Deliverable No. 3.2

Initial System Architecture

Grant Agreement No.:	270089
Deliverable No.:	D3.2
Deliverable Name:	Initial System Architecture
Contractual Submission Date:	30/01/2012
Actual Submission Date:	30/01/2012

Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	X
CO	Confidential, only for members of the consortium (including the Commission Services)	

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	***p-medicine***
Project Full Name:	From data sharing and integration via VPH models to personalized medicine
Deliverable No.:	D 3.2
Document name:	Initial System Architecture
Nature (R, P, D, O)[1]	R
Dissemination Level (PU, PP, RE, CO)[2]	RE
Version:	1.5
Actual Submission Date:	10/04/2012
Editor: Institution: E-Mail:	Manolis Tsiknakis FORTH tsiknaki@ics.forth.gr

ABSTRACT:

This deliverable aims to define the initial version of the architecture of the p-medicine platform. Additionally it presents the rationale and the process for designing the architecture based on the requirements and the user scenarios of the project. This effort is focused on the identification of the major stakeholders, their concerns, and viewpoints following well-known best practices for documenting software architecture. Another goal of the current deliverable is to present an Architectural Description (AD) document, that defines the high level architecture of the platform. The architecture definition process is nevertheless a continuous task as the system evolves and new requirements arise or other issues emerge. This document will therefore similarly evolve and new versions of it will be articulated in the course of the project.

KEYWORD LIST: Architecture, Architectural Description, Views, Viewpoints

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 270089.

[1] **R**=Report, **P**=Prototype, **D**=Demonstrator, **O**=Other

[2] **PU**=Public, **PP**=Restricted to other programme participants (including the Commission Services), **RE**=Restricted to a group specified by the consortium (including the Commission Services), **CO**=Confidential, only for members of the consortium (including the Commission Services)

MODIFICATION CONTROL			
Version	Date	Status	Author
0.1	15/12/2011	Draft	Manolis Tsiknakis
0.2	09/01/2012	Draft	Stelios Sfakianakis
0.4	20/01/2012	Draft	Stelios Sfakianakis
0.7	08/02/2012	Draft	Manolis Tsiknakis
0.8	20/02/2012	Draft	Stelios Sfakianakis
1.4	09/04/2012	Pre-final	Stelios Sfakianakis
1.5	10/04/2012	Final	Manolis Tsiknakis

List of contributors

- Manolis Tsiknakis, FORTH
- Stelios Sfakianakis, FORTH
- Giorgos Zacharioudakis, FORTH
- Lefteris Koumakis, FORTH
- Haridimos Kondylakis, FORTH
- Elias Neri, Custodix
- Fatima Schera, FhG
- Gabriele Weiler, FhG
- Michael Mock, FhG
- Marie-Luise Christ-Neumann, FhG
- Benjamin Jefferys, UCL
- Georgios Stamatakos, ICCS
- Fay Misichroni, ICCS
- Dimitra Dionisiou, ICCS
- Dawid Szejnfeld, PSNC
- Juliusz Pukacki, PSNC
- Anuj Sharma, Biovista
- Anca Bucur, Philips
- Aisan Maghsoudi, Philips

Contents

Figures

1 Executive Summary

The aim of this deliverable is to define the initial version of the architecture of the p-medicine platform. The goals of the p-medicine project are quite challenging and therefore the design of system architecture is a complex task taking into account multiple, often mutually conflicting, requirements. To this end, this document presents the rationale and the process for designing the architecture based on the requirements and the user scenarios of the project. This effort is focused on the identification of the major stakeholders, their concerns, and viewpoints following well-known best practices for documenting software architecture. The primary goal of the current deliverable is to present an Architectural Description (AD) document, that defines the high level architecture of the platform. The architecture definition process is nevertheless a continuous task as the system evolves and new requirements arise or other issues emerge. We expect therefore to revisit this document and enhance it based on the new requirements and challenges emerged and on the technical solutions that the development team of the project suggests to address them.

In the first section of this document we provide an introductory overview of the approaches used for the architecture definition process. The subsequent section represents the initial “Architecture Description” document for the p-medicine platform.

2 Introduction

Purpose of this document

The purpose of the p-medicine platform is to provide an infrastructure where physicians are supported in decision-making and in delivering individualized treatments to patients by exploiting the vast amount of heterogeneous multilevel biomedical data. For the realization of this vision new software, services, tools and models need to be in place that will support physicians in their daily care of patients. On the other hand nowadays we are facing a paradigm shift in medicine, going from hospital and clinical based care to a new standards approach, where the patient is also given a primary role in the delivery of care. The healthcare patient empowerment is therefore an additional dimension that the p-medicine platform endeavours to achieve.

The most important deliverable of the project is the definition and implementation of a software platform that will support the above mentioned objectives. This platform needs to be described in terms of its functionality, quality characteristics (e.g. security, performance), technical and implementation related properties (e.g. communication protocols, programming environments), deployment and operational attributes, etc. It is common to aggregate all these and even more aspects of a system under the term “system architecture”. Also as it often said, “every system has an *architecture*, whether understood or not; whether recorded or conceptual” [7]. But why do we need to define such a system’s architecture and document it in an *architecture description* document?

Over the past few decades, the complexity of software systems has increased substantially. The software architecture’s primary concern is to address this complexity in the systems design, building, documentation, and maintenance. This complexity presents itself in two primary guises [2]:

- Intellectual intractability. The complexity is inherent in the system being built, and may arise from broad scope or sheer size, novelty, dependencies, technologies employed, etc. Software architecture should make the system more understandable and intellectually manageable by providing high level abstractions that hide unnecessary detail, providing unifying and simplifying concepts, logically decomposing the system into sub-systems or layers, etc.
- Management intractability. The complexity lies in the organization and processes employed in building the system, and may arise from the size of the project (number of people involved in all aspects of building the system), dependencies in the project, use of outsourcing, geographically distributed teams, etc. Managerial complexities in a project management are mainly due to two important but conflicting interests; 1) maximize product goals, and 2) minimize resource used. Software architecture should make the development of the system easier to manage by enhancing communication, providing better work partitioning with decreased and/or more manageable dependencies, etc.

The common approach as hinted above in order to deal with all this complexity is the “divide and conquer” strategy where the problem(s) is broken down into two or more sub-problems that are easier to deal with, or are subject to the same strategy otherwise. But of course this decomposition of a system into smaller, more easily manageable entities requires answering questions like the following:

- How do we break this down into pieces? A good decomposition satisfies the principle of loose coupling between components (or pieces), facilitated by clean interfaces, simplifying the problem by dividing it into reasonably independent pieces that can be tackled separately.

- Do we have all the necessary pieces? The structure must support the functionality or services required of the system. Thus, the dynamic behaviour of the system must be taken into account when designing the architecture. We must also have the necessary infrastructure to support these services.
- Do the pieces fit together? This is a matter of interface and relationships between the pieces. But good fit - that is fit that maintains system integrity - also has to do with whether the system, when composed of the pieces, has the right properties.

This document therefore aims to give some initial view of the architecture for the p-medicine platform in order to address the complexity in building it and some of the issues described above. We try to follow a standards based approach further guided by established best practices. In the next paragraphs of this section we provide a review of some of the most important approaches in documenting a system architecture.

The Architecture Definition process

In recent years a realization has grown of the importance of software architecture. According to Bass et al [1] the software architecture of a system is “the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them”. The IEEE recommendation [7] defines an architecture as the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution. Software architectures are important because they represent the single abstraction for understanding the structure of a system and form the basis for a shared understanding of a system and all its stakeholders (product teams, hardware and marketing engineers, senior management, and external partners).

But how does a system architect proceed in order to design the architecture? A proposed architecture definition process is shown in the figure below:

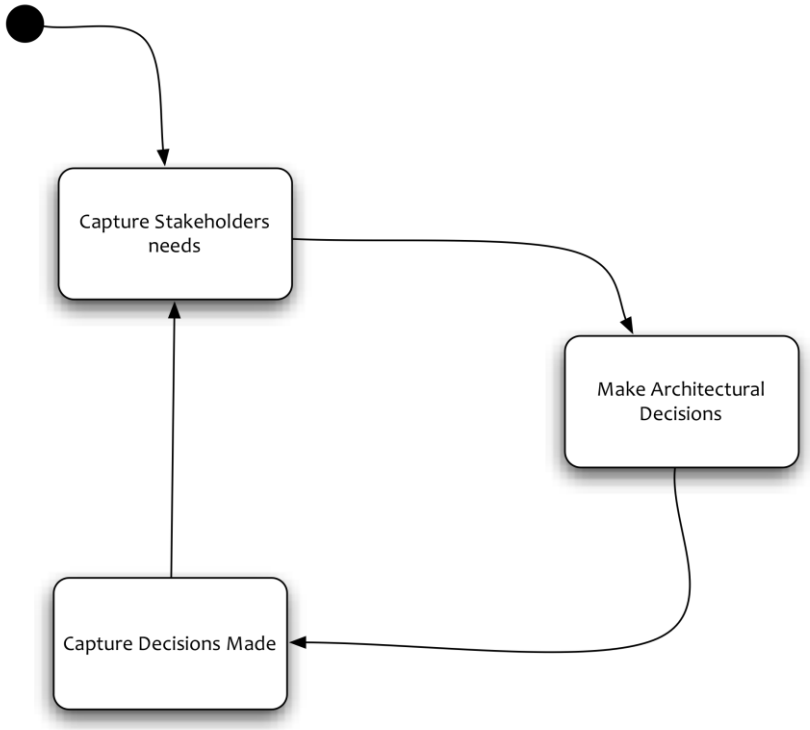


Figure 1 The architecture definition process

According to this process, there are the following steps:

- *Capturing stakeholder needs*, that is, understanding what is important to stakeholders (possibly helping them reconcile conflicts such as functionality versus cost) and recording and agreeing on these needs
- Making a series of *architectural design decisions* that result in a solution that meets these needs, assessing it against the stakeholder needs, and refining this solution until it is adequate
- *Capturing the architectural design decisions* made, in an Architectural Description

These activities form the core of the architecture definition process and are normally performed iteratively.

Review of Architectural Approaches

In order to comprehend a complex computer system, we have to understand what each of its important parts actually do, how they work together, and how they interact with the world around them – in other words, its architecture. Over the last 30 or more years a number of approaches have been proposed to describe and document software architectures. In this section we briefly describe some of the most well known ones.

2.1.1 Software Engineering Approaches

The waterfall process model is a linear series of steps that lead to delivery of the system (Figure 2). Common steps include requirements, design, implementation, testing and verification, and maintenance. Teams try to finish the current step before proceeding to the next. Going back to the previous step is allowed in order to fix mistakes, but otherwise discouraged. While waterfall processes are commonly seen in practice, few experts recommend them.

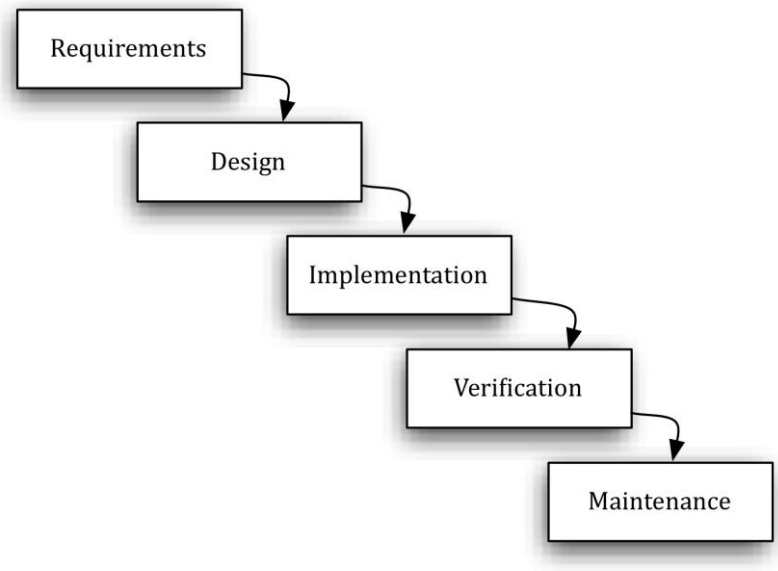


Figure 2 The waterfall model

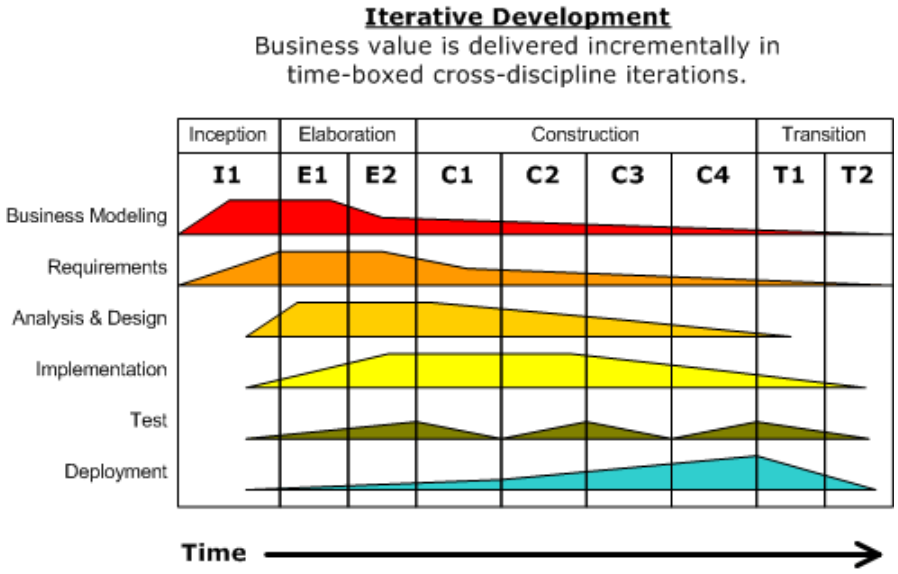


Figure 3 The iterative process of RUP

In contrast, the *spiral process model* of software development instructs engineers to build the system incrementally, starting from the highest risk items. Each turn of the spiral takes the team through all steps of software development, such as requirements, design, implementation, and testing. The spiral model is the basis of most modern processes, include agile processes and the Rational Unified Process (Figure 3).

2.1.2 The IEEE 1471 standard

The IEEE 1471 standard “Recommended practice for Architecture Description of Software-Intensive Systems” (http://www.iso-architecture.org/ieee-1471) addresses the activities of the creation, analysis, and sustainment of architectures of software-intensive systems, and the recording of such architectures in terms of architectural descriptions. A conceptual framework for an architectural description is established and the content of an architectural description is defined. Annexes provide the rationale for key concepts and terminology, the relationships to other standards and examples of usage. This recommended practice has been also adopted since 2007 as an ISO standard, ISO/IEC 42010:2007. Figure 4 illustrates the conceptual model of the architectural description, as defined in IEEE 1471.

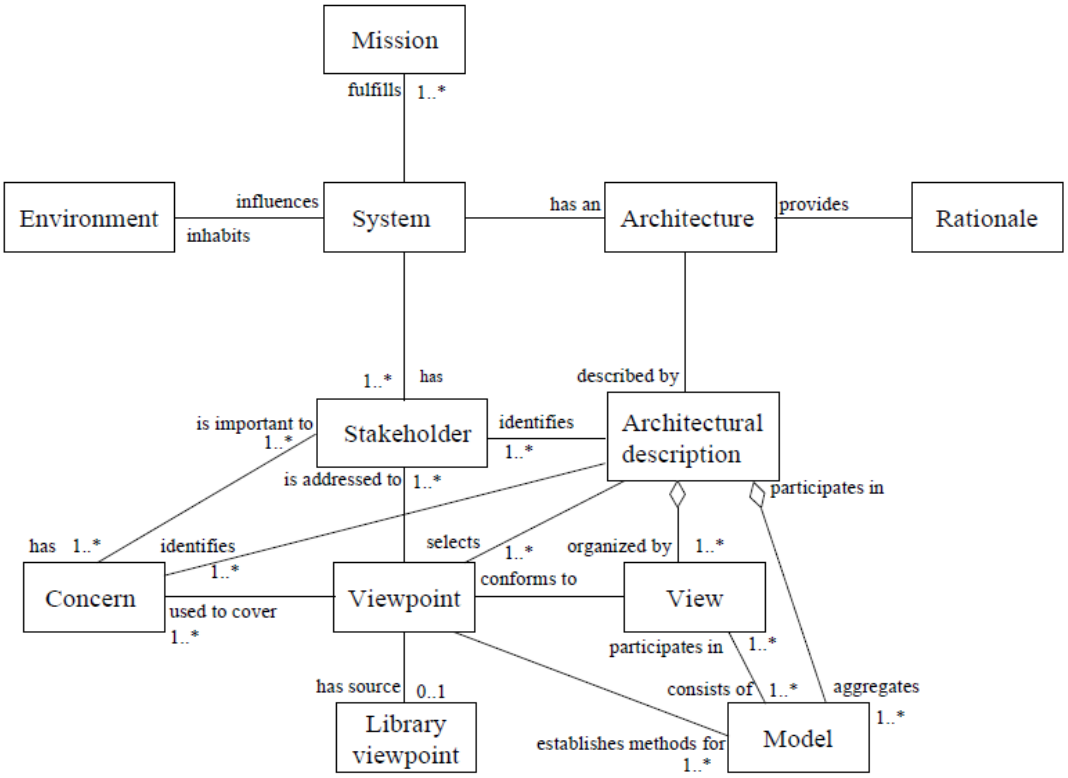


Figure 4 Conceptual model of architectural description from IEEE1471

According this conceptual model, a system has an architecture and this can be described in an architectural description. Note the distinction between the architecture of a system, which is conceptual, from the description of this architecture, which is concrete. Architectural description (AD) is defined as “a collection of products to document an architecture”. The AD can be divided into one or several views. Each view covers one or more stakeholder concerns. View is defined as “a representation of a whole system from the perspective of a related set of concerns”. A view is created according to rules and conventions defined in a viewpoint. Viewpoint is defined as “a specification of the conventions for constructing and using a view. A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis”. An AD selects one or more viewpoints for use and this choice depends on the concerns of the stakeholders that need to be addressed by the architectural description. A view may consist of one or more models and a model may participate in one or more views. Each such model is defined according to the methods established in the corresponding viewpoint definition. The AD aggregates the models, organized into views.

