountered deal with separation of duty[27], fine-grained consent[28], user-user delegation[29], role hierarchy[30] and access negotiation[31]. We have found no discussions on the subject of dealing with multiple contexts within one XACML request.

Within INTEGRATE the following solution is proposed for implementing context awareness in XACML. An additional layer is added in front of the PDP (see Figure 26), this eliminates the need to extend the PDP itself (meeting the INTEGRATE approach regarding access control decisions). It is the responsibility of this layer to handle the context-based roles, the PDP is not aware of the context in which the request is made.

[27] Rodolfo Ferrini and Elisa Bertino. 2009. Supporting RBAC with XACML+OWL. In Proceedings of the 14th ACM symposium on Access control models and technologies (SACMAT '09). ACM, New York, NY, USA, 145-154.
[28] G. Kounga, M. Casassa Mont, and P. Bramhall, “Extending xacml access control architecture for allowing preference-based authorisation,” in TrustBus, 2010, pp. 153--164
[29] Diala Abi Haidar, Nora Cuppens-Boulahia, Frederic Cuppens, and Herve Debar. 2006. An extended RBAC profile of XACML. In Proceedings of the 3rd ACM workshop on Secure web services (SWS '06). ACM, New York, NY, USA, 13-22.
[30] David Power, Mark Slaymaker, Eugenia Politou, Andrew Simpson. On XACML, role-based access control and health grids
[31] D. A. Haidar, N. Cuppens, F. Cuppens, and H. Debar. Access Negotiation within XACML Architecture. Second Joint Conference on Security in Networks Architectures and Security of Information Systems (SARSSI), June 2007.

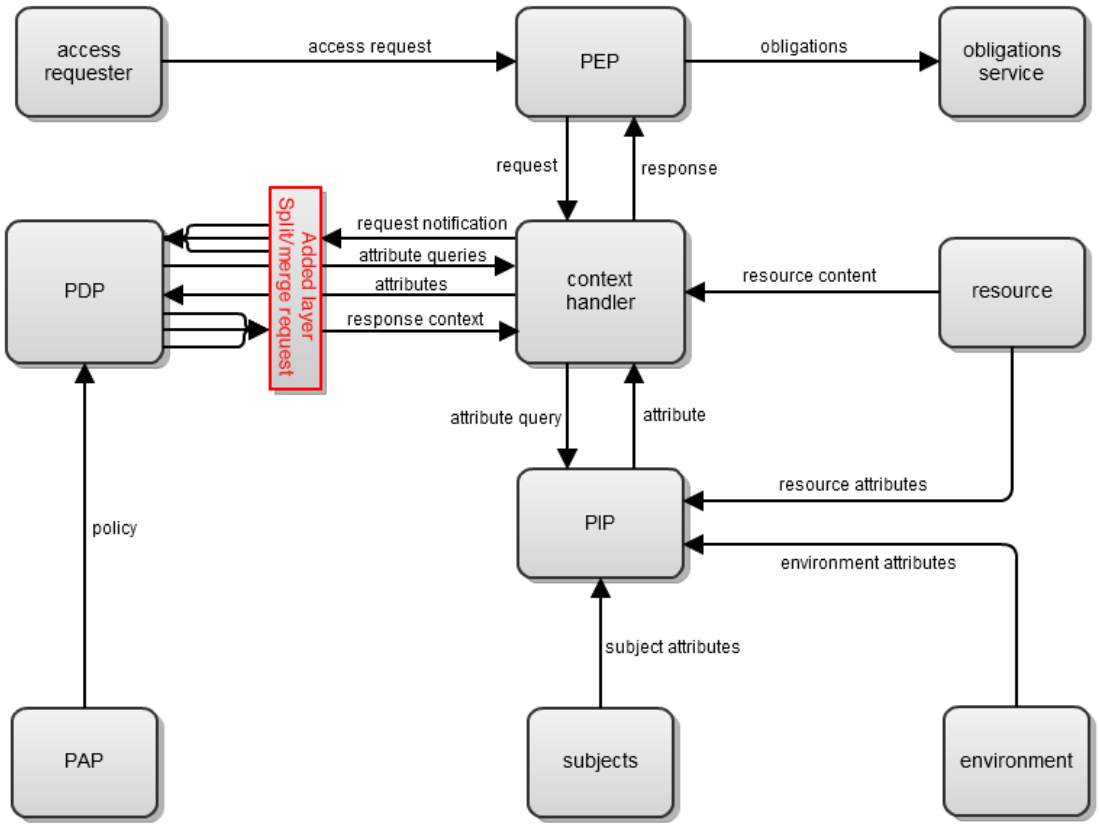


Figure 26 Contextual Attributes Solution

The additional layer (Figure 27) intercepts the requests and responses that are exchanged between the context handler and the PDP (1). The layer splits requests into one general context request and multiple single-context requests, based on the contextual part of the attributes (2). This context can for example be derived from the trial to which a resource belongs. It then adds the context of the single-context request to the environment element and sends these requests to the PDP (3). The PDP handles these requests and constructs a response based on context aware policies (4). The responses (5) are sent from the PDP to the additional layer where they are combined (6) to one general response using a combining algorithm. How this combining algorithm will work is still subject of research. This general response is finally sent to the context handler.