IEEE 1471/ISO/IEC 42010:2007 defines a set of requirements for conforming architectural descriptions that can be summarized as:

- AD identification, version, and overview information
- Identification of the system stakeholders and their concerns
- Specification of each viewpoint that has been selected and the rationale for those selections
- One or more architectural views
- A record of all known inconsistencies among the AD’s required constituents
- A rationale for selection of the architecture

It is evident from the discussion above that this standard is largely based on the definition of the most important viewpoints and the corresponding views but it does not provide any concrete definition of those. For this reason a number of different architectural frameworks

supporting different views and viewpoints have been proposed, such as the 4+1 views model, the Reference Model of Open Distributed Processing (RM-ODP[3]), the Zachman framework[4], the Department of Defense Architecture Framework (DoDAF)[5], etc. In the following paragraphs we visit two of these frameworks that we have chosen to base our methodology on.

2.1.3 4+1 Views Model

The "4+1" views model was originally developed in 1987 by Phillipe Kruchten of Rational Software [8]. According to this model each view represents a different set of important and related concepts that can be understood separately and that often have their own sets of expertise. This means that each view can be modeled (i.e., each view can be represented by a distinct set of models) and that these models can be assembled to create a complete system.

The logical view primarily supports behavioral requirements: the services the system should provide to its end users. Designers decompose the system into a set of key abstractions, taken mainly from the problem domain. These abstractions are objects or object classes that exploit the principles of abstraction, encapsulation, and inheritance. In addition to aiding functional analysis, decomposition identifies mechanisms and design elements that are common across the system.

The process view addresses concurrency and distribution, system integrity, and fault tolerance. The process view also specifies which thread of control executes each operation of each class identified in the logical view. The process view can be seen as a set of independently executing logical networks of communicating programs – processes – that are distributed across a set of hardware resources, which in turn are connected by a bus or a local area network or a wide area network.

The development view focuses on the organization of the software modules in the software development environment. The units of this view are small chunks of software – program libraries or subsystems – that can be developed by one or more developers. The development view supports the allocation of requirements and work to teams and supports cost evaluation, planning, monitoring of project progress, and reasoning about software reuse, portability, and security.

The physical view takes into account the system's requirements, such as system availability; reliability; performance; and scalability. This view maps the various elements identified in the logical, process, and development views – networks, processes, tasks, and objects – onto the processing nodes.

The graphical depiction of an architectural view is called an architectural blueprint. For the various views described above, the blueprints are composed of the UML diagrams:

- Logical View: Class diagrams, sequence diagrams and collaboration diagrams
- Process View: Class diagrams and collaboration diagrams encompassing processes
- Development View: Component diagrams
- Physical View: Deployment diagrams
- Use Case View: Use case diagrams

[3] ITU-T Rec. X.901-X.904 / ISO/IEC 10746, http://www.rm-odp.net/

[4] http://www.zachman.com/about-the-zachman-framework

[5] http://dodcio.defense.gov/dodaf20.aspx

2.1.4 Rozanski and Woods Viewpoint Set

Rozanski and Woods [6] prescribed a useful set of six viewpoints (in the ISO 42010 sense) to be used in documenting software architectures. They have essentially extended the 4+1 model by providing the Information viewpoint to deal with data related concerns, like structure, ownership, distribution, etc. and the Operational viewpoint in order to describe how the system is installed, monitored etc. Their six viewpoints are the following:

- The **functional view** documents the system's functional elements, their responsibilities, interfaces, and primary interactions. A functional view is the cornerstone of most architecture documents and is often the first part of the documentation that stakeholders try to read. It drives the shape of other system structures such as the information structure, concurrency structure, deployment structure, and so on. It also has a significant impact on the system's quality properties, such as its ability to change, its ability to be secured, and its runtime performance.
- The **information view** documents the way that the architecture stores, manipulates, manages, and distributes information. The ultimate purpose of virtually any computer system is to manipulate information in some form, and this viewpoint develops a complete but broad view of static data structure and information flow. The objective of this analysis is to answer the important questions around content, structure, ownership, latency, references, and data migration.
- The **concurrency view** describes the concurrency structure of the system and maps functional elements to concurrency units to clearly identify the parts of the system that can execute concurrently and how this is coordinated and controlled. This entails the creation of models that show the process and thread structures that the system will use and the interprocess communication mechanisms used to coordinate their operation.
- The **development view** describes the architecture that supports the software development process. Development views communicate the aspects of the architecture of interest to those stakeholders involved in building, testing, maintaining, and enhancing the system.
- The **deployment view** describes the environment into which the system will be deployed, including capturing the dependencies the system has on its runtime environment. This view captures the hardware environment that the system needs, the technical environment requirements for each element, and the mapping of the software elements to the runtime environment that will execute them.
- The **operational view** describes how the system will be operated, administered, and supported when it is running in its production environment. For all but the simplest systems, installing, managing, and operating the system is a significant task that must be considered and planned at design time. The aim of the operational view is to identify system-wide strategies for addressing the operational concerns of the system's stakeholders and to identify solutions that address these.

The selected approach

The p-medicine project has contractually committed to the use of standards and standard based methodologies. We have therefore chosen the architecture definition process and outcome to be in conformance to the IEEE 1471/ISO/IEC 42010:2007 standard. But as explained above this standard provides a template or an "ontology" for the description of a system's architecture that needs to be instantiated by the selection of a specific views/viewpoints set. Consequently the approaches that are centered on the notion of stakeholders, views, and viewpoints are in conformance to the ISO/IEEE 42010. Especially the Rozanski and Woods selection of the viewpoints but also their introduction of the perspectives (i.e. non functional, quality attributes) in the discussion seems to be the most

appropriate for describing the p-medicine architecture. In the next section of this document we therefore proceed to define our architecture following (not very closely, sometimes) their approach.

In any case the underlying principle of our methodology is to define “just enough architecture” [10], which means to continue until the basic requirements are met and the identified risks are addressed.

3 The p-Medicine Architecture

Introduction

As described in the previous section we have decided to follow an approach that conforms to the ISO/IEEE 42010 "*Systems and software engineering - Architecture description"* standard. In particular we have chosen to base the p-medicine architecture definition process on the set of viewpoints proposed by Rozanki and Woods [6]. Nevertheless this is definitely work in progress. The architectural description document is a live document in the sense that it evolves as the development of the actual system proceeds, as new requirements emerge, or previous decisions are reconsidered. It is therefore natural that some views are not fully described or have been totally eliminated because currently there's not input to drive them.

Stakeholders and Requirements

3.1 Stakeholders

A stakeholder is anyone who has an interest in or concerns about the system that we actually building. According to the description of work and the initial set of scenarios described in Deliverable 2.2 the p-medicine stakeholders are:

- Domain Users. These can be further classified in bioinformaticians, clinicians, users of clinical trials management systems, etc.
- Patients. In p-medicine where personalized provision of health and patient empowerment services are to be delivered, patients represent an additional type of stakeholders.
- Software Engineers/Developers. The people who actually build the system.
- Maintainers. The people that evolve and fix the system
- Administrators. The people who administer the system and keep it running.

In this document and at the current version thereof we mostly focus on the domain users and the patients, and secondly on the developers stakeholders. Focusing on the domain users/patients means that we elaborate on their concerns, which mostly have to do with the functionality, and some of its quality attributes such as security and usability. On the other hand the developers' concerns relate to the development process, its phases (e.g. design, code, test), and various "satellite" issues like the choice of the programming environment, the development tools, etc.

3.2 Overview of requirements

From the goals of the project as described in its technical annex we see that the two most important requirements for the platform to be developed are the management of the multilevel and heterogeneous patient data and the provision of tools for the publication, annotation, protection, analysis, etc. of these data. Therefore, in connection with the scientific/technical dimensions of the work, p-medicine will develop a data warehouse and a workbench with a tools repository. Heterogeneous pseudonymized data from different origins will be stored in the data warehouse for further use by the scientific community. Clinical data will be exploited coming from hospital information systems and clinical trials. The legal framework of the project, which is based on the results of ACGT (Advancing Clinico-genomic trials[6]), will be further developed and will guarantee data privacy and security. Most important for *p-medicine* are validated tools and services that provide interfaces to allow interoperability

[6] http://eu-acgt.org/

with biobanks, genetic databases, and medical imaging systems and data warehouses. These tools have to meet requirements to be used in large, international multicentre clinical GCP conform trials and need to be able to be integrated into existing systems used by ECRIN and other communities. This includes aspects like data security by adopting the legal and ethical framework based on international requirements and approved concepts for anonymization and pseudonymization including validation. Previous R&D work done in European funded projects like ACGT, ContraCancrum and ECRIN (European Clinical Research Infrastructures Network) fit perfectly into this approach and will be heavily drawn on. The following figure (fig. 2.2) shows the main components and their interdependency of the *p-medicine* system architecture from a clinical perspective.

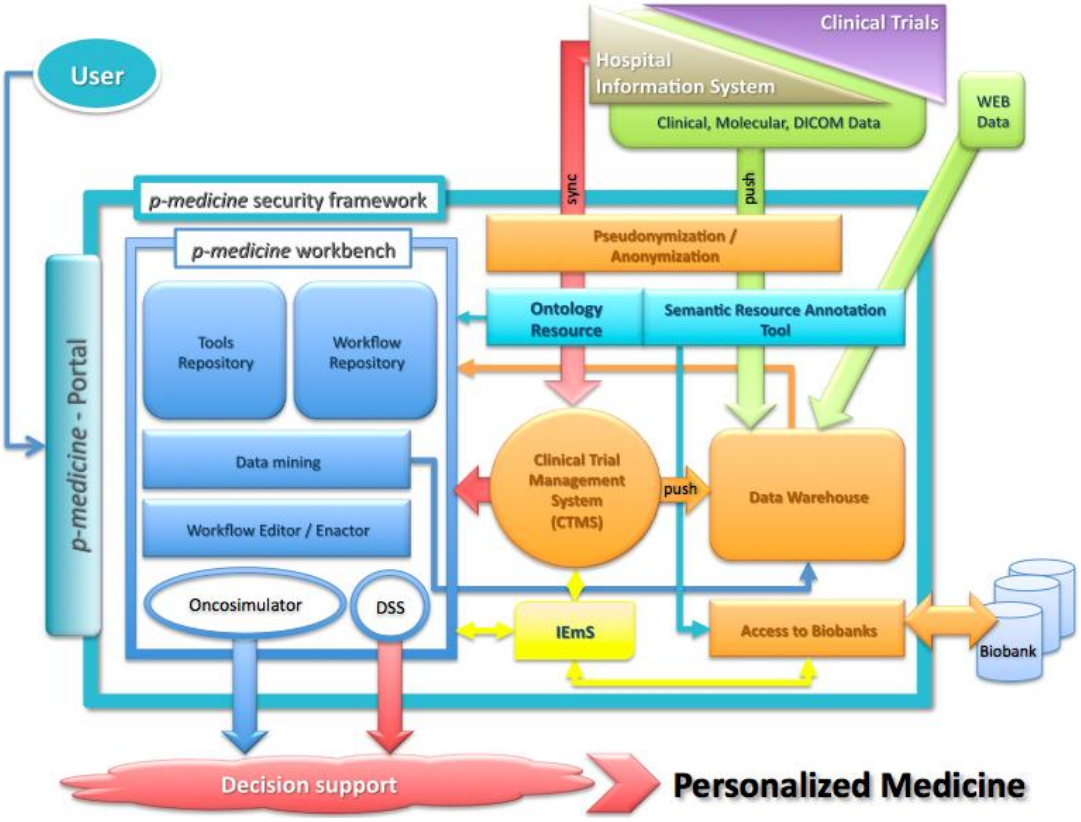


Figure 5 The architecture of p-medicine from a clinical perspective.

3.3 System Scenarios

The functional view of the p-medicine (i.e. “what the system does”) can be illustrated by following a layered approach of functional requirements alongside with the “cross-cutting” (vertical) personalized user scenarios, as shown in the next picture:

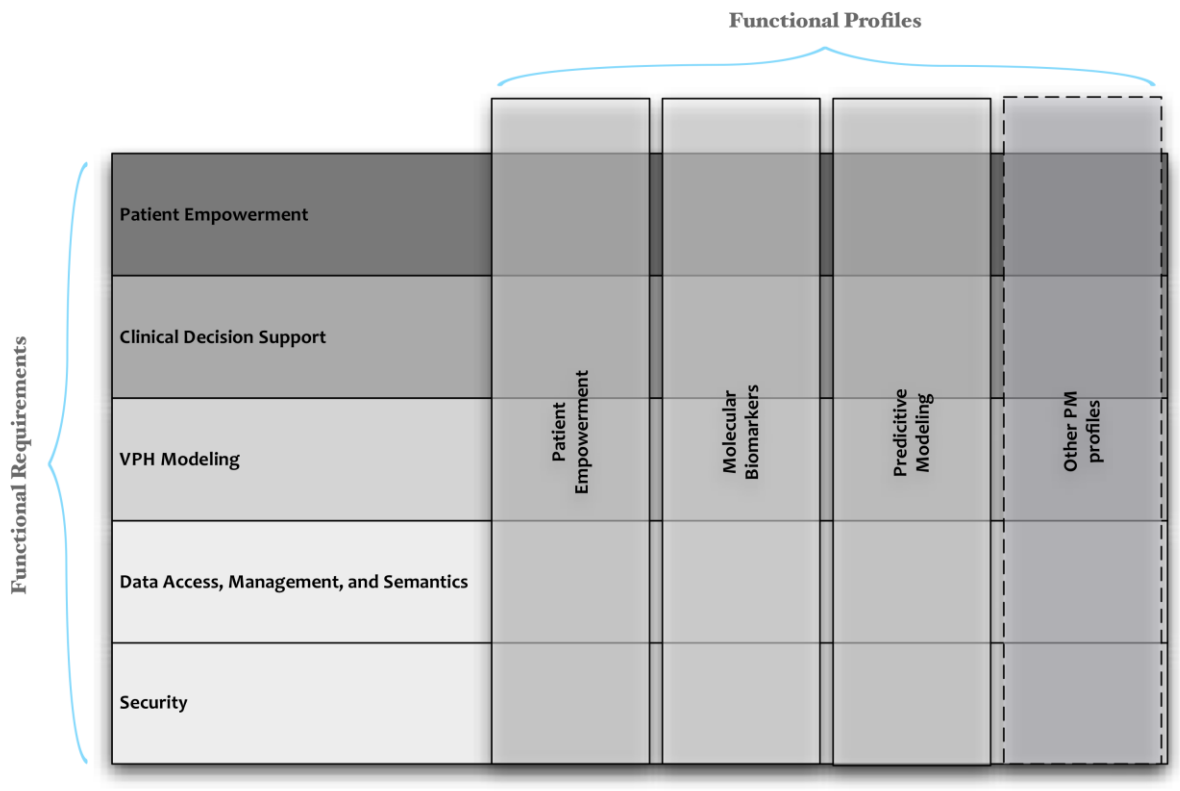


Figure 6 The Functional (meta)view

In this figure we represent major functional requirements as layers that stack up. We have identified the following general categories:

- Security. This dimension deals with the security, privacy, and access control of the sensitive patient data that is generally all pervasive, and not a “functional profile” (from the users point of view) per se.
- Data Access and Management. The p-medicine platform is primarily a system for storing, processing, and maintaining data. This layer therefore deals with the handling of data from their initial import to all the stages of their “life cycle” by maintaining linkage and provenance. This layer also incorporates semantic harmonization tools, which are responsible for semantic annotation, translation, ontology maintenance, etc
- VPH Modeling. This is where tools and components supporting the modeling and simulation of tumor growth and response to drugs and other therapy plans are located.
- Clinical Decision Support tools
- Patient Empowerment tools

On the other hand selected user scenarios as represented as vertical blocks that span most of the functional requirements due to their “cross cutting” nature. We call them “Functional Profiles”. A “functional profile" is a selected set of functions that are applicable for a particular purpose, user, care setting, domain, etc. Functional profiles help to manage the master list of functions. We can identify the following general application areas for these profiles:

- Knowledge Discovery. This incorporates scenarios like new biomarkers discovery and experimentation in order to produce new knowledge that of course needs to subsequently be validated.
- Patient Empowerment, where the patient actively participates and interacts with the system in order to become aware of new possibilities for improving his health or helping the active research, like searching for clinical trials to enroll in.

- Predictive Modeling

In the following paragraphs we are allocating the architectural elements (“components”), which we have identified from the use case scenarios, in these categories, both horizontally and vertically. This may seem to be preposterous since the use cases are described and further analyzed in the Functional View in paragraph 3.3.6.2 but it makes easy to see the full picture and also to further describe an indicative “vertical” scenario (functional profile) in paragraph **Fehler! Verweisquelle konnte nicht gefunden werden.**.

3.3.1 Architectural Elements

Name	Acronym	Description	Work Package
Ontology Based Trial Management Application	ObTiMA	ObTiMA provides a friendly user interface to manage clinical trials. Users can create and complete eCRFs and manage clinical trials. Moreover ObTiMA can be used as an interface to other tools.	WP8
p-Medicine Data Warehouse	DWH	It is the central repository of p-medicine’s data. It allows the storage of data in a secure, distributed, highly accessible environment.	WP7
Portal	Portal	The p-medicine Portal provides a user-friendly interface both to patients and physicians in order to access information stored in the Data Warehouse. It visualizes patient data and manages consent and re-consent.	WP14
Translation Tool	Translation Tool	This component translates external data in HDOT compliant format. In order to do that the data should be accompanied by the	WP4

		proper meta-data (its schema and the corresponding mapping between the schema and the HDOT).	
Annotation Tool	Annotation Tool	This tool provides a user-friendly interface to a Database Manager in order to annotate an external Database using the HDOT ontology.	WP4
External Databases	ExtDB	External Databases will offer data that are not already stored in the DW.	To be defined from WP13
Identity Provider	IdP	The Identity Provider (IdP) is a service provider within a federation responsible for authentication. It provides identity assertions to other service providers.	WP3
Clinical Decision Support	CDS		
Identity Assertion Consumer		An Identity Consumer is an entity, that is part of a service provider, which consumes the assertions provided by the Identity Provider. It will verify the received assertion and pass it to the service provider's application layer.	WP3
Policy Enforcement Point	PEP	A Policy Enforcement Point (PEP) is a logical entity which requests and enforces authorization decisions.	WP3
Policy Decision Point	PDP	A Policy Decision Point (PDP) is an	WP3

		entity that makes authorization decisions. A PDP accepts authorization requests and will make a decision based on policies fetched from a Policy Administration Point (PAP).	
Policy Administration Point	PAP	A Policy Administration Point is an endpoint, which manages policies. It will provide a PDP with all policies required to produce an authorization decision.	WP3
Policy Information Point	PIP	A Policy Information Point (PIP) is an endpoint which provides missing information to a PDP i.e. attribute information. For example if a policy requires information on a specific attribute which has not been provided with the authorization request, a PDP might request a PIP for information on that attribute.	WP3
Trusted Third Party	TTP	The TTP cryptographically transforms a pseudonym so that it cannot be re-identified without going back through the TTP. The TTP therefore controls whether it is allowed to re-identify a pseudonym.	WP8
Custodix Anonymization Tool	CAT	This tool anonymizes data (patient data, images, etc.) and stores the identity of each specific datum	WP3

		in order to be able to relate it again to its source	
Custodix Anonymization Tool Services	CATS	The anonymization services de-identify input files based on a predefined set of transformation rules.	WP3
Personal Information Management System	PIMS	Feeding patient identifying information from different sources, allows PIMS to match and link patient records to real-life persons. Afterwards PIMS can issue domain specific pseudonyms to any requestor.	WP8

3.3.2 Application Domain Areas

P-medicine is clinically driven project that focuses in three different diseases with currently running clinical trials. The three selected diseases are Wilms Tumour, Breast Cancer and Acute Lymphoblastic Leukaemia (ALL). Trials for these diseases are selected by p-medicine in a way that they can address different aspects of the project. Data coming from these trials will be stored in the data warehouse in a secure and anonymized way according to the legal and ethical framework of p-medicine.

3.3.2.1 Breast Cancer

Treatment for breast cancer has already been improved by discovery of reliable surrogate markers of response. The disease is now split into many subsets based on hormone receptor data, genomic signatures and imaging characteristics and the evidence base for validated therapeutic choices is more advanced than any other area of oncology. Electronic patient records interfaced with bio-banks, genetic databases, and medical imaging systems will be available for new methodologies of data analysis. The primary aim of our studies is to maximize efficacy of therapy while minimizing side effects.

P-medicine will support a breast cancer scenario based on immunohistochemistry (IHC), gene expression and clinical data. The clinical data will be provided by ObTiMA. The gene array data will be provided as CEL files and will be stored at the data warehouse. The breast cancer scenario also supports data from external databases and specifically from the online pathway database KEGG (http://www.genome.jp/kegg/pathway.html). Access to KEGG data will be supported by the external DB access tool. Integration of KEGG data with gene expression and clinical data will be supported by the annotation tool. The analysis of the data will be based on the available tools from the p-medicine Data Mining Pattern Environment. Specific cohorts of patients with breast cancer will be produced as a result.

Detailed description of the pathways and breast cancer scenario can be found at the Deliverable 11.1. The p-medicine use case template for the specific scenario can be found at the appendix 2 (this use case is not available at D2.2 [4] of p-medicine).

3.3.2.2 Leukaemia

New approaches that go beyond the statistical approaches currently applied in data mining may be helpful in gaining new perspectives on treatment strategies for clinical application at different levels in ALL especially in those patients with a dismal response to treatment. In particular, the increasing dimensionality and complexity of available clinical and genetic/genomic data demands more comprehensive solutions in order to resolve the bottleneck of data interpretation.

P-Medicine will support two main scenarios for ALL treatment with three different group data. The scenarios are oriented to:

1. Minimal residual disease (MRD): MRD is the name given, to small numbers of leukaemic cells that remain in the patient during treatment or after treatment when the patient is in remission (no symptoms or signs of disease). It is the major cause of relapse in cancer and leukaemia.
2. Disease recurrence: To find indicative patterns within basic, treatment, response, gene expression and genomic data that can discriminate the VHRL (very high risk leukaemia) and non VHRL patients.

The data that will be available is based on three different patient groups from trial ALL-BFM 2000 (data on basic characteristics at diagnosis, treatment, response and outcome, only, are available for more than 4000 patients):

Group 1: representative cohort of 664 patients

- Basic data: gender, age at diagnosis, white blood cell count at diagnosis, blood blast count, hemoglobin levels and platelet counts at diagnosis, FAB classification, complete immunophenotyping data, ploidy status, status for prognostic relevant chromosomal translocations (ETV6/RUNX1, BCR/ABL, MLL/AF4, E2A/PBX1), percentage of bone marrow blasts, extramedullary disease (CNS, testis, and others).
- Treatment data: risk group stratification, cumulative drug doses, information on HSCT and cranial irradiation, information on time frame for the application of treatment phases.
- Response data: prednisone response, blast percentages in the bone marrow on treatment days 15 and 33, MRD analyses on treatment days 33 and 78.
- Outcome data: relapse, treatment-related mortality, secondary malignancy.
- Gene expression data: low-density array of 95 genes previously associated with treatment response and/or outcome.

Group 2: case-control of 50 VHRL and 50 non-VHRL patients:

- All of the data from Goup 1.
- Genomic data: leukemic genome-wide gene expression profiles (cDNA, Stanford Functional Genomics Facility), Affymetrix 6.0 SNP arrays (leukemic and germline), leukemic epigenetic profiles (custom-made array, University of Münster, Germany).

Group 3: cohort of 475 ETV6/RUNX1-positive patients

- All of the above except gene expression data (group 1) and genomic data (group 2) plus:
- Germline genetic data: Affymetrix 5.0 SNP arrays.

All data except genomic data (group 2) and germline genetic data (group 3) will be stored in the data warehouse after pseudonymisation. Genomic and genetic data will be available from external data sources using database manager in order to annotate the external sources using the HDOT ontology.

All the scenarios follow the general sequence diagram and can be supported by the p-medicine components described at the general components diagram (figure Figure 26: General component diagram for all the clinical use cases).

Use cases of ALL scenarios can be found at the appendix 2 (use cases of ALL are not available at D2.2 [4] of p-medicine).

3.3.2.3 Wilms Tumor

The Wilms tumour trial will be used to employ the newly developed atools of p-medicine. According to PSN_1 scenario (pathway scenario for nephroblastoma Deliverable 2.2[]) KEGG pathway, gene expression and clinical data will be combined to enrich the clinical decision support of the physician.

The Wilms tumour use case describes how clinical data from a clinical trial can be statistically analysed together with molecular data within Obtima and biology external data (KEGG pathways database). The scenario is described in detail in D2.2 [4] and presents the interaction between a physician and the Obtima in order to combine clinical data stored in Obtima, molecular data stored in DW and biology online database to perform data analysis.

Gene expression data from nephroblastoma serve as the source of disrupted metabolic pathways. These data needs to be normalized and then correlated to pathway data coming from the KEEG pathway database (http://www.genome.jp/kegg/pathway.html). These tools will analyse the tumour of disrupted metabolic pathways. By correlation to clinical data of patients, individual pathway disruptions or main disruptions for a cohort of patients with nephroblastoma will be produced as a result. In individual patients it will be possible to find disrupted pathways in the tumour for selecting specific drugs for treatment, like ATRA (all-trans retinoic acid) if the retinoid pathway is disrupted.

3.3.3 System Quality Scenarios

System quality scenarios model how the system should react to a change in its environment, such as an increase in workload or a security breach.

So far we have not developed any such scenario. However, for the time being the quality assurance scenarios are being developed in WP15 and will be extensively reported at month 12. This WP is in the process of identifying objectives that need to be specifically tested in each case, define the proper evaluation criteria and devise monitoring procedures that will be executed.

Architectural Drivers

3.3.4 Goals

P-medicine brings together internationally recognized leaders in their respective fields with the aim to *create an innovative computational, service-oriented infrastructure that will facilitate this gradual translation from current medical practices to personalized medicine*. In achieving this objective *p-medicine* has formulated a coherent, integrated work plan for the design, development, integration and validation of all technologically challenging areas of work.

Our emphasis is:

- On drafting an open and modular architectural framework for the tools and services to be developed, so that adoption of the p-medicine services will not be an all-or-nothing decision;
- On efficient sharing and handling of the enormous personalized data sets - including policies, security, modeling, cloud storage, etc.;

- On enabling demanding VPH simulations, for which standardization and semantic data integration and interoperability is a major issue addressed;
- On building and standardizing tools and models for VPH research, such as the VPH Toolkit[7], by defining a formalism to make the knowledge that is implicitly encoded in these tools explicit and thus improve the re-use of tools and solutions;
- On providing tools for large-scale, privacy-preserving data mining, and literature mining, a key factor in VPH research.

On the policy front, we focus in making sure that policies with respect to privacy, non-discrimination, and access are aligned to maximize both the protections and the benefits to patients.

3.3.5 Constraints

One of the most important constraints that affect the architecture is the adherence to the legal guideline and to the regulation related for the sharing of patient data. In particular the use of pseudonymization and de-identification of patient data when used for clinical research outside the confines of a particular health provision activity, e.g. in a hospital or medical centre, is principal along side with the reverse process, i.e. the re-identification, when specific clinical findings require contacting the patient. Data storage even for the anonymized data should also provide the necessary security mechanisms to keep them out of sight for the unauthorized personnel.

3.3.6 Principles

The definition of the architecture is also guided by principles. A principle is a fundamental statement of belief, approach, or intent that may refer to current circumstances or a desired future state.

There is a strong consensus among the partners of the project on the promotion of open source and open standards. This means that open source will be both adopted as a development process and leveraged by the reuse of existing free and open source tools. The rationale for this decision is based on the research character of the project and also on the success of the open source initiative in similar endeavors. The project is also clinically driven. It targets the fulfillment of urgent needs of the cancer research community and aims to strengthen the integration of the European Research Area. To this end the use of open standards and interoperable data formats and ontologies is also of utmost importance.

Architectural Views

In this section we describe the architecture of the p-medicine platform from the various viewpoints of Rozanski and Woods.

3.3.6.1 Context View

The system context provides an overview of the system and the actors and other systems that it interacts with. A context diagram for the p-medicine, when considered a single, unified system, can be seen in Figure 7. The main feature of this type of diagram is that it shows more clearly the connectors i.e. the channels of communication with the external systems.

There is a “fuzzy” boundary that encloses the p-medicine platform based on the security constraints and the semantics. The security framework in use clearly puts a “hard” constraint on what can be considered part of the p-medicine, as we also described above. But also the dependence on the common terminology and semantic infrastructure require a certain integration barrier for the inclusion of data services in the p-medicine system.

[7] http://www.vph-noe.eu/wp3

Nevertheless, it is the goals and the requirements of the system that more clearly delineate the borders of the p-medicine architecture. According to these requirements, the following are external entities for the system under development:

- Hospital information systems, which are the primary "feeders" for clinical data
- Clinical Trials Management systems, which provide additional data, patient information, and trial specific information.
- Public –omics databases and domain specific knowledge bases like KEGG and Gene Ontology
- Biobanks
- Public registries for tools

In the same figure we have also included the major stakeholders of the system. These are essentially the users of the p-medicine platform:

- Patients
- Bioinformaticians
- Clinicians
- Clinical trials users

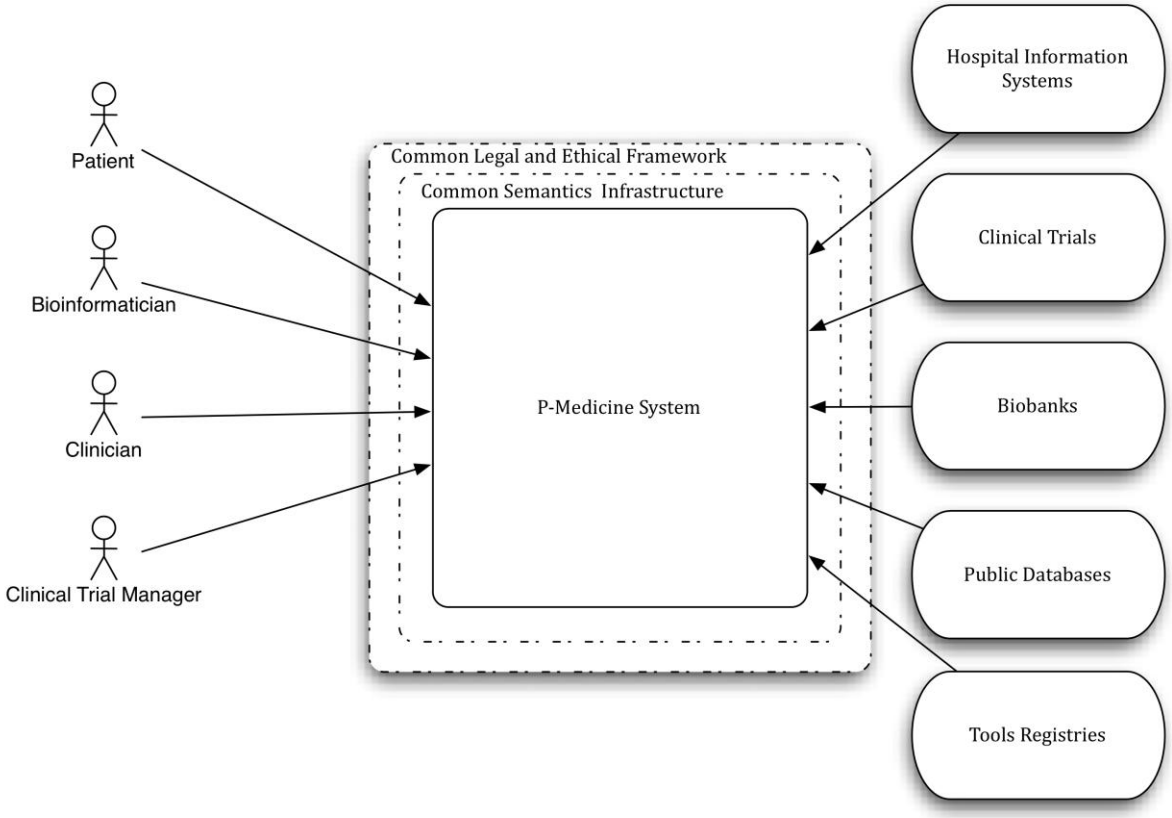


Figure 7 The p-medicine platform as a unified system and its interactions with external entities

3.3.6.2 Functional View

A logical view of the architecture based on the required functionality already defined in the project's description of work can be seen in Figure 8. At this abstraction level we don't explicitly depict the components' functionality, the details of their interactions, and their dynamic behavior (e.g. when these interactions take place, etc.). In the following paragraphs we are going to describe some of the identified scenarios and the responsibilities of the components, their interactions, etc. will become clearer.