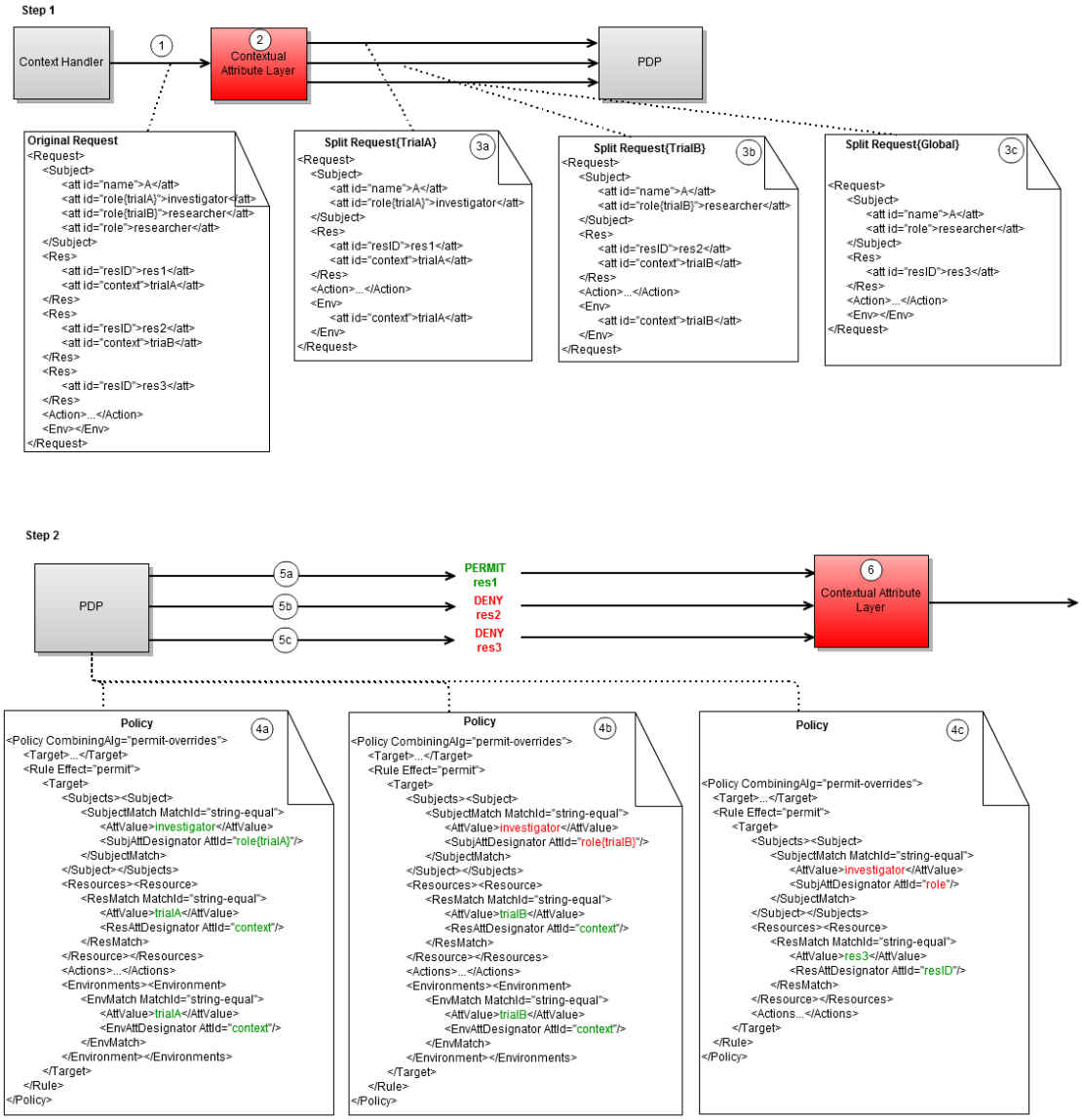


Figure 27: Contextual Attributes Example

The proposed solution for adding extended functionality to the XACML framework, introducing an XACML request proxy in front of the PDP, is generic (see e.g. next section). Multiple proxy layers can be added before the PDP. The added components manipulate the requests and pass them to the next layer, this process repeats until the request is sent to the PDP. This architecture looks promising because of its simplicity. However, further research is needed into the possible (unwanted) effects introduced by stacking different “functionalities”.

11.4.2 Vocabulary mapping

Vocabulary mapping maps terms written in a particular vocabulary to terms of another vocabulary. This mapping between different vocabularies can be accomplished by using for example ontologies. In ontologies, different vocabularies can be matched using ontology-based data matching strategies.

In INTEGRATE, vocabulary mapping will also be needed in the context of access control. Allowing the use of different vocabularies in the authentication platform, allows easier definition of policies[32]. By mapping the different attributes, the local entities can simply use their locally defined vocabularies while the INTEGRATE platform maps those to a common vocabulary. This means that policies can be written in a way that resembles natural language (instead of using a more technical centrally defined vocabulary).

The standard XACML specification does not define such vocabulary mapping functionality. An additional layer needs to be placed (similar to the contextual attributes solution discussed in 11.4.1) in front of the PDP, in order to provide this functionality. The general structure of this layer is shown in Figure 28. A request containing local vocabulary terms coming from the PEP is intercepted by the translation layer that is placed before the PDP. Each term (containing a key/value pair) found in the request is sent to a central translation service. This service interprets the given local vocabulary term and translates it to the corresponding term of the central vocabulary (using mapping tables). The resulting translated terms are placed in a new request which is send to and evaluated by the PDP.

[32] *I. Ciuci, B. Claerhout, L. Schilders, R. Meersman. 2011. Ontology-Based Matching of Security Attributes for Personal Data Access in e-Health*

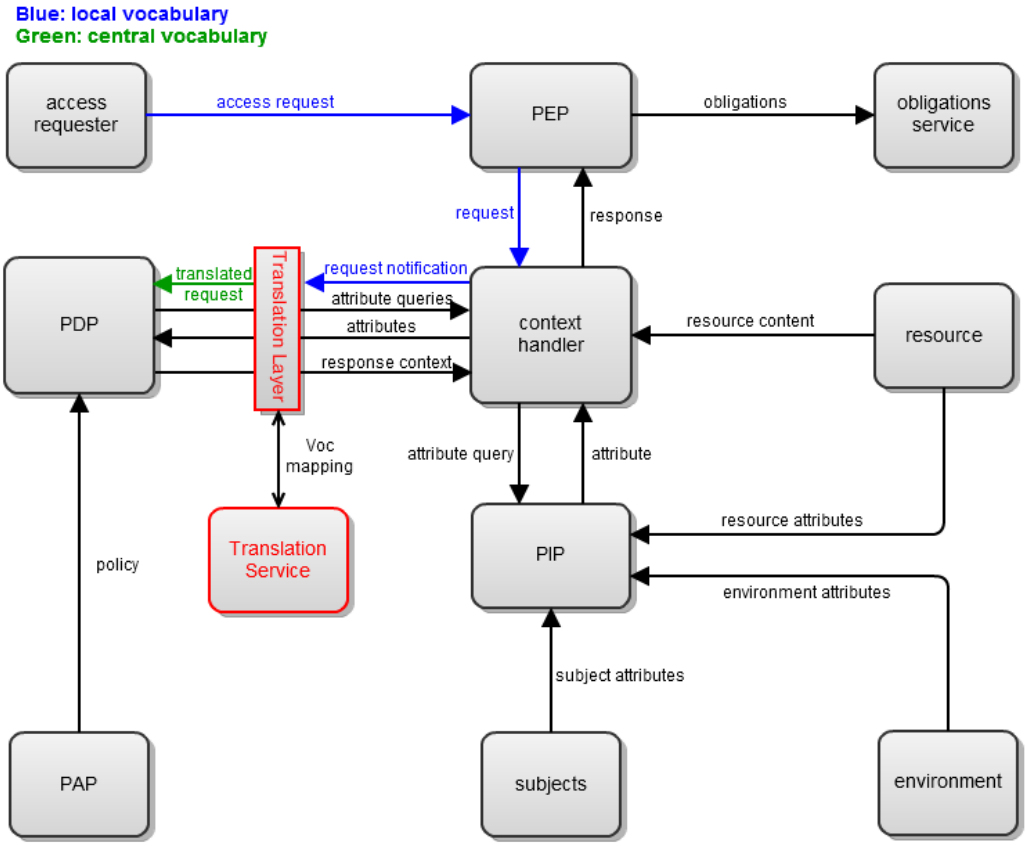


Figure 28: Vocabulary Mapping Solution

An example of this process is shown in Figure 29:
John is an oncologist in hospital A. He participates as an investigator in a certain trial. He wants to access Jane's files, a patient in this trial. He logs in as an oncologist (role=oncologist). The request containing this role-attribute is sent to the PDP. This request is captured by the translation module. This module translates the locally defined attribute “role=oncologist” to the central equivalent “role=investigator”. The requirements for access to the file are subsequently tested. The PDP observes that the person requesting access has the role of investigator and makes a decision.