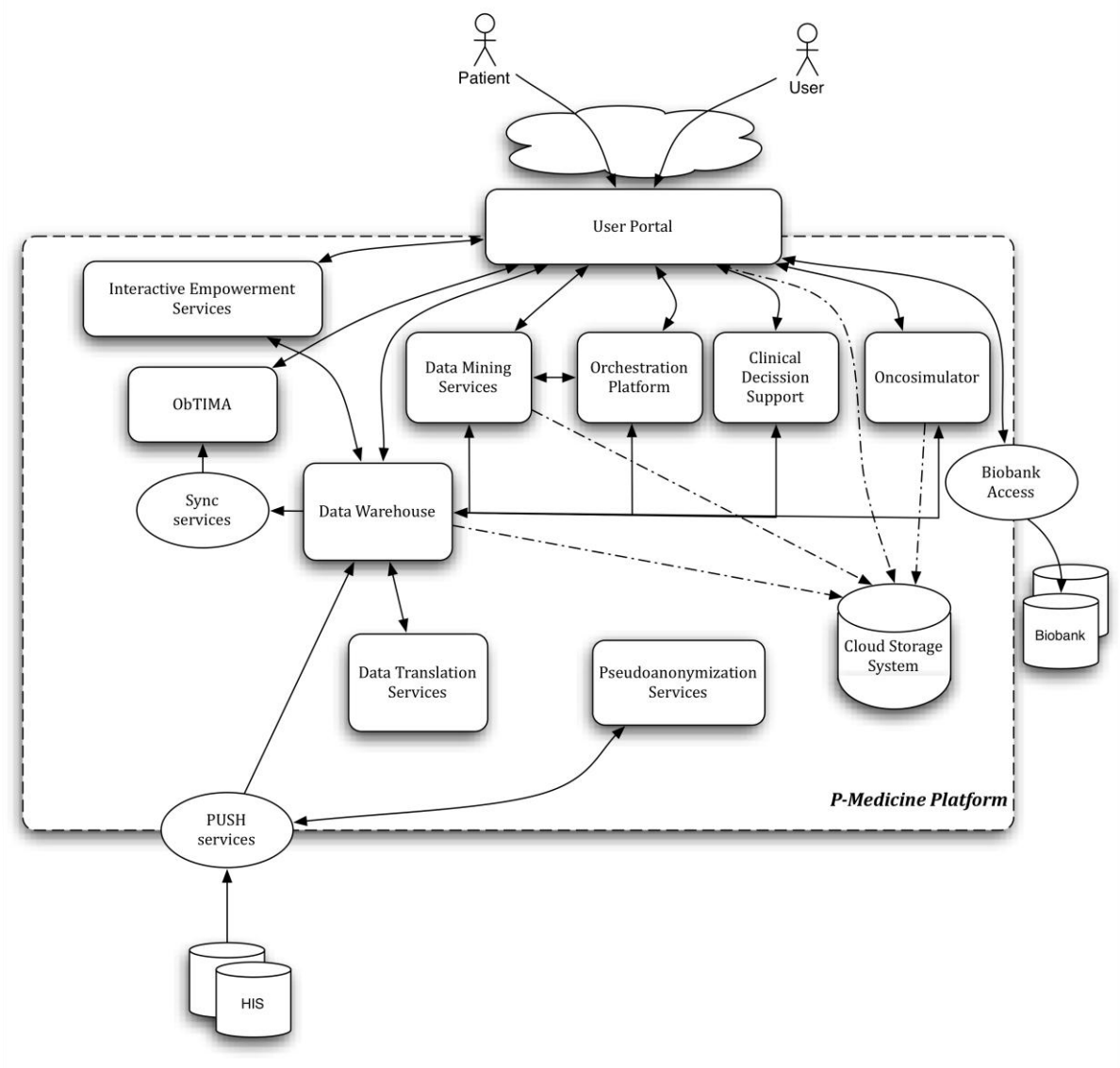


Figure 8 The main components of the system and their interactions

3.3.6.2.1 Functional scenarios

Functional requirements capture the intended behavior of the system, i.e. *what* the system does. This behavior may be expressed as services, tasks or functions the system is required to perform.

It is natural that these functional requirements evolve during the development of the systems and the delivery of new releases. The functional requirements of previous releases need to be explicitly taken into account. Later releases are accommodated through architectural qualities such as extensibility, flexibility, etc. The latter are expressed as non-functional requirements or system qualities.

For the description of the functional requirements we have used *use* cases. Use cases have quickly become a widespread practice for capturing functional requirements. Each use case defines a goal-oriented set of interactions between external actors and the system under consideration. *Actors* are parties outside the system that interact with the system[8]. An actor may be a class of users, roles users can play, or other systems.

[8] OMG Unified Modelling Language (OMG UML), Superstructure, V2.1.2", http://www.omg.org/spec/UML/2.1.2/Superstructure/PDF/ Retrieved January 12, 2012

A use case is initiated by a user with a particular goal in mind, and completes successfully when that goal is satisfied. It describes the sequence of interactions between actors and the system necessary to deliver the service that satisfies the goal. It also includes possible variants of this sequence, e.g., alternative sequences that may also satisfy the goal, as well as sequences that may lead to failure to complete the service because of exceptional behavior, error handling, etc. The system is treated as a “black box”, and the interactions with system, including system responses, are as perceived from outside the system.

Thus, use cases capture *who* (actor) does *what* (interaction) with the system, for what *purpose* (goal), without dealing with system internals. A complete set of use cases specifies all the different ways to use the system, and thus defines all behavior required of the system, bounding the scope of the system.

On the other hand a *scenario* is an instance of a use case, and represents a single path through the use case. Thus, one may construct a scenario for the main flow through the use case, and other scenarios for each possible variation of flow through the use case (e.g., triggered by options, error conditions, security breaches, etc.). Scenarios may be depicted using sequence diagrams

Use cases are useful in capturing and communicating functional requirements, and as such they play a primary role in product definition. An architecturally relevant subset of the use cases for each of the products to be based on the architecture also plays a valuable role in architecting. They direct the architects to support the required functionality, and provide the starting points for collaboration diagrams (or sequence diagrams) that are helpful in component interface design and architecture validation.

In this section we have selected a number of use cases that were described in the Deliverable 2.2, in order to elicit the p-medicine functional requirements. The selection was made on the criterion that they should be “cross-cutting” and characteristic for the goals of the project.

3.3.6.2.2 Generic Use cases

3.3.6.2.2.1 Security

Security is a cross cutting functionality in p-medicine platform since many components process sensitive personal data and there are various security components that need to be implemented in order to offer a reliable and secure system. First of all, a mechanism is needed that allows the users to authenticate themselves by providing personal credentials, with which the users can confirm their identity on the different sites/services of the platform. Another important part of security is access control in order for a user to only access and manipulate resources of the p-medicine on which he is authorized. And of course there are other security components needed for the encrypted storage of data, pseudonymisation of patients, safe transmission of data providing confidentiality and integrity and others. Below we present some selected scenarios, which showcase the security functionality needed from p-medicine, as defined by the user requirements elicitation phase (deliverable D2.2).

3.3.6.2.2.1.1 Single Sign On

The end-user needs to authenticate herself/himself on different sites or services of the p-medicine platform. An architecture where a user needs to provide his credentials for each site/service separately is not sustainable and not user-friendly. A better architecture uses a central identity provider (IdP), as shown in the following diagram.

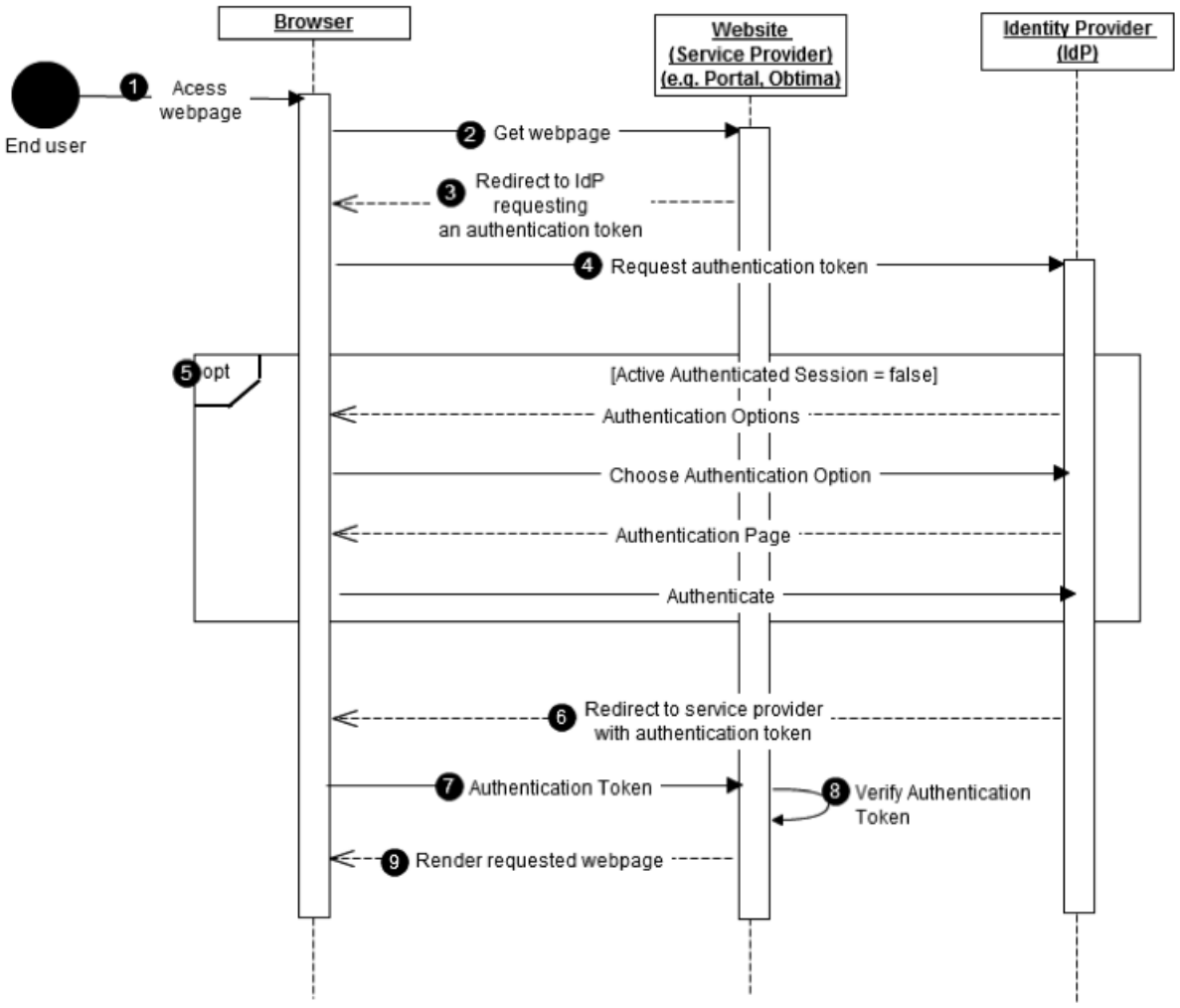


Figure 9 The sequence diagram for the "*Single Sign On*" use case

The interactions shown are as follows:

- **1 & 2**: An end-user browses to a web-page giving access to a p-medicine service (e.g. portal or Obtima).
- **3 & 4**: The web server detects that the end-user is not authenticated locally and redirects the end-user to the p-medicine identity provider.
- **5**: The identity provider (IdP) detects whether the end-user has an active Single Sign On (SSO) session. If no active session is detected, the end-user is prompted to select an authentication method. Initially only one authentication method will be provided (username/password). If username and password are valid, the end-user is authenticated and an SSO session is created on the IdP.
- **6 & 7**: The IdP redirects the end-user back to the original webpage the end-user wanted to access, passing through the end-user's authentication token.
- **8**: A local service on the web server verifies the authentication token received by the IdP and creates a local session if valid.
- **9**: The requested webpage is rendered.

3.3.6.2.2.1.2 Single Sign-out

A user that is authenticated on one or more sites/services using SSO, may want to logout from all this sites/services. This logout should be user-friendly, making it

possible to logout from all the sites/services in one simple action (single sign-out). The steps that are needed for single sign-out are explained in the following use case.

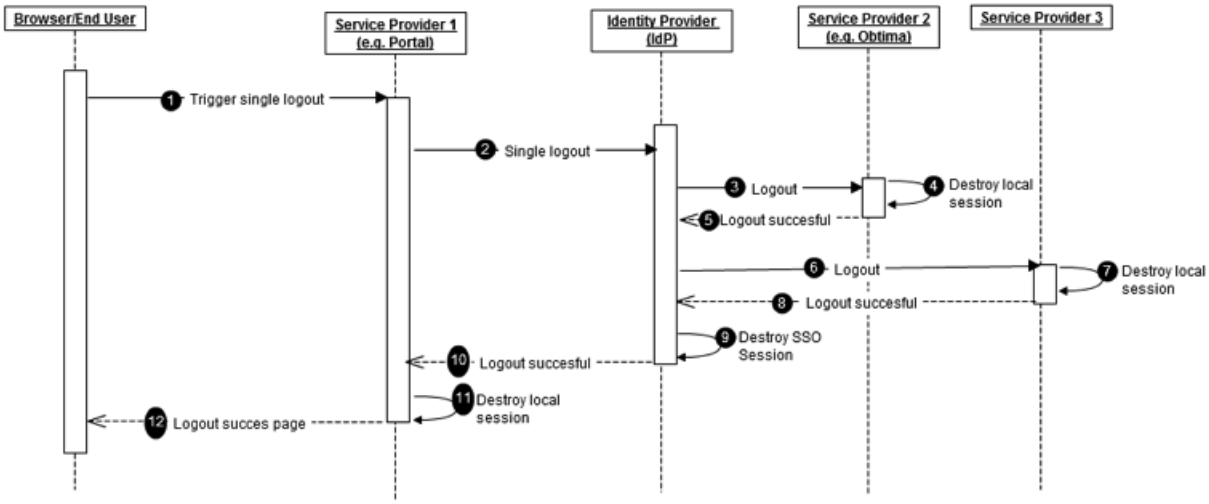


Figure 10 The sequence diagram for the "*Single Sign Out*" use case

The steps are as follows:

- **1:** The end-user selects the logout link on the local p-medicine web-site/service he is currently working on.
- **2:** The local service sends a single logout (SLO) request for the end-user to the Identity Provider (IdP).
- **2 - 8:** The IdP sends a logout request for the end-user to all connected p-medicine services (except the one that requested logout). Each of the contacted services attempts to destroy their local end-user session. Upon success they send back a logout response to the IdP, indicating the end-user session was successfully destroyed.
- **9:** The IdP destroys the SSO session of the end-user.
- **10:** The IdP returns a logout response to the initiating service provider indicating the success of the single logout request.
- **11 & 12:** The service provider destroys its local session and renders a logout success page.

3.3.6.2.2.1.3 Authorization

When a user wants to perform an action (e.g. read) on a shared resource (e.g. a data set) through a service (e.g. the data warehouse), the service needs to verify that the user is allowed to perform this action on the resource. On a complex and distributed system such as the p-medicine platform, a good design is to have a central configuration point where these authorization rules are maintained. We identify the following entities:

- A Policy Enforcement Point (PEP) is a software component which requests and enforces authorisation decisions.
- A Policy Administration Point (PAP) is an endpoint, which manages policies. It will provide a PDP with all policies required to produce an authorisation decision.
- A Policy Decision Point (PDP) is an entity that makes authorisation decisions. A PDP accepts authorisation requests and will make a decision based on policies fetched from a Policy Administration Point (PAP).

- A Policy Information Point (PIP) is an endpoint which provides missing information to a PDP i.e. attribute information. For example if a policy requires information on a specific attribute which has not been provided with the authorisation request, a PDP might request a PIP for information on that attribute.

The interactions between these components when a service (the Data Warehouse, in this example) is accessed in shown in the next sequence diagram:

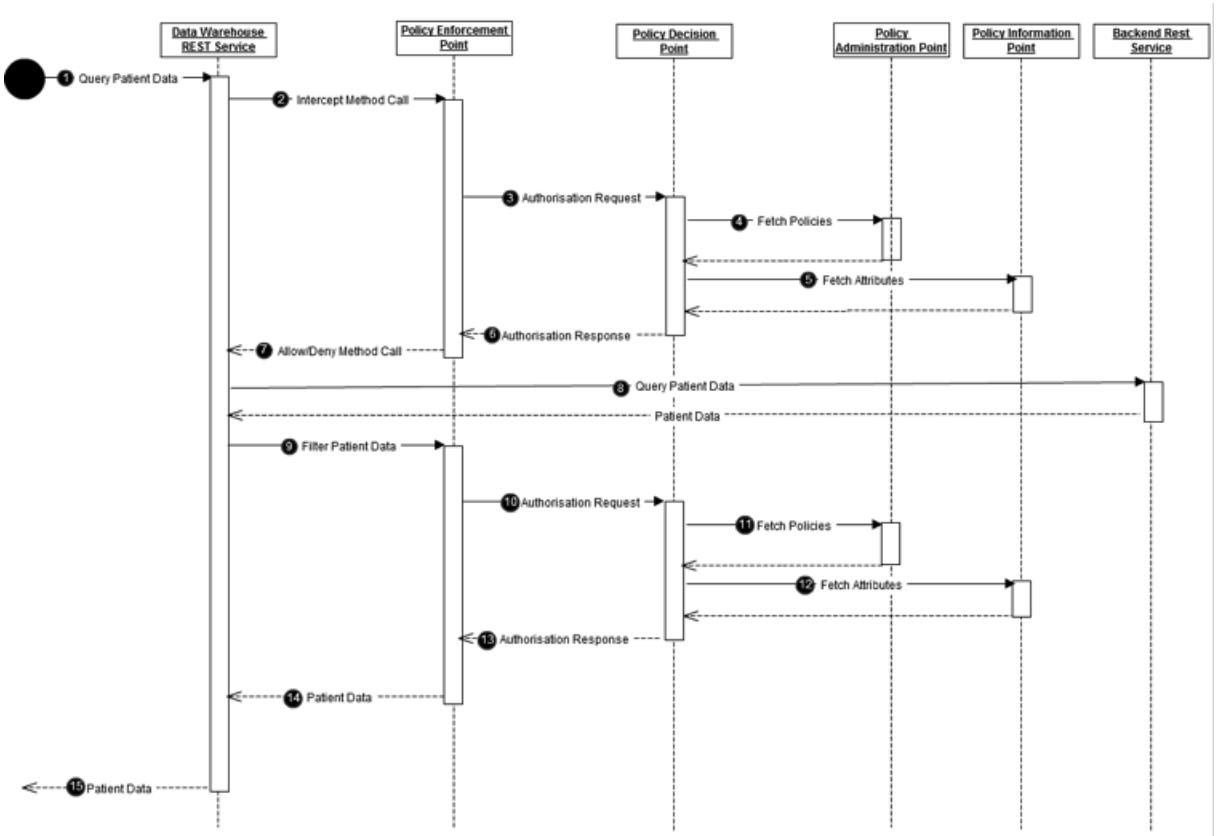


Figure 11 The sequence diagram for the "*Authorization*" use case

Some description of the steps shown in the diagram is in order:

- **1:** A front-end service (which in this use case acts as client), e.g. a portlet processing or rendering patient data, queries the data warehouse for patient data.
- **2:** The method call on the data warehouse is intercepted by the policy enforcement point (PEP).
- **3:** The PEP creates an authorisation request containing the action performed (query patient data) and the subject performing the action (this subject was passed with the REST query to the data warehouse through i.e. a forwarded or delegated token). The PEP then sends this authorisation request to the p-medicine policy decision point (PDP).
- **4:** The PDP will fetch all policies relevant to the received authorisation request from the policy administration point (PAP).
- **5:** If the information on the subject in the authorisation request is insufficient to take an authorisation decision, the PDP will query the policy information point (PIP) for more information on the subject.
- **6:** The PDP will take a decision and send it back to the data warehouse's PEP.

- **7:** The PEP will either deny or allow the user's action to be performed based on the authorisation decision received from the PDP.
- **8:** If the action is allowed the data warehouse sends queries the backend services that will return the requested patient data.
- **9:** The PEP intercepts the returned patient data to filter out all data the user does not have access to.
- **10:** For each data item, the PEP will create an authorisation request and send it to the PDP.
- **11 & 12:** Similar as in step 4 & 5 the PDP will fetch from the PAP all policies relevant to relevant to the fetched data. If the information on the subject in the authorisation request is insufficient to take an authorisation decision, the PDP will query the PIP for more information on the subject.
- **13:** The PDP will take a decision and send it back to the data warehouse's PEP.
- **14:** The PEP will remove all data items from the response for which a “deny” decision was returned by the PDP and return all remaining data items.
- **15:** All queried patient data to which the subject has access are returned to the client that issued the request.

3.3.6.2.2.1.4 Anonymization

Sharing heterogeneous data from multiple sources can be a threat to personal integrity, which shall be minimized. In p-medicine only pseudonymized or anonymized datasets will be used, which requires a certain software infrastructure to be in place so that no scientist working with the data will ever know the true identity of the study subjects. This infrastructure is mainly composed of the Custodix Anonymization Tool (CAT) and its service-oriented evolution CATS (Custodix Anonymization Tool Services) that are responsible for the transformation (i.e. pseudonymisation, encryption, etc.) of input files (plain text, CSV, XML, etc.) prior to entering into the p-medicine environment.

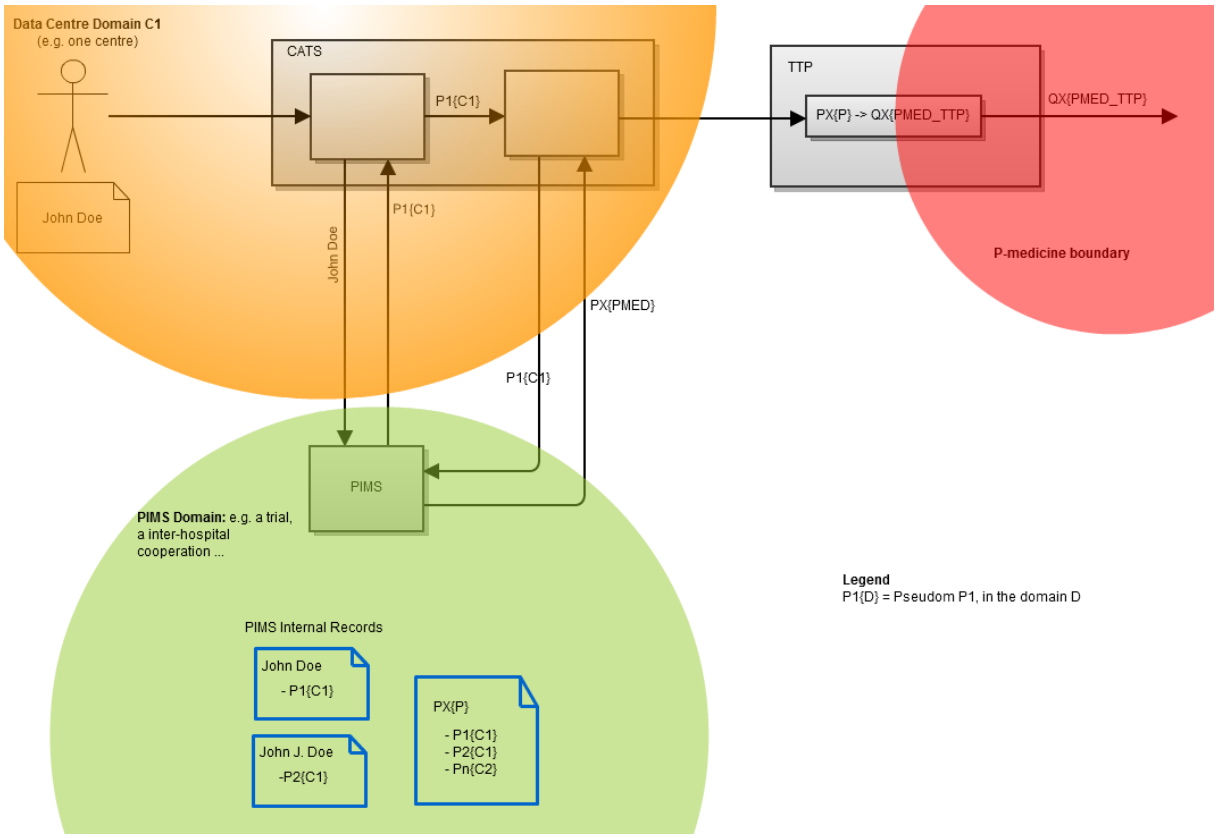


Figure 12 Mapping between pseudonyms

Data centers upload their data through CAT or CATS into p-medicine (Figure 12). CAT, which is an application or service (CATS) that runs locally within the data centre domain (domain C1), will remove all identifying data (attributes) of the patient information to be uploaded and replace them by pseudonyms retrieved from the Personal Information Management System (PIMS). Feeding PIMS with personal identifying information coming from different sources, allows PIMS to issue pseudonyms to different domains. The pseudonymized data is then uploaded through a trusted third party (TTP) into p-medicine. The TTP will transform (encrypt) the pseudonyms so that the patients cannot be re-identified without going through the TTP. The next use case diagram explains in more detail this anonymization flow by using CAT as a service (CATS).

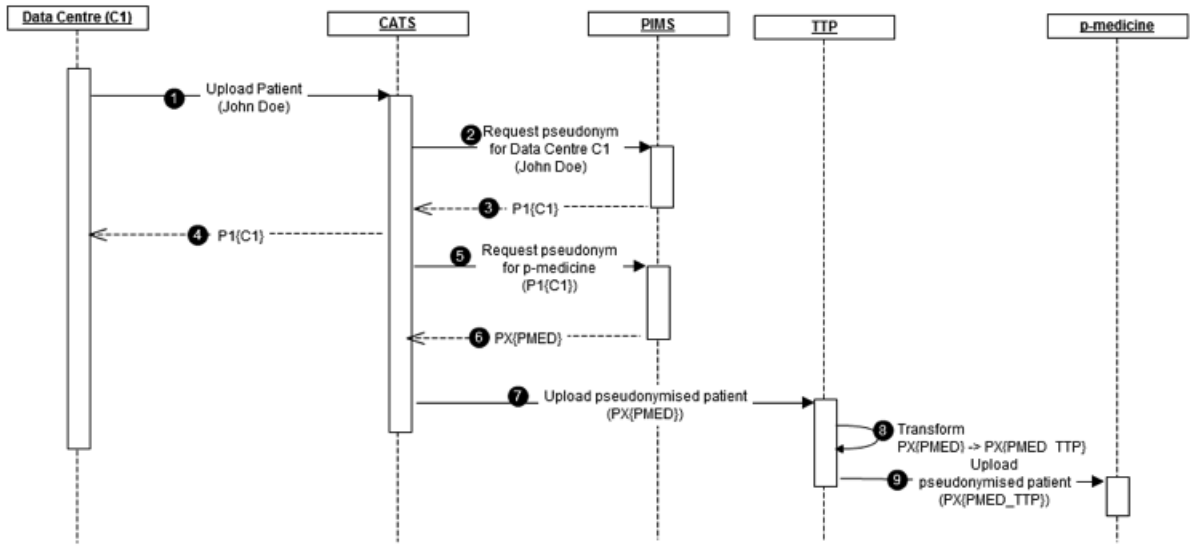


Figure 13 The sequence diagram for the pseudonymization use case

The steps are as follows:

- **1**: A data centre 'C1' uploads data on patient 'John Doe' to CATS.
- **2**: CATS removes all identifying information (attributes) from the uploaded data and requests a pseudonym from PIMS by sending it these identifying attributes.
- **3 & 4**: PIMS returns a pseudonym $P1_{C1}$ to CATS which is then also returned to the data centre for linking.
- **5 & 6**: CATS will then request a pseudonym of '$P1_{C1}$' for p-medicine: 'PX_{PMED}'. If 'John Doe' was uploaded several times by multiple data centres the same p-medicine pseudonym 'PX_{PMED}' will be returned.
- **7**: The pseudonymized data (with pseudonym 'PX_{PMED}') is then uploaded to the trusted third party (TTP) by CATS.
- **8**: Next the TTP will transform the pseudonym 'PX_{PMED}' into a pseudonym 'PX_{PMED_TTP}' through some cryptographic operation (e.g. Hash-based Message Authentication Code - HMAC). This way, the p-medicine users will not be able to link back the pseudonymized data with the original patient. Re-identification is hereby only possible by going through the TTP.
- **9**: The pseudonymized data (with as pseudonym 'PX_{PMED}') is then finally uploaded to p-medicine.

3.3.6.2.3 Dataflow Use Cases

In this section relevant Uses Cases that capture the dataflow in the p-medicine platform will be described and analyzed.

3.3.6.2.3.1 Data Translation for PUSH services

This scenario, described extensively in D2.2 [4], describes the case that a user pushes his data into the p-medicine data warehouse (DW). In order to do that, the data should be translated into the HDOT format. The DW invokes the translation services in the semantic layer, providing the data received and an ontology annotation that permits to translate that data. The semantic layer returns the data in HDOT format. The UML sequence diagram and

the corresponding component diagrams of the specific scenario are shown on Figure 14 and Figure 15.

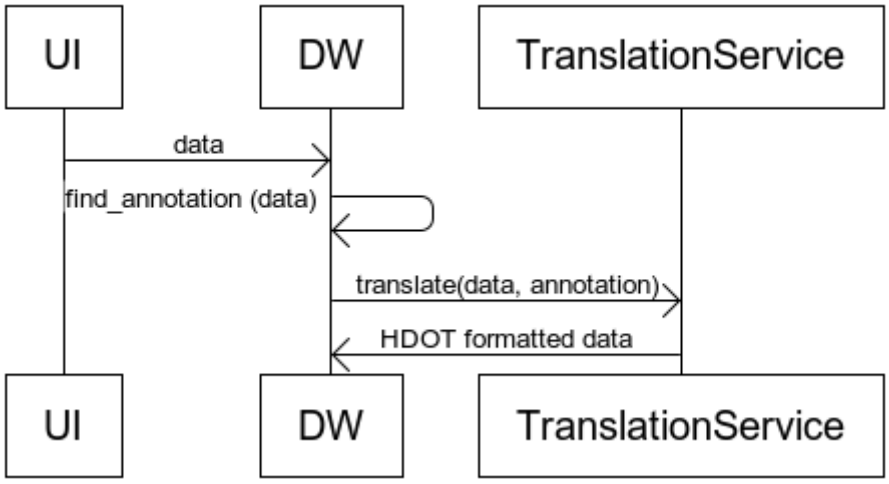


Figure 14 Sequence Diagram of the "*Data Translation for PUSH services*" Use-Case

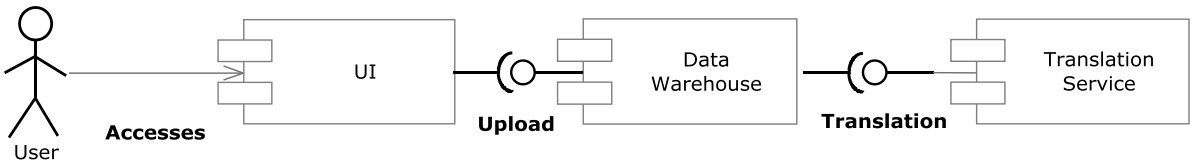


Figure 15. Component Diagram of the "*Data Translation for PUSH services*" Use-Case

The description of each specific component is provided bellow

Component Name	User Interface (UI)
Responsibilities	To provide a user friendly interface to the user in order to publish his data into the data warehouse.
Collaborators	Data Warehouse, Translation Service
Rationale	There should be a component with a user friendly interface to allow data upload.
Issues and notes	Using the UI the user should be able to upload different types of files. Files could be CSV, Excel, Access files etc.

Component Name	Data Warehouse
Responsibilities	To allow the storage of data in a secure, distributed, highly accessible environment. Moreover, the Data Warehouse should be able to provide a data upload service, which should be able to invoke the Translation service when required.
Collaborators	UI, Translation Service
Rationale	There should be a central repository of data, that will allow further

	elaboration on the collected data
Issues and notes	All data in the Data Warehouse should be stored using an HDOT compliant format. Another issue is the interoperability between the data upload service and the translation service.

Component Name	Translation Service
Responsibilities	To translate data in HDOT compliant format. In order to do that the data should be accompanied by the proper meta-data (its schema and the corresponding mapping between the schema and the HDOT). Of course these meta-data once stored can be retrieved from the data warehouse at the runtime and do not need to be provided each time new data are uploaded
Collaborators	Data Warehouse
Rationale	Since data that the users want to store in the central DW might not use the HDOT format, they should be translated in order to be usable.
Issues and notes	If meta-data are not provided with the data, there should be stored in the DW and provided as well.