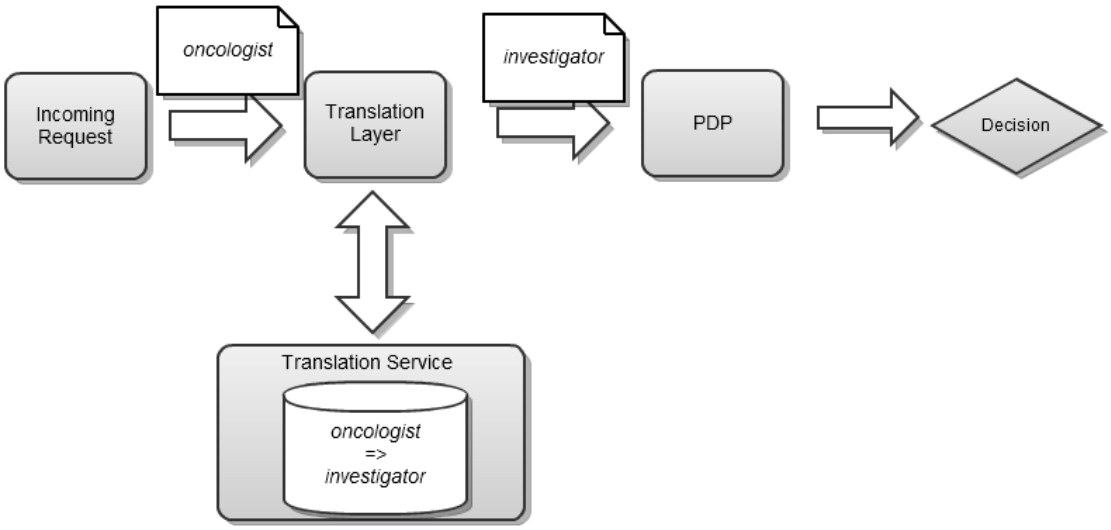


Figure 29: Vocabulary Mapping Example

The translation of the attributes allows for more generic policies. To clarify this, the previous example will be used. If there would not be a translation service, every policy concerning the investigator role would have to be copied (or extended) to represent the equivalence relation between the investigator and the oncologist. This creates an explosion in the size or amount of policies. Because more (or longer) policies need to be tested, the PDP is slowed down significantly. With the addition of this translation service, these problems are solved. When the oncologist role is encountered, it gets translated to the investigator role.

11.4.3 Endpoint security

Each main service (i.e. cohort selection, trial management, molecular screening, …) of INTEGRATE needs to be secured to meet the requirements of the legal framework defined in deliverable D1.3. Security is usually seen as "vertical", this means that security tends to be tightly coupled with the functionality of an application (penetrating each application layer). Most security aspects are built right into a solution. This usually implies that if one wants to change the access rules, the application code of the solution needs to be changed accordingly.

A better way of working would be to implement the security in a separate layer, as most services have similar security requirements, i.e. authentication, authorization, etc. Although the requirements that different services have towards such a layer are similar, it remains difficult to create application-independent, re-usable security components, because it is hard to find a generic mechanism for tying them into various applications, in particular with respect to content-related (i.e. fine-grained) access control mechanisms.

INTEGRATE however aims to provide security modules that create a minimal amount of overhead for the programmer and should be easily to integrate with the functional components (generic and lightweight components). More specifically, it should address the following concerns:

- Incorporation of the security modules does not require application programmers to code their business logic against a particular API in order to enable the security

framework (i.e. business logic does not need to incorporate dependencies on the security framework just to get it working);
- it should be straightforward to disable the security modules in order to facilitate functional testing (ideally by changing a single configuration item);
- a declarative style for enabling security features (*what* must it support) should be favored over a procedural approach (*how* must this be supported), e.g. using configuration property files or a domain specific language (DSL);
- convention should be favored over configuration, i.e. support common security requirements without having to write a lot of configuration (minimal security by default);
- the security configuration should be managed separately from the business logic (separate configuration files, class files, packages ...).

Integration of the authentication module is a prerequisite for using the other security modules in the framework. This module is responsible for verifying if a user has an active authenticated session running at the endpoint it is protecting. If not, the user is redirected to the IdP component where the proper credentials (supported by the IdP) need to be provided. If authentication at the IdP is successful, the user will be provided with an authentication token (proof-of-authentication) and directed back to the application where the authentication module will check the authentication token issued by the IdP to allow making further access decisions. Addressing the above concerns is relatively straightforward for the authentication module, provided the application can be easily split up in a *public* and a *private* zone (i.e. no application domain knowledge is required to determine whether authentication is required). In INTEGRATE we consider this to be the case for all main services, in particular because this is required for reliable audit logging. As such, a generic authentication component will be provided that can be easily integrated in the different endpoints (hooking into local session management) and ensures alignment of local user management with the INTEGRATE global identity management.

Protecting access for “use of a service” (i.e. the resource being protected is the service invocation itself) is relatively easy. This functionality can be provided though a proxy-style PEP (Figure 30), which is put in front of the actual service end-point (i.e. the implementation), provides the same interface and simply delegates the call to the underlying implementation if access is granted[33]. When the user tries to access the service, the PEP intercepts the request. The PEP module queries the Policy Decision Point (PDP) to check if the user has the required access rights to use the component. The PDP makes the decision and sends the response back to the PEP. When access is granted, the PEP allows the user to access the actual endpoint implementation. Note that this proxy-style PEP is also perfectly capable of dealing with access rules that depend on the parameters supplied in the call. In general, the proxy-style PEP can enforce access control as long as the access control policies can be fully evaluated before the service is executed. In the first stage iteration, the INTEGRATE security implementation will mainly concentrate on a generic proxy-style PEP approach.

[33] Note that proxy-style PEPs can be embedded in the application itself or can be deployed as a separate application component (on the network level). The former approach requires compatibility on the technology level (e.g. Java bytecode on the Java platform) while the latter approach requires compatibility on the interface level (e.g. SOAP web services). Within INTEGRATE we will initially be focusing on the former approach. We may extend our work to the latter approach if sufficient grounds for re-use of such a component can be found.

In a next step, we will also tackle access control policies that can only be evaluated after the call to the endpoint implementation is allowed (post invocation), i.e. those requiring inspection of the resource(s) contained in the service invocation response. Note that proxy-style post invocation interception by the PEP is only suitable for those operations that do not trigger side-effects (unless these side-effects are performed on resources that are managed by a transaction that can be rolled back and the policy enforcement can be included in the transaction context).

As an example of the above situation, imagine a service provided by a datawarehouse that list all datasets available to the user. Typically, the application (service implementation) itself filters which patients are visible to the user. In a centrally managed framework, this filtering needs to be done based on globally managed (and dynamically changeable) policies. Unfortunately, XACML policies do not lend themselves to answering questions like “to which datasets does user U have access”. XACML policy evaluation requires a “Y/N” question. Hence for each available dataset, the question needs to be asked: “does user U have access to dataset Y”. This inherent characteristic makes it difficult to provide a generic mechanism which introduces a clean separation between functionality (the query) and the access control (filter).
While this example can be covered by a proxy-style PEP, a serious performance penalty can be expected: the result set will typically contain a lot of results that are not relevant for the requesting user (this boils down to bypassing the query engine efficiency).

Finally, there may be services for which access control policy evaluation requires a level of granularity beyond the level of the resource being protected and interpretation of the service invocation response is insufficient because it does not provide enough information for policy evaluation. In that case, one or more PEPs will need to be interwoven within the application being protected at the right level of granularity, e.g. in the form of aspects (a concept used in Aspect Oriented Programming for organizing cross-cutting concerns).

It should be noted that the latter situations are typically consequences of dealing with legacy applications or application components. As such, one should consider first whether it makes sense to refactor those legacy applications into a set of more loosely coupled services (i.e. migrate to a higher level of service-orientation) as this provides new opportunities for service invocation interception in the context of access control enforcement. This approach may even provide additional benefits such as an increased level of reuse of common functionality.

In the initial implementation of INTEGRATE, this problem will not be addressed by a generic solution, but rather in the classical way (tight integration). However, the solution should guarantee that access decisions themselves are conform the centrally managed policies (which can change dynamically). The above observations apply equally to audit logging, and thus a similar approach will be taken for audit log integration.

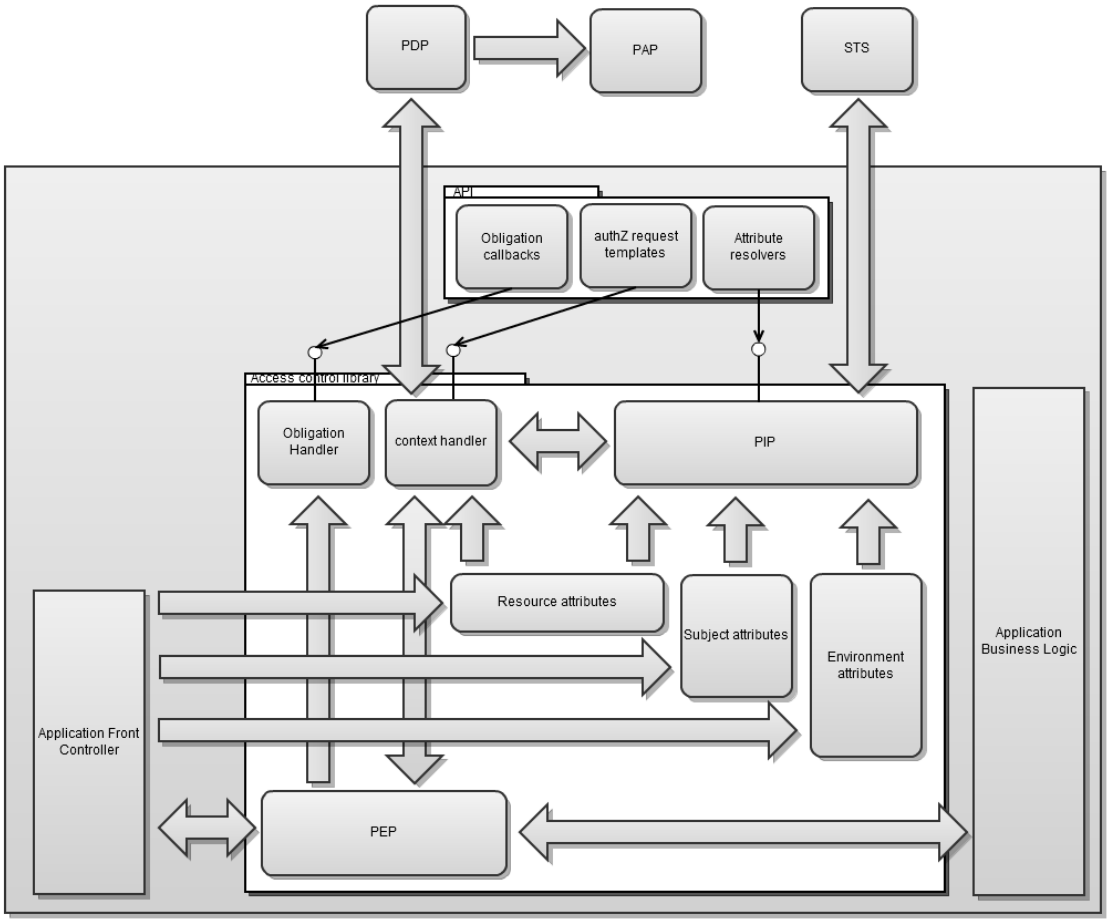


Figure 30: Endpoint Security

11.4.4 Other

To facilitate the process of making access control decisions, the IdP not only provides authentication, but also acts as a PIP to provide additional attributes to the policy decision process. This information is typically identity information such as address, date of birth, etc.

11.5 Summary

The security implementation within the INTEGRATE project needs to be modular and re-usable. A modular approach facilitates re-use and extension of the security framework. The two main issues that are dealt with are authentication and authorization. The authentication solution will be based on SAML and more specifically on Shibboleth; an architecture and implementation of a federated Single Sign-On authentication and authorisation infrastructure heavily coupled with SAML[34].

The main challenge of the INTEGRATE project is authorization. The INTEGRATE requirements relating to authorization encompass a number of different challenges. A first issue that needs to be dealt with is the integration of consent within authorization. Another problem arises when dealing with the multiple contexts (e.g. the same person can have different roles within different trials). Dealing with these roles is difficult when using standard the standard XACML RBAC implementation. The proposed solution is to add an extra component to between the context handler and the PDP, this component handles this role issue. A final challenge consists of providing a service that maps terms specific to the PEP vocabulary to terms of the vocabulary specific to the PDP. Like for the contextual attributes solution an extra component will be place between the context handler and PDP.

[34] more information in deliverable D2.2: Inventory of reusable/available relevant solutions and components

12 (PART III) Implementation Status

12.1 Introduction

As part of the D2.4, a status update on the implementation work is given. INTEGRATE takes an iterative approach towards implementation, which means that initial implementations aim at demonstrating large parts of the integrated system, be it with limited functionality (and stubs). Functionality is to be added and refined through a cyclic process (agile development principles). This approach is especially useful in research as it allows access to prototypes at an early stage and allows for radical design changes up until a late stage. It also enables researchers to experiment, test and extend components without disrupting the rest of the system.

Two main architectural blocks (closely linked with the scenario's of D1.2) discussed in the initial system architecture (see above) were selected for the demonstrator:

- An initial (simplified) version of the molecular testing, discussed in 6.1 of this document.
- An initial version of the analytical tools and sharing of predictive models, discussed in 0.

The next sections describe the main goals, approaches and views of both blocks.

12.2 Molecular Testing Scenario Demonstrator

12.2.1 Goal

This demonstrator will provide a first implementation of the molecular testing scenario specified in D1.2. Although a simplified version of this scenario will be implemented, the main flow remains untouched, i.e. an investigator should be able to check (by using a set of trial criteria) whether a selected patient is eligible for a selected trial. The project aims to demonstrate the system described in this section at the first year review (May 2012).

The goal of this integrated demonstrator is to evaluate the general approach to semantic data linkage chosen in INTEGRATE. Many factors (choice of common data model, core data set, semantic reasoner, model of clinical trials,...) determine the practical capabilities and limitations of the final solution. By building an integrated prototype and testing it in a realistic scenario, we will gain better insight into the real impact of the different design decisions made. This not only allows partners responsible for a component to improve their own design, but also to better understand the impact of their components on other partners' work. Furthermore, aiming for early integration allows for quick feedback from end-users (and increases their involvement in the development process).

12.2.2 Approach

Because INTEGRATE opted for this agile style of development focusing primarily on the feasibility of the integrated solution, rather than on producing one or two finalised sub-components, a pragmatic approach to build the first iteration of the molecular screening scenario is required.

For this first iteration a number of S&T (science and technology) short-cuts have been taken (see for example the approach storing the rules for criterion matching explained

further). If the chosen approach appears to lead to success, these shortcuts will gradually be replaced by more functional or more generic solutions in the next iterations.

12.2.3 High Level Overview

In the functional view of the molecular testing scenario, the different interconnected components are listed (Figure 6 in part 1). In the demonstrator only a subset of these components is used. Figure 31 gives a high-level overview.