3.3.6.2.3.2 User uploads DICOM images, after pseudonymizing them through Obtima to DW

This use-case describes how DICOM data can be send from a local hospital to the data warehouse after automatic pseudonymization of the data. In a second step it describes how DICOM data can be downloaded for reviewing or post-processing. The scenario is described in detail in page 327 of D2.2 [4]. Initially the local Physician logs into Obtima and select the DICOM files that wants to submit to the DW. Obtima calls the CATS system to anonymize the images and then it sends them to be stored at the DW. Then a reference radiologist can see the list of images uploaded as pending in Obtima. The images are visualized and then he can then he can write and submit his report to be stored into the system. The corresponding sequence and component diagrams of this use-case are shown on Figure 16 and Figure 17 respectively.

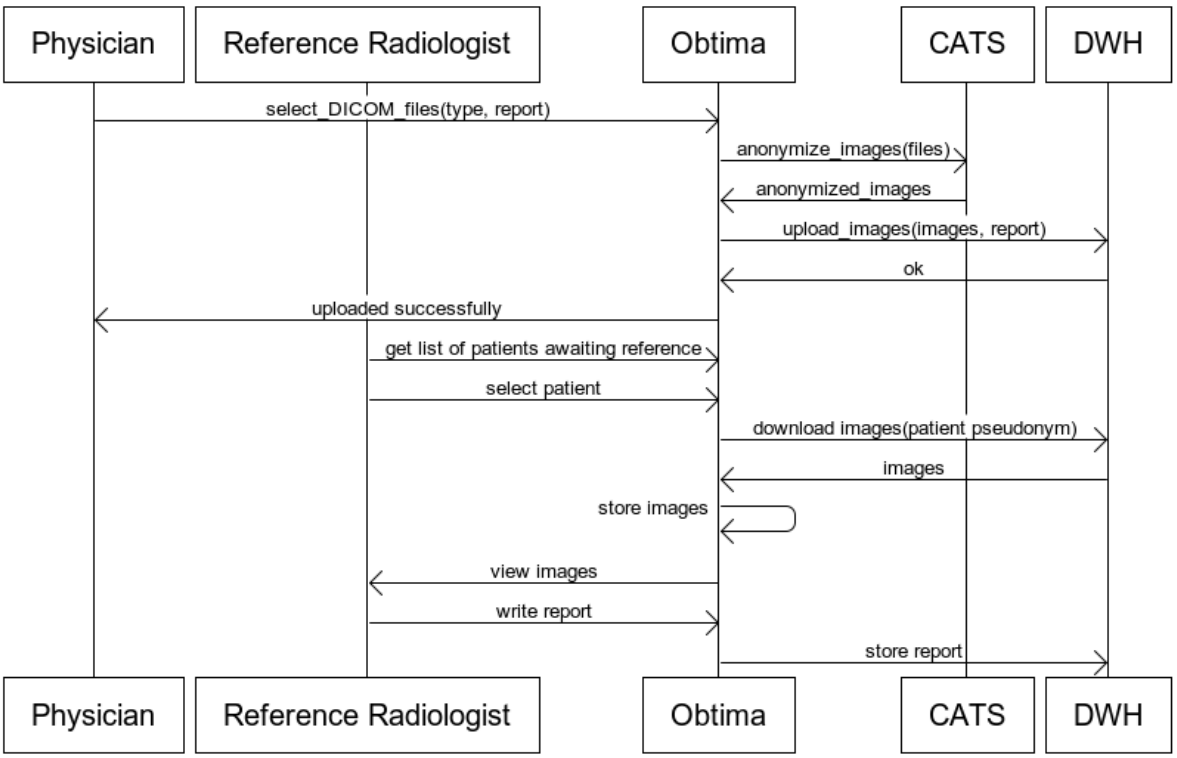


Figure 16 Sequence Diagram of the use-case "*User uploads DICOM images, after pseudonymizing them through Obtima to DW*"

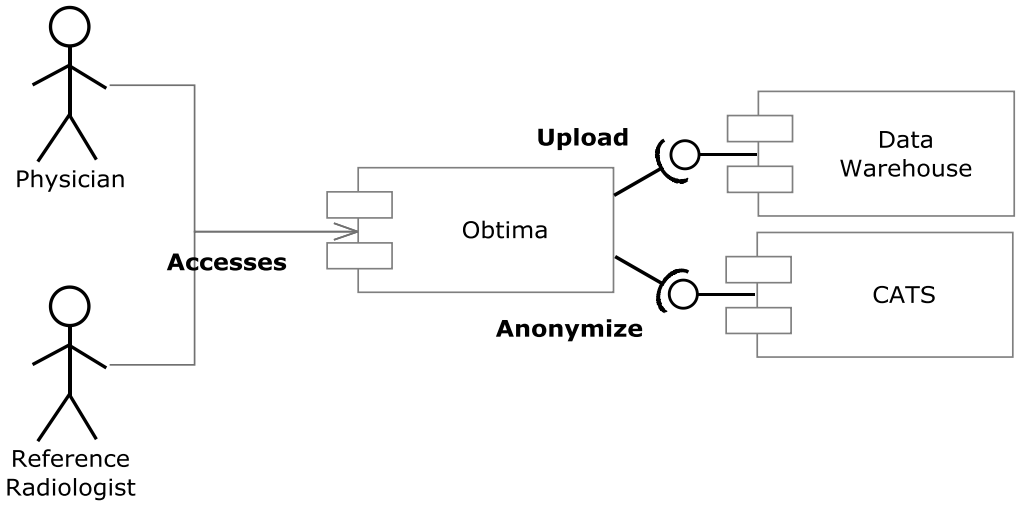


Figure 17 Component Diagram of the use-case "*User uploads DICOM images, after pseudonymizing them through Obtima to DW*"

The description of each specific component is provided bellow

Component Name	Obtima
Responsibilities	To provide a friendly user interface for publishing data and images into the data warehouse, to create, fill and disseminate eCRFs, to manage clinical trials.
Collaborators	Data Warehouse, CATS

Rationale	There should be a component with a user friendly interface to allow eCRF completion, trial management, data and image publishing.
Issues and notes	Obtima should store data in the warehouse in a HDOT compliant format. However, Obtima has also an internal database that needs to be synchronized with the Data Warehouse.

Component Name	Data Warehouse
Responsibilities	To allow the storage of data in a secure, distributed, highly accessible environment. In this specific use case the Data Warehouse is used to store and retrieve DICOM images and reports on them.
Collaborators	Obtima
Rationale	There should be a central repository of data, that will allow further elaboration on the collected data
Issues and notes	All data in the Data Warehouse should be stored using an HDOT compliant format. Moreover, images should be semantically enriched using the HDOT.

Component Name	CATS
Responsibilities	To anonymize imaging data, and to store the identity of each specific datum in order to be able to relate it again to its source.
Collaborators	Obtima
Rationale	The data stored in HDOT should be anonymized in order to be compliant with legislation.
Issues and notes	Specific meta-data should accompany the data to be anonymized in order to identify how the anonymization should be applied.

3.3.6.2.3.3 Ontology annotation of external databases

Annotation of external databases in terms of the HDOT ontology is necessary for data to be stored and integrated in the p-medicine Data Warehouse. The tool will offer data managers a graphical interface to perform this annotation. The interface should be intuitive enough for end users lacking deep RDF understanding to be able to correctly annotate their data. The corresponding use-case is described in detail in page 366 of D2.2 [4].

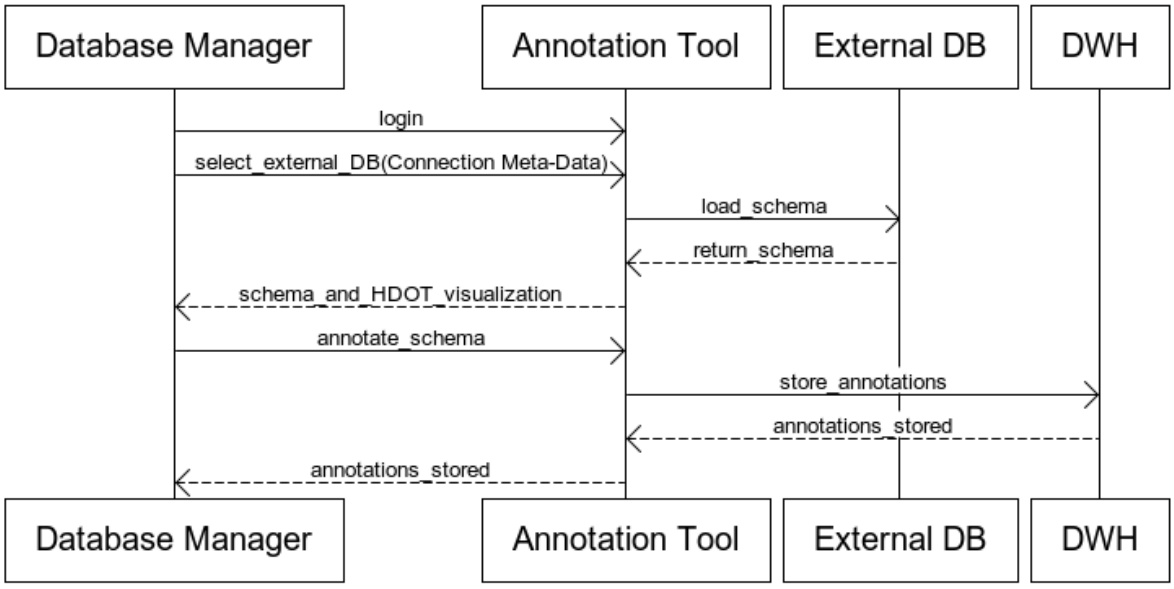


Figure 18 Sequence Diagram of the use-case "*Ontology annotation of external databases*"

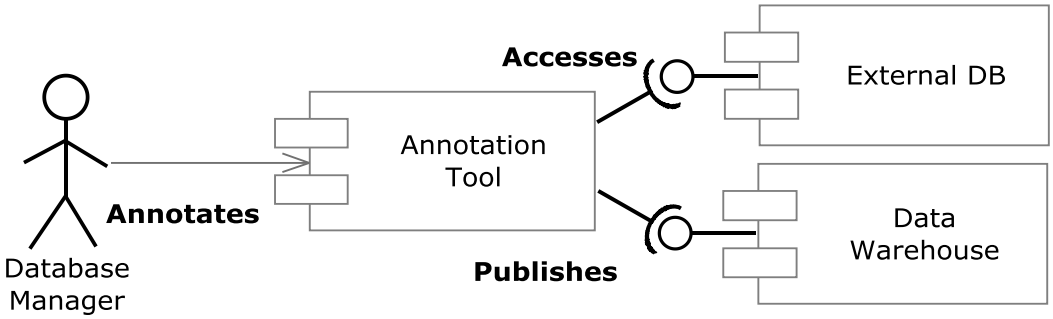


Figure 19 Component Diagram of the use-case "*Ontology annotation of external databases*"

The description of each specific component is provided bellow

Component Name	Annotation Tool
Responsibilities	To provide a user friendly interface to a Database Manager in order to annotate an external Database using the HDOT ontology.
Collaborators	External DB, Data Warehouse
Rationale	Writing textual annotations of a Database is difficult and proper visualization should be provided to the user to be aided to this difficult task.
Issues and notes	The annotation tool might be incorporated into the portal

Component Name	External DB
Responsibilities	External Databases will offer data that are not already stored in the DW

Collaborators	Annotation Tool
Rationale	External Databases should be able to be integrated/queried/used by the main p-medicine infrastructure
Issues and notes	Since the Databases are external to the p-medicine, their availability is in question. So data might have to be extracted, transformed according to the annotations and loaded to the Data Warehouse, but this remains to be decided.

Component Name	Data Warehouse
Responsibilities	To allow the storage of data in a secure, distributed, highly accessible environment. In this use-case the warehouse will be used to store the database annotations in the warehouse for future reuse.
Collaborators	Annotation Tool
Rationale	There should be a central repository of data, that will allow further elaboration on the collected data
Issues and notes	The annotations should be stored in a specific format

3.3.6.2.3.4 Gene expression and clinical data analysis from 1 or more trials through Obtima

This use case describes how clinical data from a clinical trial can be statistically analysed together with molecular data within Obtima. The scenario is described in detail in page 344 of D2.2 [4] and presents the interaction between a physician and the Obtima in order to combine clinical data stored in Obtima and molecular data stored in DW to perform data analysis. As shown in Figure 20 the physician first selects the trial(s) and then the corresponding CRF fields. By selecting the trial(s) a list of molecular data corresponding to the selected cohort is downloaded from the DW. Then the Physician selects the proper tool and sets the proper parameters for the following statistical analysis. The data are analysed and a report is returned to the user.

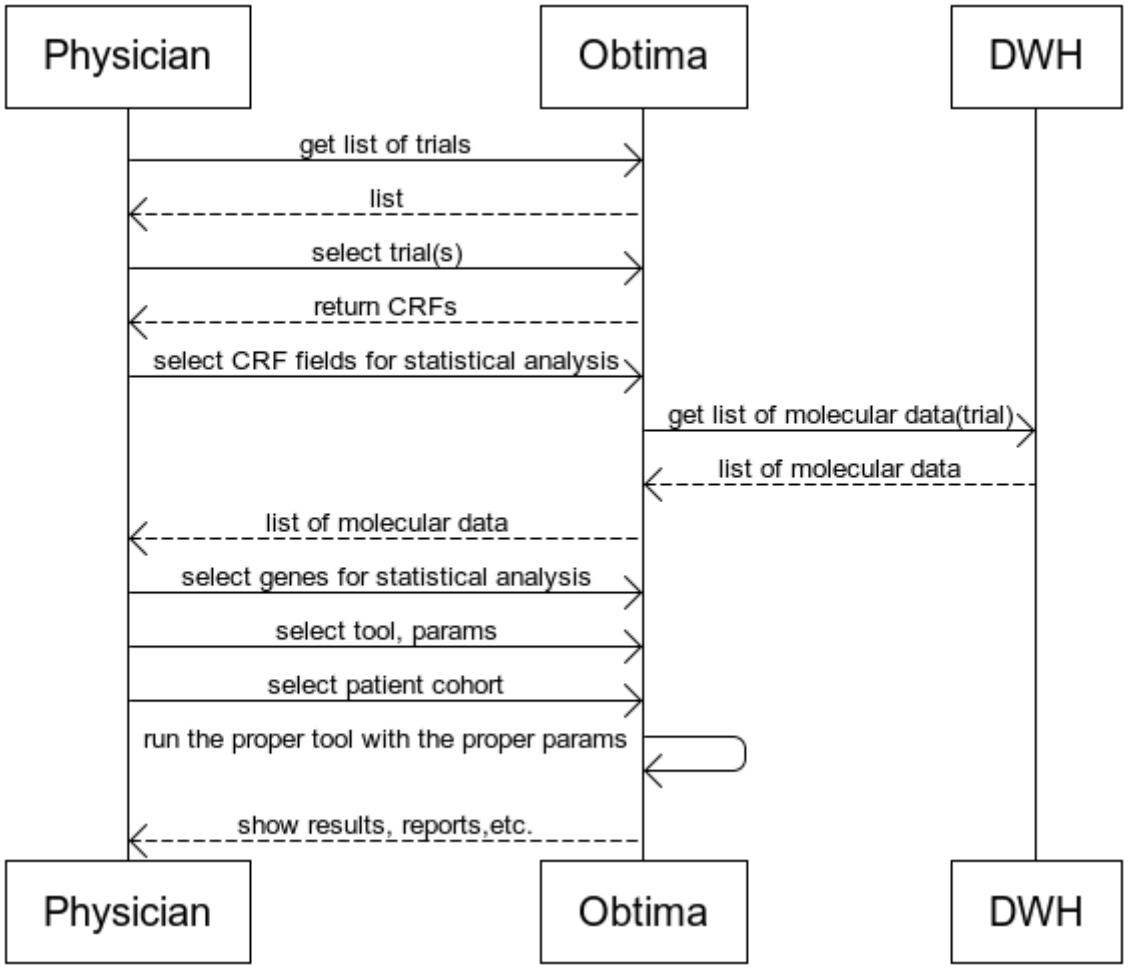


Figure 20 Sequence Diagram of the use-case "Gene expression and clinical data analysis from 1 or more trials through Obtima"

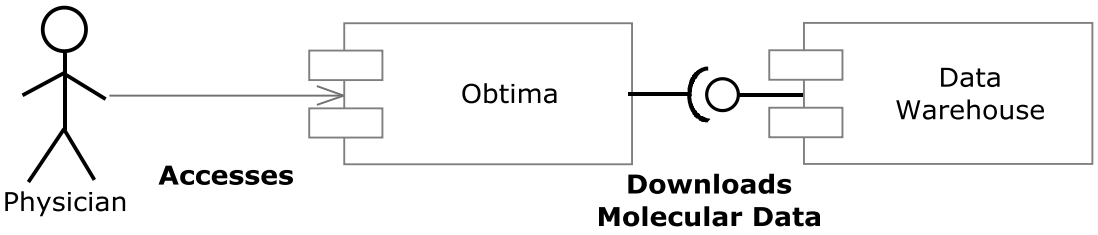


Figure 21 Component Diagram of the use-case "Gene expression and clinical data analysis from 1 or more trials through Obtima"

The component diagram of this use-case is shown on Figure 21, and the corresponding components are described below.