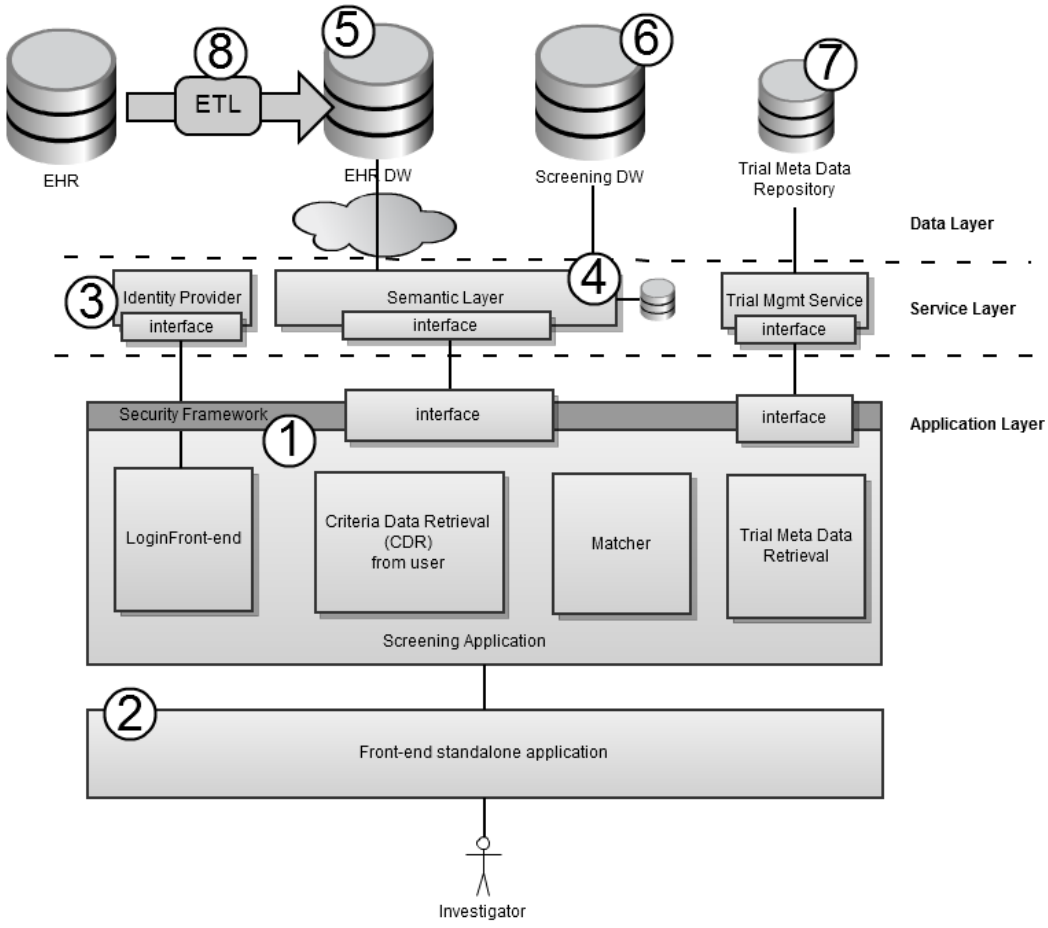


Figure 31: Informal Overview of the Molecular Testing Demonstrator

A major component of the demonstrator is the screening application itself (1,2 and 3 on the figure), which is developed as a standalone client[35]. Specific effort is allocated to designing a GUI (2) in such a way that it offers an intuitive and user-friendly visualisation of controls and information. The envisaged user-friendliness provided by this GUI (especially when it comes to quick assessment of results) should lead to easier acceptance and increased productivity for the end users. The client application

[35] The decision between standalone client or web-application is mainly dominated by the graphical functionality requirements of the innovative GUI in view of the overall technology choice (Java). The standalone client approach could still be changed, this however does not change the principles described in this document.

relies heavily on external services which require authentication (see 9.1), the main application therefore needs the necessary components to authenticate a user with the central identity provider and forward the obtained credentials to the respective services (3). Note that in the first year demonstrator, the security components are not integrated.

The main application itself groups a collection of sub-components that are connected and interacting with each other:

- Criteria data retrieval (CDR)
  - This component is responsible for obtaining information necessary for the evaluation of a (certain type of) criterion directly from a user for storage into the screening DW (electronic data capture).
- Trial meta-data retrieval
  - this component serves as an interface to the central trial management service, in which the meta-data of all available trials is stored. This includes the trial criteria.
- Matcher
  - The matcher evaluates the matching rules of the criteria. The basics of operation are explained further.

The subcomponents of the screening application interact with services provided by the central platform:

- Trial management service (see 6.1.1)
  - The trial management service manages trial information and as such exposes the trial meta-data.
- Semantic layer (see 6.3)
  - This layer combines all components and services required for the semantic integration (reasoning, mapping, ...). It exposes a service that abstracts the underlying data sources and presents data to applications according to a single integrated data model (the INTEGRATE common information model). In the demonstrator two data sources are present: an instance of a local EHR data source and the platform's central screening datawarehouse.
- Identity provider (see 9.1)
  - The service provider responsible for the authentication of users to the screening application. Excluded from the demonstrator.

There are two datawarehouses involved in the demonstrator. Both datawarehouses are based on the INTEGRATE common data model:

- Local EHR DW
  - This datawarehouse contains a frequently updated export of the local (= site to which the investigator belongs) EHR. The export process (ETL process, (8) on the figure) includes transformation of the data to the INTEGRATE common data model. For the demonstrator, this process will not be automated. The EHR DW will be pre-loaded with EHR data.
- Screening DW

- A central component of the platform that stores all data collected during the screening process. The idea is that as much as possible, data relevant for the evaluation of the criteria is copied into the screening DW. This promotes the screening DW to become a central reference for the screening data and facilitates the export of the screening data into the research environment.

The simplified application flow is straightforward (cf. molecular testing scenario in D1.2):

1. An investigator logs in (at his local site)
2. The investigator submits a specific patient for screening
3. The investigator selects a number of trials for which eligibility needs to be checked
4. The matcher is run based on data present in the local EHR, the central screening datawarehouse and direct data entry by the user.

12.2.4 Information View

Figure 32 gives an overview of the main information models that will be used for the molecular testing demonstrator. On the left side the model of the semantic layer is given while on the right the trial management meta model is given.

In the semantic information model, the data of the screening datawarehouse is structured according to the common data model that is RIM based. The semantic layer itself exposes data according to the central information model.

In the trial meta-data information model, the trial meta-data repository contains data that is structured according to the model explained in 7.1 .

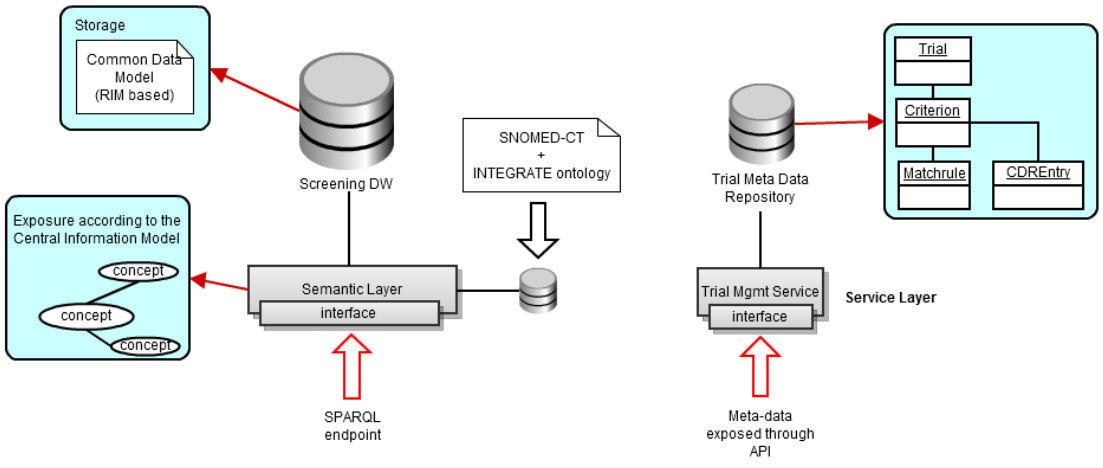


Figure 32: Molecular Testing Demonstrator Information View

12.2.5 Matcher

12.2.5.1 Matching rules and scripts

The agile development approach (aiming to demonstrate the integrated system) relies on early conceptual implementation of the most complex components and a gradual refining in the subsequent development iterations. A major simplification made in the demonstrator is in the modelling of criteria for use by matcher and retrieval of the criteria relevant data by EDC. Figure 33 shows the simplified model (descending from "trial" as specified in the trial meta-model see Figure 14).

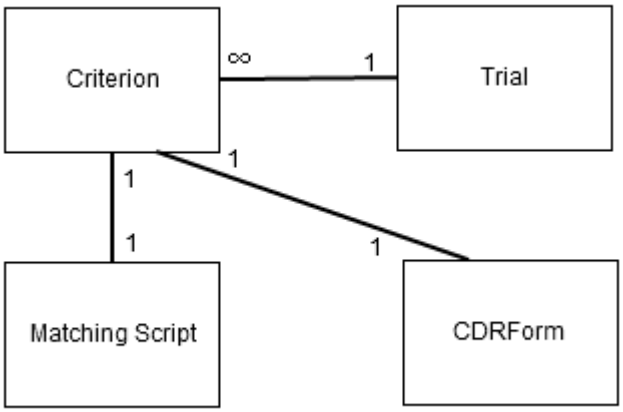


Figure 33: Simplified Criterion Model

In this model a criterion is directly linked to a matching rule. In the current implementation, this matching rule is a script (literally a piece of executable code) which can be executed by the matcher. This simplification does not conflict with the parallel work on formalization of clinical trial criteria. In the end, the formalization will allow the "matching scripts" to be automatically generated (be it up-front or at runtime). Due to the appropriate decomposition, this subcomponent can be further developed (upgraded with the results of the results of criterion formalisation) with minimal impact on the rest of the implementation.

A matching script checks for a specific patient whether he/she matches this rule. The rule has the following arguments:

Argument	Description
patient-id	Patient identity
data-source	Targeted data source on which the rule will be run. The rule will be evaluated using data from this specific data source. At this point in time the semantic layer does not support federation, hence only one data source can be targeted at a time.

And returns the following information:

Return	Description
Matchresult	One of three (self-explanatory codes): MATCH,NONMATCH, UNDETERMINATE

Evidence	The information (according to the CIM) which has been relevant to the evaluation of the rule, in case there was a MATCH or NONMATCH. This "evidence" can then be used for: • copying relevant data into the screening DW, so that this effectively acts as a central storage for screening related data. • rendering decision information in the GUI.

An example of such a script is given below:

```
// Note: The URLs are URIs to identify relations and entities on the
ontology. The numbers reference SNOMED codes:
// 118522005 = ID, 116154003 = Patient, 184099003 = Date of birth,
439401001 = Procedure

def matchRuleScript(patientID, targetID) {
    def result = new QueryResult();
    def query = new SPARQLQuery("SELECT ?patient ?birthDate WHERE { \
            ?patient a <http:/fp7-
integrate.eu/coredataset/116154003>. \
            ?patient <http:/fp7-integrate.eu/coredataset/118522005>
${patientID}. \
            ?patient <http:/fp7-integrate.eu/coredataset/184099003>
?birthDate. }");

    if(query.execute(targetID) == null)
        {   result.resultcode = UNDETERMINATE }
    else {
        def DOB = query[0][1];
        def age = CalculateAge(DOB);
        result.addEvidence("birthDate", DOB);
        if (age>18 && age<70) {
            result.resultcode = MATCH
        } else {
            result.resultcode = NONMATCH
        }
    }

    return result;
}
```

Code 1: Example script

Each criterion has an implicit "Y/N version" matching rule associated with it, regardless of the existence of a matching script. For each criterion, it can be explicitly recorded in the screening DB if a patient matches the criterion. The implicit "Y/N version" matching rule allows to check this result.

This system is introduced to cover for criteria which are impossible to automatically check. There are different reasons why a criterion could not be automatically validated. Next to the obvious reason that criteria can simply be too complex to be modelled in the current CIM and semantic layer implementation, there are also the issues which can never be (fully) solved, e.g. "absence" of a medical fact (is it really absent or just

missing from the information source) or criteria which do not relate to the recorded data (e.g. "The patient does not plan to move further than x km from the hospital in the next 2 years.") and are subjective.

12.2.5.2 Data Retrieval

The same simplified approach as with the matching rules is taken for the data entry of missing information (see Figure 33). Linked to each criterion, there is a form (cf. CDRForm) that can be rendered by the application. This form can be used to retrieve extra information relevant for the evaluation of the criterion. Data is recorded according to the CIM, so that these can be easily stored in the screening DW (for further processing).

As with the matching rules, there is an inherent "Y/N version", which renders the literal criterion and takes "Matches" or "Does not match" as input from the user, and stores it as such.

12.2.6 Matcher Flow

The matcher is the top level component that matches a selected patient with a set of selected trials, based on the available information on that patient, and the individual eligibility criteria recorded for each of the trials. It is basically a workflow component that makes use of the criteria matching scripts and data retrieval components. It is important to understand that the complexity is in the sub components and the flow itself can easily be changed.

A conceptual flow for checking a single criterion is shown on Figure 34. The actual implemented flow will deal with multiple criteria from multiple trials and with ways to optimize the evaluation order for e.g. reduced manual data entry. This is however out of scope of this section aiming only to clarify the principles of operation.

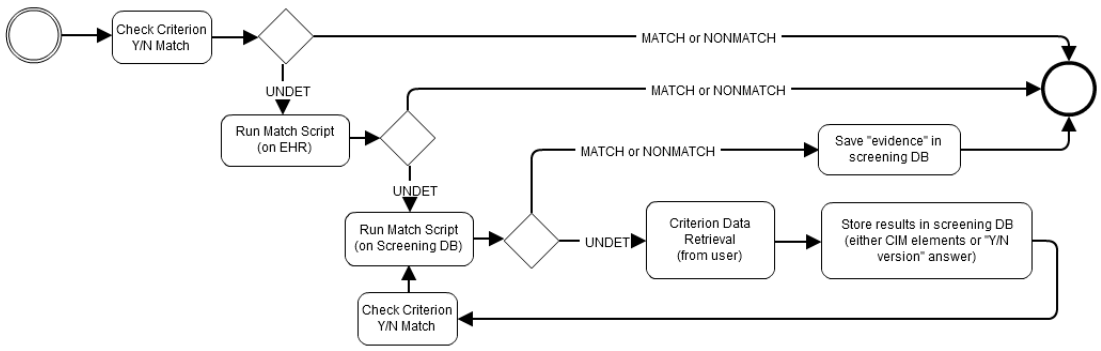


Figure 34 Checking a Single Criterion Flow

- First off all, the matcher checks whether the there is a "Y/N version" answer available for the criterion.
- When no such answer can be found, the match result will be UNDETERMINATE.