Component Name	Obtima
Responsibilities	To provide a friendly user interface for managing clinical trials and storing clinical information. Moreover, this tool contains several statistical analysis tools, and can access molecular data as well from the DW. In this use-case scenario will be used to combine the analysis tools and to provide analysis reports to the user.

Collaborators	Data Warehouse
Rationale	There should be a component with a user friendly interface to allow eCRF completion, trial management, data and image publishing and analysis.
Issues and notes	Obtima should be able to access data from the DW and combine them with the data from its own database. So Obtima's internal database should be HDOT compliant as well or the proper mappings to the HDOT should be made available and used.

Component Name	Data Warehouse
Responsibilities	To allow the storage of data in a secure, distributed, highly accessible environment. In this use case molecular data are retrieved from the warehouse.
Collaborators	Obtima
Rationale	There should be a central repository of data, that will allow further elaboration on the collected data
Issues and notes	All data in the Data Warehouse should be stored using an HDOT compliant format and proper interfaces should be provided for data access, in a secure manner.

3.3.6.2.3.5 Pathway scenario for patient empowerment: Informed consent

This scenario describes the use-case, documented in page 343 of D2.2 [4], where a patient is able to provide, withdraw and manage consent for clinical trials. Actually the patient is able to login to the patient portal, where a list of relevant trial questions is displayed. The user moves through the information and questions providing the answers, and then he electronically signs the information gathered. The sequence diagram of the described use-case is shown in Figure 22 and the corresponding component diagram in Figure 23

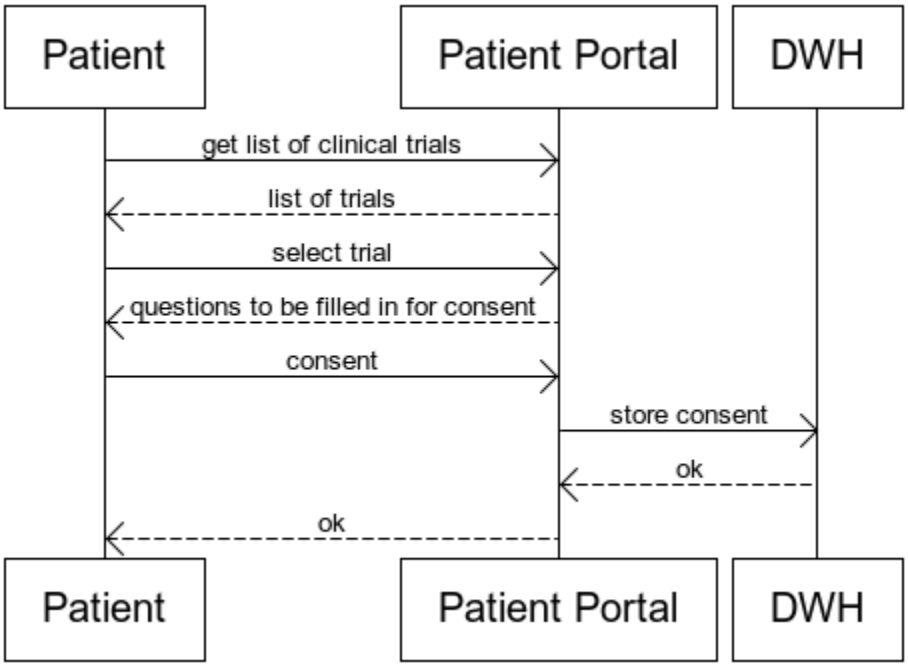


Figure 22 Sequence Diagram of the use-case "*Pathway scenario for patient empowerment: Informed consent*"

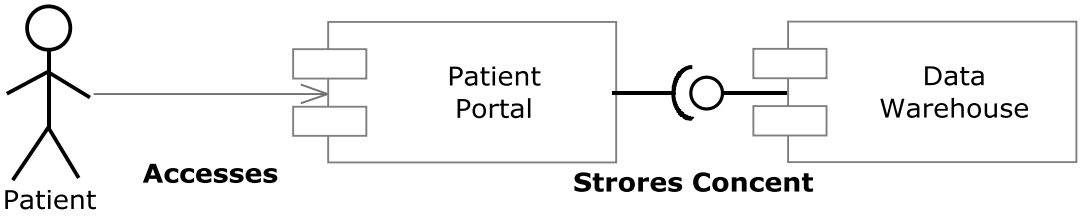


Figure 23 Component Diagram of the use-case "*Pathway scenario for patient empowerment: Informed consent*"

The components shown in Figure 23 are described below:

Component Name	Patient Portal
Responsibilities	To provide a user-friendly interface to patients in order to access information stored in the DW. To visualize patient data and manage consent and re-consent.
Collaborators	Data Warehouse
Rationale	There should be a component with a user friendly interface to allow Consent editing.
Issues and notes	The consent should be electronically signed.

Component Name	Data Warehouse
Responsibilities	To allow the storage of data in a secure, distributed, highly accessible environment.

Collaborators	Patient Portal
Rationale	There should be a central repository of data, that will allow further elaboration on the collected data
Issues and notes	All data in the Data Warehouse should be stored using an HDOT compliant format and proper interfaces should be provided for data access, in a secure manner. Moreover, since the consent might be changed or resubmitted after some time, temporal information about consent signing should be stored as well. Finally according to the answers to consent the specific patient's data should be used or not at the analysis.

3.3.6.2.3.6 Components for the Dataflow use-cases

By combining all use cases above the components that are involved in p-medicine dataflow are shown in Figure 24. Each one of those components has been described in previous sub-sections.

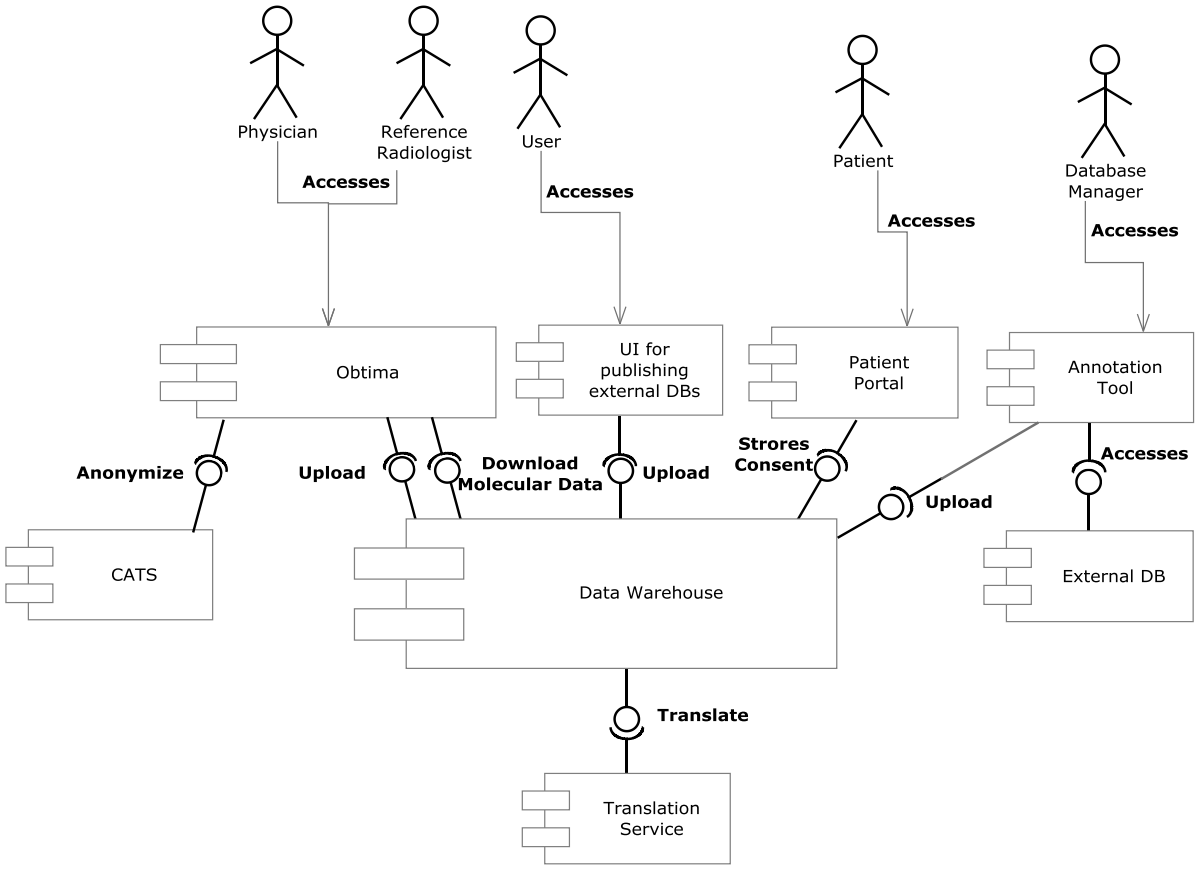


Figure 24 Component Diagram for the Data-flow use-cases

3.3.6.2.4 Clinically Oriented Use Cases

With respect to the architecture of p-medicine, the supported clinical use cases follow a common pattern for the realization of their scenarios. The challenge is to implement an architecture that will be able to support all the scenarios to the three supported cancer domains. To achieve this we need:

- Smooth link and integration of data.
- Tools that will be domain independent.

Clinical use cases aim to assist physician in clinical decision support or to improve the physician's knowledge based on data mining analytical tools on large distributed and heterogeneous data repositories. Using the P-medicine portal the physician has access to data mining user interface UI or to the clinical decision support system (CDS) UI. User has not direct access to databases or to analytical tools. Data mining UI and CDS UI are user-friendly frontends that aim to:

- Hide the complexity of the supported analytical tools
- Hide the complexity of the access, annotation and integration of data from various data sources.

The physician can select and combine data for the analysis from:

- The p-medicine data warehouse.
- The Obtima clinical trial system.
- External online or local databases (which are supported by the external DB access tool).

Also a wide range of analytical tools (e.g. R statistics, data mining pattern service, literature mining services, workflow environment) from the data mining and workflow engine tool or the clinical decision support tool can be selected for the analysis of the data. The user can also select complete analysis solutions (workflows) already available into the p-medicine benchmark.

Data is processed at the data mining workflow engine or the CDS engine and the results of the analysis are visualized to the user. The user can also store the results and the analysis procedure (workflow) into the data warehouse.

The general sequence diagram for all the clinical use cases can be found in Figure 25 and the general component diagram for all the clinical use cases can be found in Figure 26.

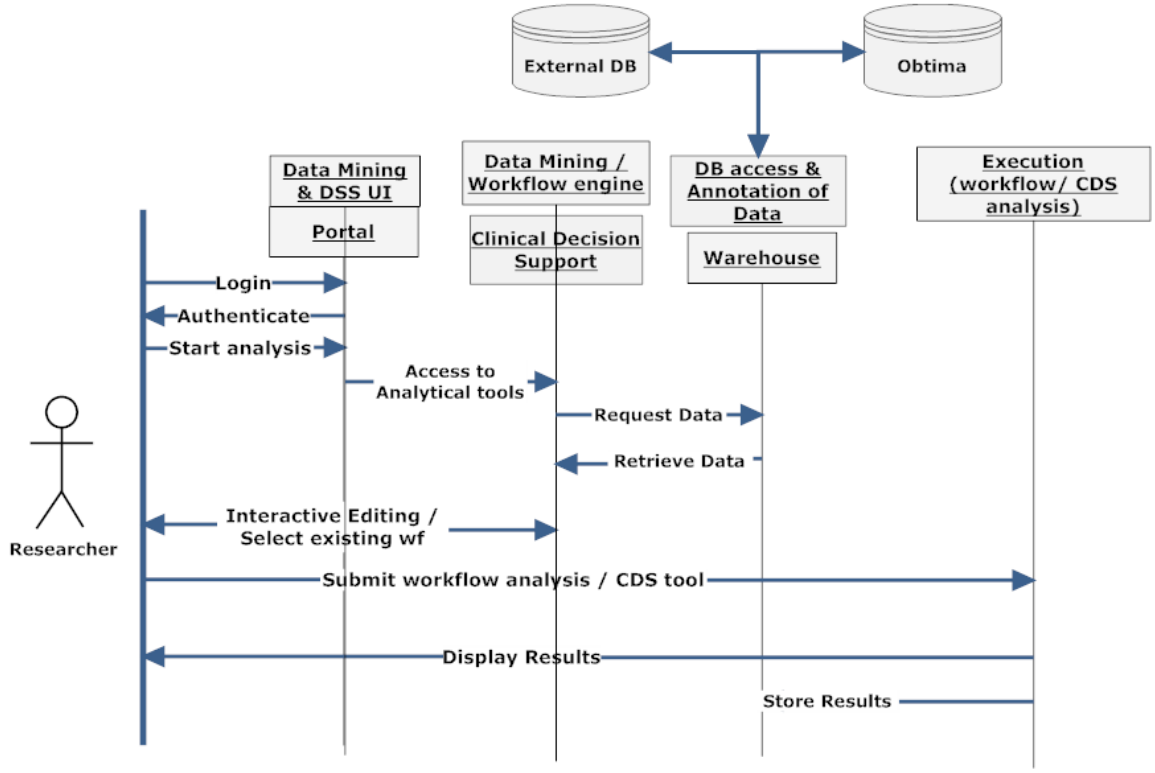


Figure 25: General sequence diagram for all the clinical use cases

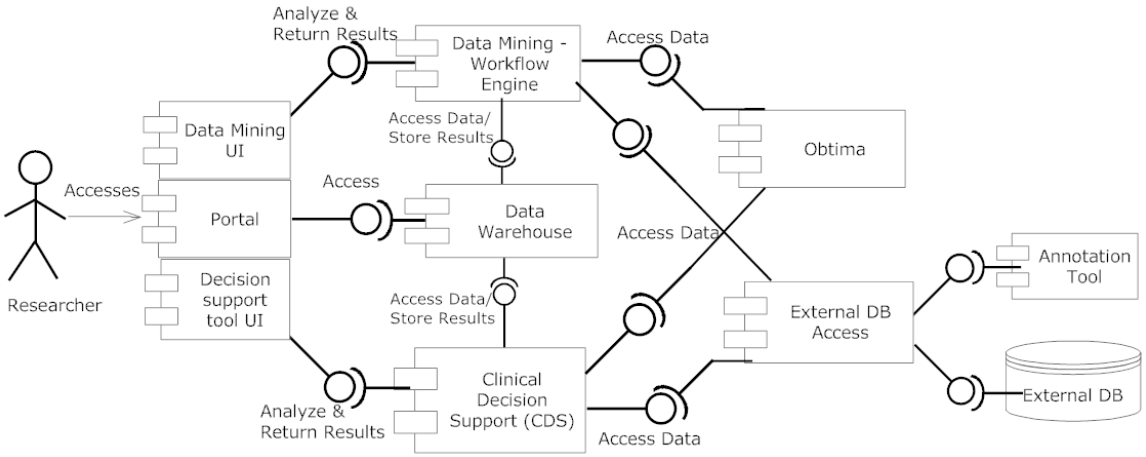


Figure 26: General component diagram for all the clinical use cases

The components shown in Figure 26 are described below:

Component Name	Portal
Responsibilities	To provide a user-friendly interface to users in order to access information stored in the DW.
Collaborators	Data Warehouse

Rationale	There should be a component with a user friendly interface to allow authorization/roles-rights of the user.
Issues and notes	

Component Name	Data Warehouse
Responsibilities	To allow the storage of data in a secure, distributed, highly accessible environment.
Collaborators	Annotation Tool
Rationale	There should be a central repository of data, that will allow further elaboration on the collected data
Issues and notes	The annotations should be stored in a specific format

Component Name	Obtima
Responsibilities	To manage clinical trials and to store clinical information. Moreover, this tool contains several statistical analysis tools, and can access molecular data as well from the DW.
Collaborators	Data Warehouse
Rationale	
Issues and notes	Obtima should be able to access data from the DW

Component Name	Data-Mining Workflow Execution Environment
Responsibilities	To execute Data-Mining executable steps
Collaborators	Data-Mining Pattern Server
Rationale	There should be an execution engine for the Data-Mining executable steps
Issues and notes	The executable steps should be first validated from the Data-Mining Pattern Server, and there should be proper messaging system in case runtime errors occur.