- Subsequently the match script downloaded from the trial management service is executed on the (local) EHR data if access to the local EHR is available. If not, this step can simply be skipped.
- In the case that there is a definite MATCH or NONMATCH, the "evidence" returned by the script is saved in the screening DW. When the criterion matching script is now run on the screening DW, it should return the same result (unless it is time dependent).
- In the case that the script cannot determine a match (returning UNDETERMINATE), the user will be asked for the information required for evaluating the criterion. The matcher will do so by presenting the CDR Form linked to the criterion in the trial management service. The user has the option to answer the "Y/N version" of the criterion (if no form is available, the "Y/N version" will be immediately presented). The entered results are stored in the screening DW, and the "matching loop" repeated, which will now end in a MATCH or NONMATCH result.

12.3 Analytical Tools & Sharing of Predictive Models Demonstrator

12.3.1 Goal

This demonstrator will provide a first implementation of the overall statistical analysis and predictive modeling, as clearly defined in D.5.1.

The goal of this process is to demonstrate a simplified version of the set of statistical tools and predictive models that will be allocated to the INTEGRATE analysis platform, responsible for analyzing any clinical, genomic and imaging data that are stored in the INTEGRATE data-warehouses. These tools and models are the main analysis components found in the web-based platform, related to a set of specific research questions that need to be answered (i.e. identify a gene expression signature that predicts the tumor response to a specific drug used across multiple breast cancer neo-adjuvant trials).

12.3.2 Approach

The demonstration of the analytical tools and predictive models follows a simplified developing architecture compared to the one specified in the use cases scenarios (see D.2.4). In its formal version, data retrieving will be performed by querying requests from the platform to the INTEGRATE data-warehouses; models and tools will be provided by exposed interactions between the platform and the model repositories; the final analysis report will be accessed via the platform for evaluation and download.

Because the web-based architecture of the INTEGRATE analysis platform is still under development, the demonstration for the first review will be based on a framework where the data-warehouses and the model repositories are placed locally on a stand-alone computer and all the interactions between them are simulated by local transactions between the storage places. Finally, all the software scripts related to the several analysis scenarios will be executed in batch mode, assuming that a virtual user requests a specific analysis to be implemented via the INTEGRATE analysis platform which is replaced, for demonstrative purposes, by a stand-alone computer.

12.3.3 Overview of the Analysis architecture

The architecture of the demonstrator version of the analytical tools will be (as mentioned previously) simplified. The components that will be implemented in the demonstrator version are listed in Figure 35. The tools and models repository and the data-warehouses are simulated by local storage places on the stand-alone computer. The "local storage of models and tools" is divided into several sub-storage places; places for storing publicly available libraries used in our scripts and the scripts that are related to a specific analysis scenario, respectively. Following the same structure, the eligible dataset(s) that are queried from the INTEGRATE datawarehouses, are stored into separate storage folders (i.e. folders with path names of the form "Datawarehouse/Scenario1", etc.).

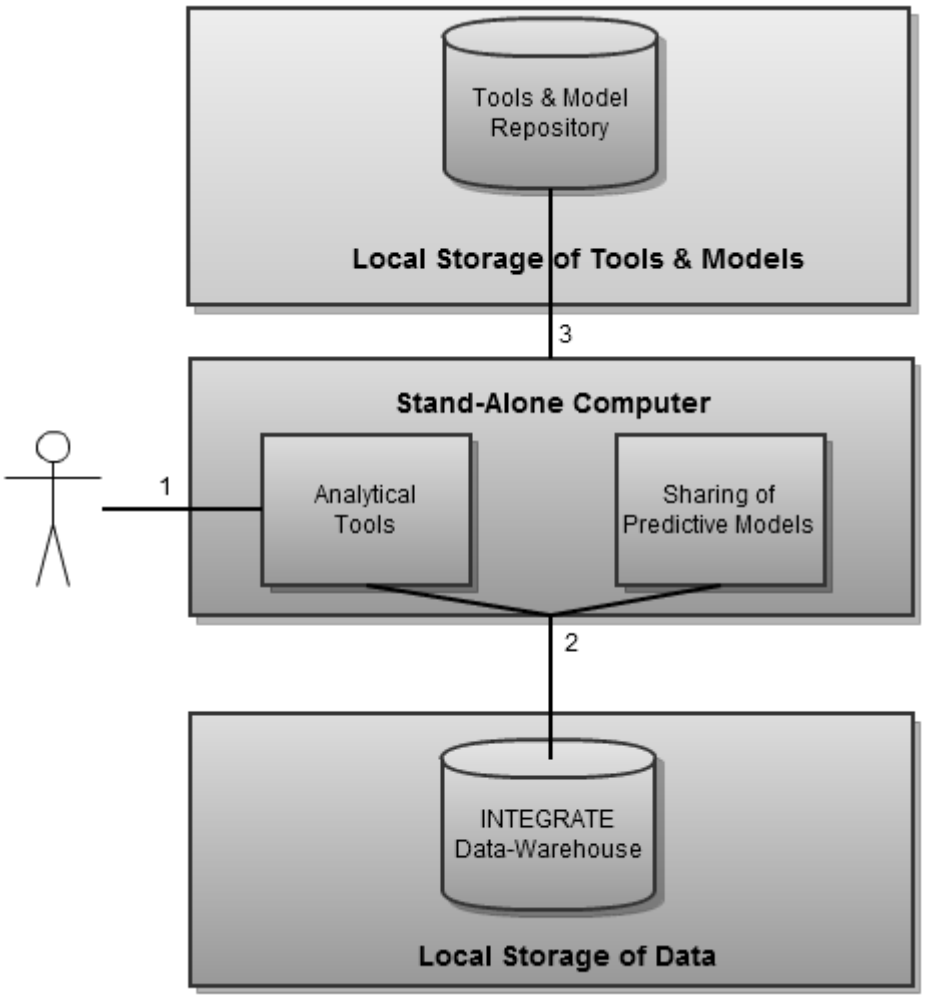


Figure 35: A simplified version of the analytical tools and predictive modeling architecture

The simplified application flow is straightforward:

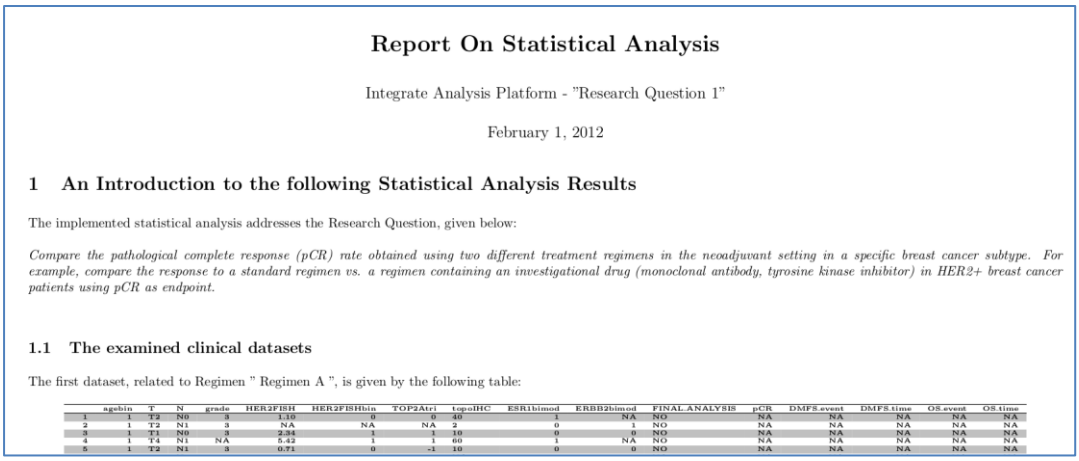
1. A virtual user selects that he wants to run a script that answers a specific research question (i.e. assess whether pathological complete response is a valid surrogate marker for disease free survival and overall survival).

2. A script, related to the selected question, is running in batch mode. All scripts are taking information regarding the name of the dataset(s) and their storage location as input. An example script, running in batch mode, is given below. The script takes as input the name of the two different regimens (regimen "A" and "B", respectively) and their location full path.

```
gmanikis@gmanikis-VirtualBox:~$ sudo R --slave --args 'Regimen A' /home/gmanikis/my_r_files/DataWarehouse/
Scenario2/StandardRegimen.txt 'Regimen B' /home/gmanikis/my_r_files/DataWarehouse/Scenario2/Investigationa
lRegimen.txt  < /home/gmanikis/my_r_files/ResearchQuestion_2.R
```

Figure 36: A script running in batch mode

3. The script integrates the coding lines into Latex documents and dynamic reports are created. These reports can be updated automatically if the input data or the analysis results change. The analysis report, consisting of figures, tables and short descriptions of the analysis results and is exported as a PDF file. A short preview of the analysis report is given in the following snapshots.



Report On Statistical Analysis

Integrate Analysis Platform - "Research Question 1"

February 1, 2012

1 An Introduction to the following Statistical Analysis Results

The implemented statistical analysis addresses the Research Question, given below:

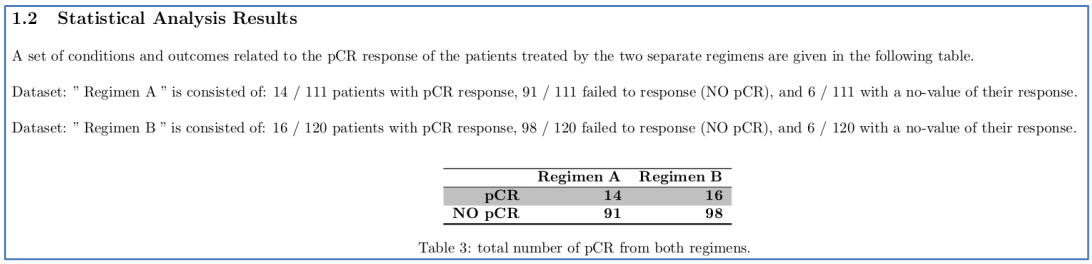
Compare the pathological complete response (pCR) rate obtained using two different treatment regimens in the neoadjuvant setting in a specific breast cancer subtype. For example, compare the response to a standard regimen vs. a regimen containing an investigational drug (monoclonal antibody, tyrosine kinase inhibitor) in HER2+ breast cancer patients using pCR as endpoint.

1.1 The examined clinical datasets

The first dataset, related to Regimen " Regimen A ", is given by the following table:

	agebin	T	N	grade	HER2FISH	HER2FISHbin	TOP2Atri	topoIHC	ESR1bimod	ERBB2bimod	FINAL.ANALYSIS	pCR	DMFS.event	DMFS.time	OS.event	OS.time
1	1	T2	N0	3	1.10	0	0	40	1	NA	NO	NA	NA	NA	NA	NA
2	1	T2	N1	3	NA	NA	NA	2	0	1	NO	NA	NA	NA	NA	NA
3	1	T1	N0	3	2.34	1	1	10	0	0	NO	NA	NA	NA	NA	NA
4	1	T4	N1	NA	5.42	1	1	60	1	NA	NO	NA	NA	NA	NA	NA
5	1	T2	N1	3	0.71	0	-1	10	0	0	NO	NA	NA	NA	NA	NA

Figure 37: Snapshots of an Analysis Report (1/3)



1.2 Statistical Analysis Results

A set of conditions and outcomes related to the pCR response of the patients treated by the two separate regimens are given in the following table.

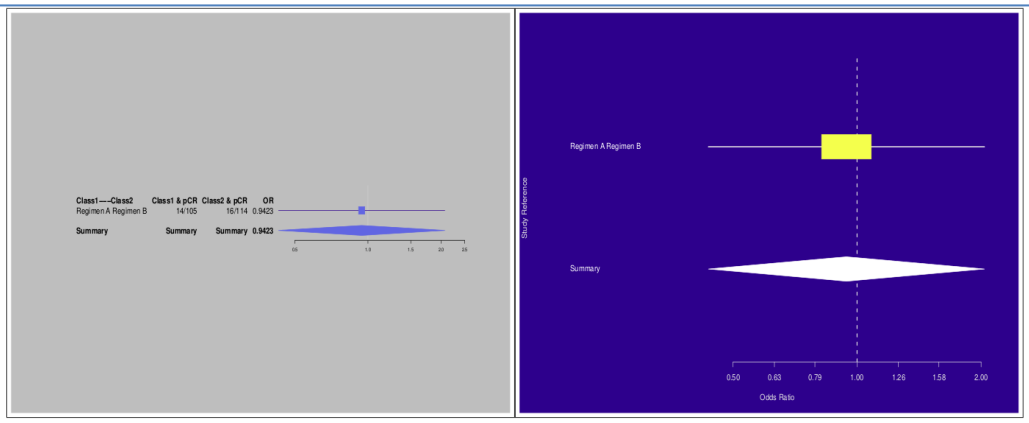
Dataset: " Regimen A " is consisted of: 14 / 111 patients with pCR response, 91 / 111 failed to response (NO pCR), and 6 / 111 with a no-value of their response.

Dataset: " Regimen B " is consisted of: 16 / 120 patients with pCR response, 98 / 120 failed to response (NO pCR), and 6 / 120 with a no-value of their response.

	Regimen A	Regimen B
pCR	14	16
NO pCR	91	98

Table 3: total number of pCR from both regimens.

Figure 38: Snapshots of an Analysis Report (2/3)



(a) Forest Plot+Info (b) Forest Plot

Figure 1: Forest Plots for the Clinical Data

Therefore, patients treated with " Regimen B " respond 1.061224 times more often than patients who received regimen " Regimen A "

Usefull Notes:

How odds ratio results are interpreted: An OR of 1.00 means that patients treated by the first or the second regimen were equally likely to response (pCR). An OR higher than 1 means that the first group was more likely to experience the event (pCR) than the second group. An OR of less than 1 means that the first group was less likely to experience the event.

Figure 39: Snapshots of an Analysis Report (3/3)

12.4 Summary

The framework described above is introduced for the demonstration needs of the first review, the exact content of it will be defined collectively by the consortium before the review meeting. This however does not change the principles and the general application flow described in the document.

13 Glossary

Abbreviation	Description
ABAC	Attribute-Based Access Control
BRIDG	Biomedical Research Integrated Domain Group
CDM	Common Data Model
CDMS	Clinical Data Management System
CIM	Common Information Model
CRF	Case Report Form
CT	Clinical Trial
DICOM	Digital Imaging and Communications in Medicine
DW	Datawarehouse
EHR	Electronic Health Record
ETL	Extract Transform Load
GUI	Graphical User Interface
IC	Informed Consent
IdP	Identity Provider
IRB	Institutional Review Board
LDAP	Lightweight Directory Access Protocol
MRI	Magnetic resonance imaging
OWL	Web Ontology Language
PAP	Policy Administration Point
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIP	Policy Information Point
RBAC	Role-Based Access Control
SAML	Security Assertion Markup Language
SLO	Single Log-Out
SNOMED	Systematized Nomenclature of Medicine
SNOMED-CT	Systematized Nomenclature of Medicine - Clinical Terms
SOAP	Simple Object Access Protocol
SSO	Single Sign-On
STS	Security Token Service
TTP	Trusted Third Party
XACML	EXtensible Access Control Markup Language
XML	EXtensible Markup Language

14 Table of Figures