Component Name	Clinical Decision Support (CDS)
Responsibilities	To develop tools able to support the clinicians to efficiently access all relevant data and infer knowledge necessary to reach

	the most accurate diagnosis and prescribe the most suitable treatment
Collaborators	Decision Support Tool User Interface
Rationale	There should be an execution engine for the CDS
Issues and notes	

Component Name	Annotation Tool
Responsibilities	To provide a user friendly interface to a Database Manager in order to annotate an external Database using the HDOT ontology.
Collaborators	External DB, Data Warehouse
Rationale	Writing textual annotations of a Database is difficult and proper visualization should be provided to the user to be aided to this difficult task.
Issues and notes	The annotation tool might be incorporated into the portal

Component Name	External DB access
Responsibilities	External Databases will offer data that are not already stored in the DW
Collaborators	Annotation Tool
Rationale	External Databases should be able to be integrated/queried/used by the main p-medicine infrastructure
Issues and notes	Since the Databases are external to the p-medicine, their availability is in question. So data might have to be extracted, transformed according to the annotations and loaded to the Data Warehouse, but this remains to be decided.

Component Name	External DB
Responsibilities	Interface to access the External database (e.g. uri)
Collaborators	Data Warehouse
Rationale	To enrich knowledge with more data.
Issues and notes	Security

3.3.6.2.4.1.1 Oncosimulator

This scenario describes the case that a user executes an oncological simulation. The user is able to choose one of the simulation models implemented in p-Medicine (and described in detail in D.12.1) i.e. OS-BRCA for Breast Cancer (two clinical trials involved), OS-WT for Wilms tumor (Nephroblastoma), OS-ALL for Acute Lymphoblastic Leukemia, and provide the appropriate input data. At the end of the execution the user retrieves the simulation results, visualized, if possible. The UML sequence diagram and the corresponding component diagram of the specific scenario are shown in Figure 27 and Figure 28.

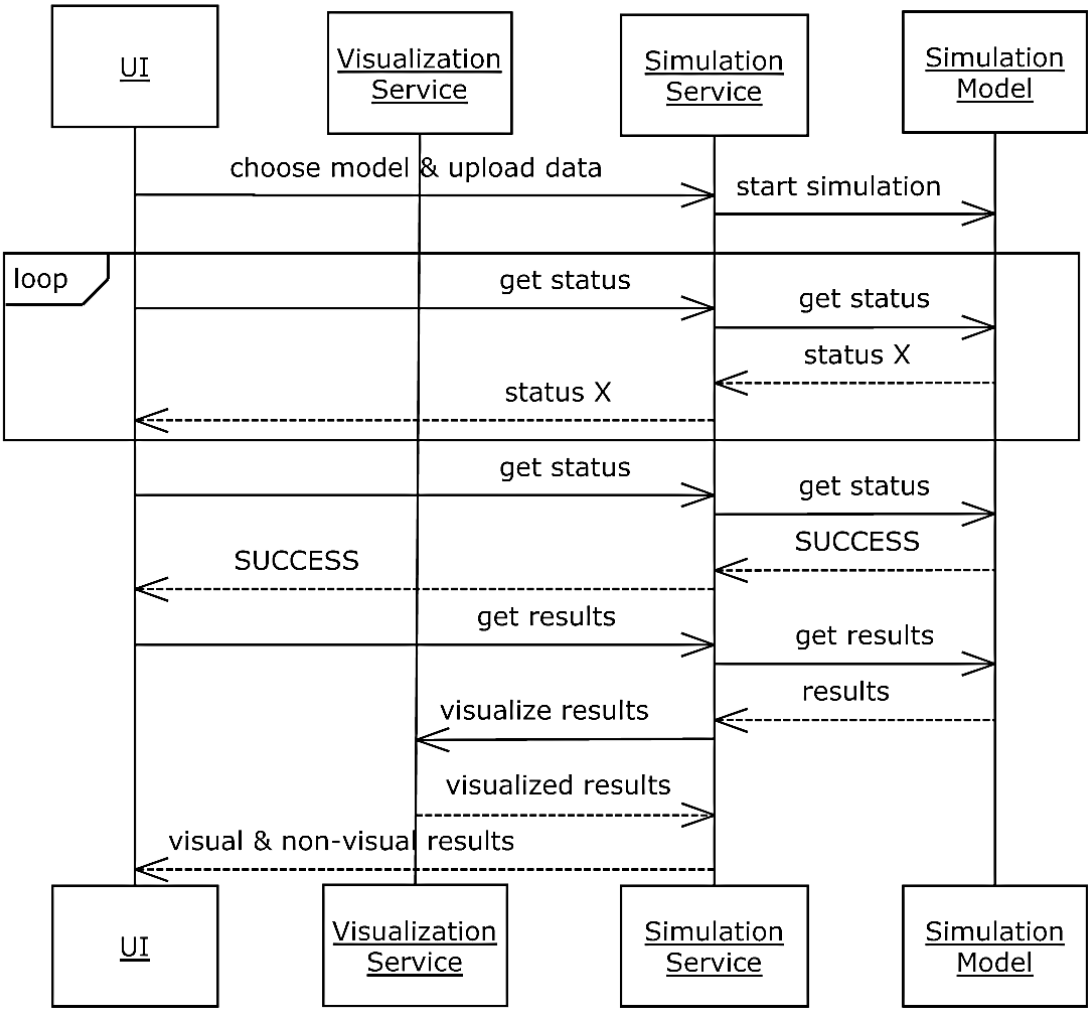


Figure 27 The sequence diagram for the Oncosimulator use case

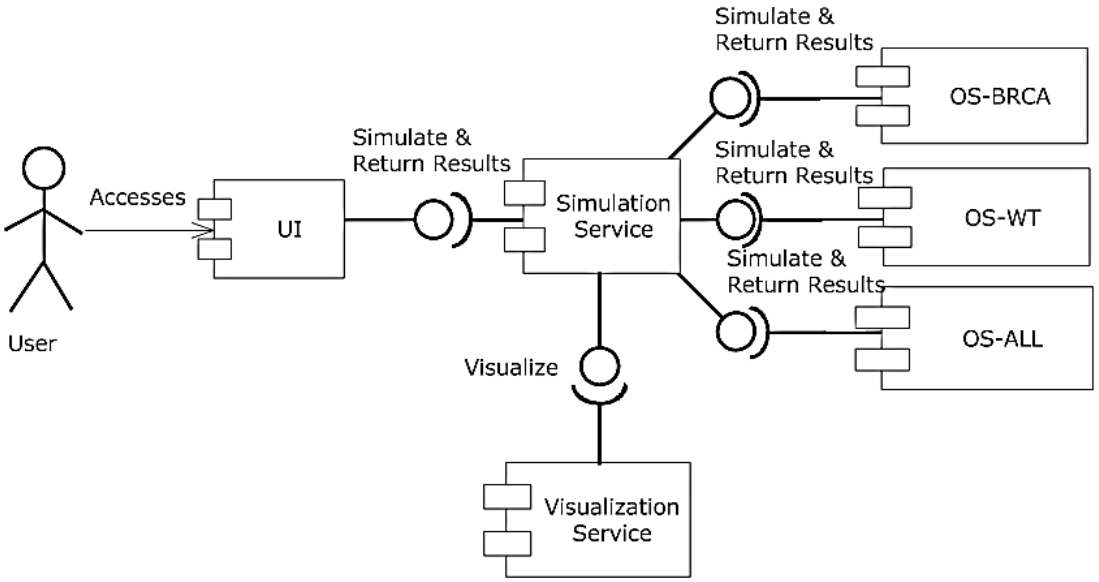


Figure 28 Component diagram for the Oncosimulator use case

The description of each specific component is provided bellow

Component Name	User Interface (UI)
Responsibilities	To provide a graphical interface to the user in order to choose a specific onco-simulation scenario, to configure the simulation model, to upload the input data and to retrieve/present the simulation results.
Collaborators	Simulation Service
Rationale	There should be a component with a user friendly graphical interface in order for the user to interact with the simulation models.
Issues and notes	Using the UI the user should be able to upload different types of files. Files could be csv, DICOM, raw images, zip etc.

Component Name	Simulation Service
Responsibilities	To choose the appropriate simulation model and provide the input data for the model according to the information provided by the UI, to trigger the start of the simulation, to detect the end of the simulation, to provide the output data or the error description to the UI.
Collaborators	UI, Visualization Service, OS-BRCA, OS-WT, OS-ALL
Rationale	There should be a central simulator service that would be responsible for the execution of the onco-simulations.
Issues and notes	The Simulation Service should communicate with the collaborators with a standardized way.

Component Name	Visualization Service
Responsibilities	To visualize the results of the onco-simulations.
Collaborators	Simulation Service
Rationale	The results of the onco-simulations that correspond to visual information must be visually presented to the user.
Issues and notes	The Visualization Service should communicate with the Simulation Service in a standardized way.

Component Name	Breast Cancer branch of Oncosimulator (OS-BRCA)
Responsibilities	To simulate the tumor growth and treatment response in the case of breast cancer.
Collaborators	Simulation Service
Rationale	There should be a clinically oriented multiscale model of breast cancer.
Issues and notes	During the simulation, in regular short time intervals, information concerning the progress of simulation should be provided. The final ending status (successful or erroneous) must also be provided.

Component Name	The Acute Lymphoblastic Leukemia branch of the Oncosimulator (OS-ALL)
Responsibilities	To simulate the evolution and treatment response in the case of acute lymphoblastic leukemia.
Collaborators	Simulation Service
Rationale	There should be a model simulating the temporal evolution and response to therapy of a non-solid type of cancer, such as acute lymphoblastic leukemia.
Issues and notes	During the simulation, in regular short time intervals, information concerning the progress of simulation should be provided. The final ending status (successful or erroneous) must also be provided.

Component Name	The Wilms tumour (Nephroblastoma) branch of Oncosimulator (OS-WT)
Responsibilities	To simulate the tumor growth and treatment response in the case of nephroblastoma.

Collaborators	Simulation Service
Rationale	There should be a clinically-oriented multiscale model of nephroblastoma.
Issues and notes	During the simulation, in regular short time intervals, information concerning the progress of simulation should be provided. The final ending status (successful or erroneous) must also be provided.

3.3.6.2.4.2 Patient Empowerment

In this paragraph we describe the use cases involving the patients in interaction with the p-medicine system.

3.3.6.2.4.2.1 Consent and Re-consent

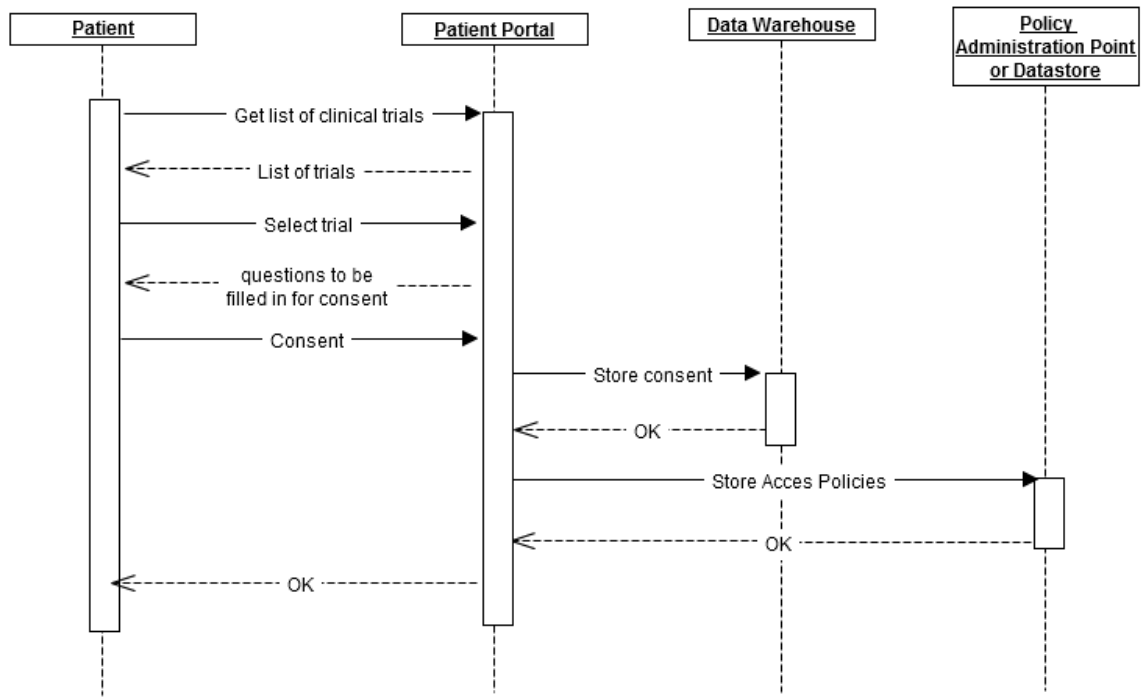


Figure 29 Patient Consent sequence diagram

From the security point of view, when a patient gives his consent on the use of his medical data (or subsets of this medical data) for a specific trial (possibly with added limitations concerning exportability, duration, etc.), a consent policy will be generated and stored on the Policy Administration Point[9]. This policy will allow access to the patient's data for the specific trial within the limitations as given by the patient. The Policy Decision Point[10] will then, when an access request is made, fetch the authorization and consent authorization policies. This way access is given if the user, who requests access, has sufficient rights and if the patient accessed has given his consent.

[9] A Policy Administration Point (PAP) is an endpoint, which manages policies. It will provide a PDP with all policies required to produce an authorisation decision.

[10] A Policy Decision Point (PDP) is an entity that makes authorisation decisions. A PDP accepts authorisation requests and will make a decision based on policies fetched from a Policy Administration Point (PAP).

3.3.6.2.4.2.2 “Searching for clinical trials”/Patient enrolment

There are two possible use cases in which a user can get enrolled into p-medicine.

3.3.6.2.4.2.2.1 Enrolment triggered from p-medicine user management service

It is preferred to enrol users through the p-medicine user management service.

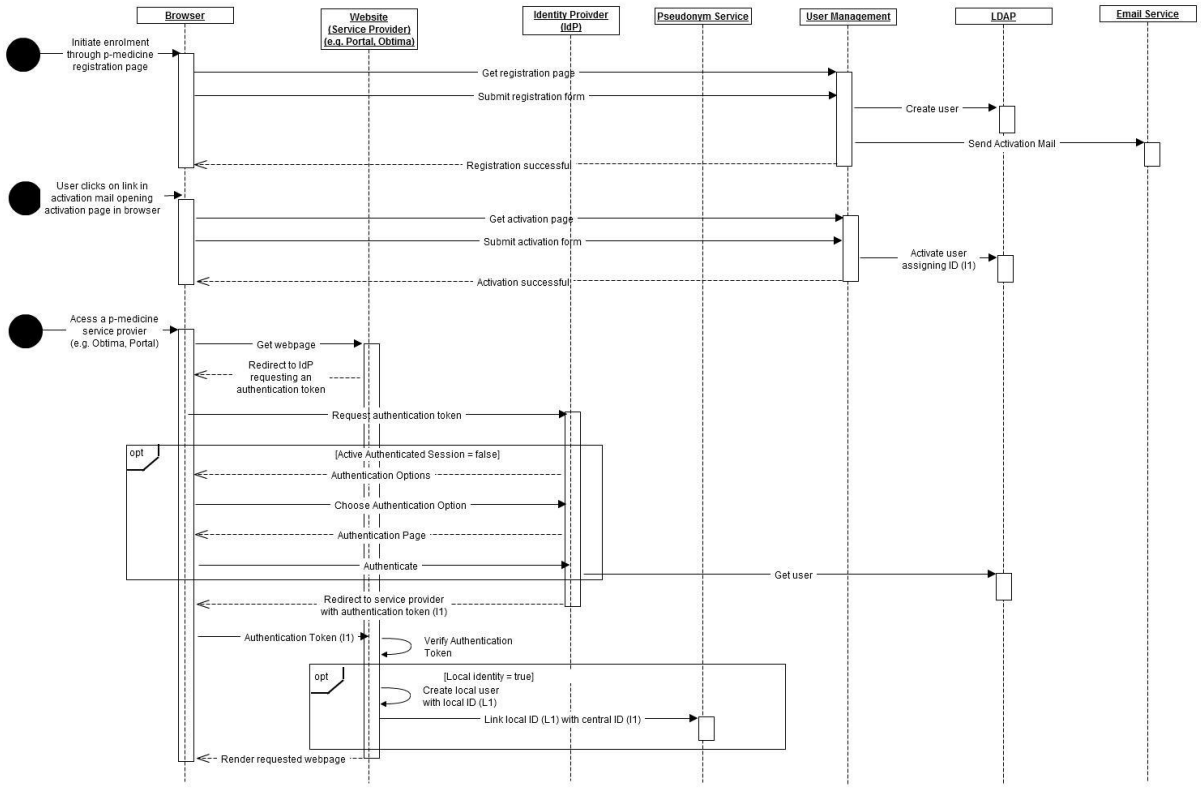


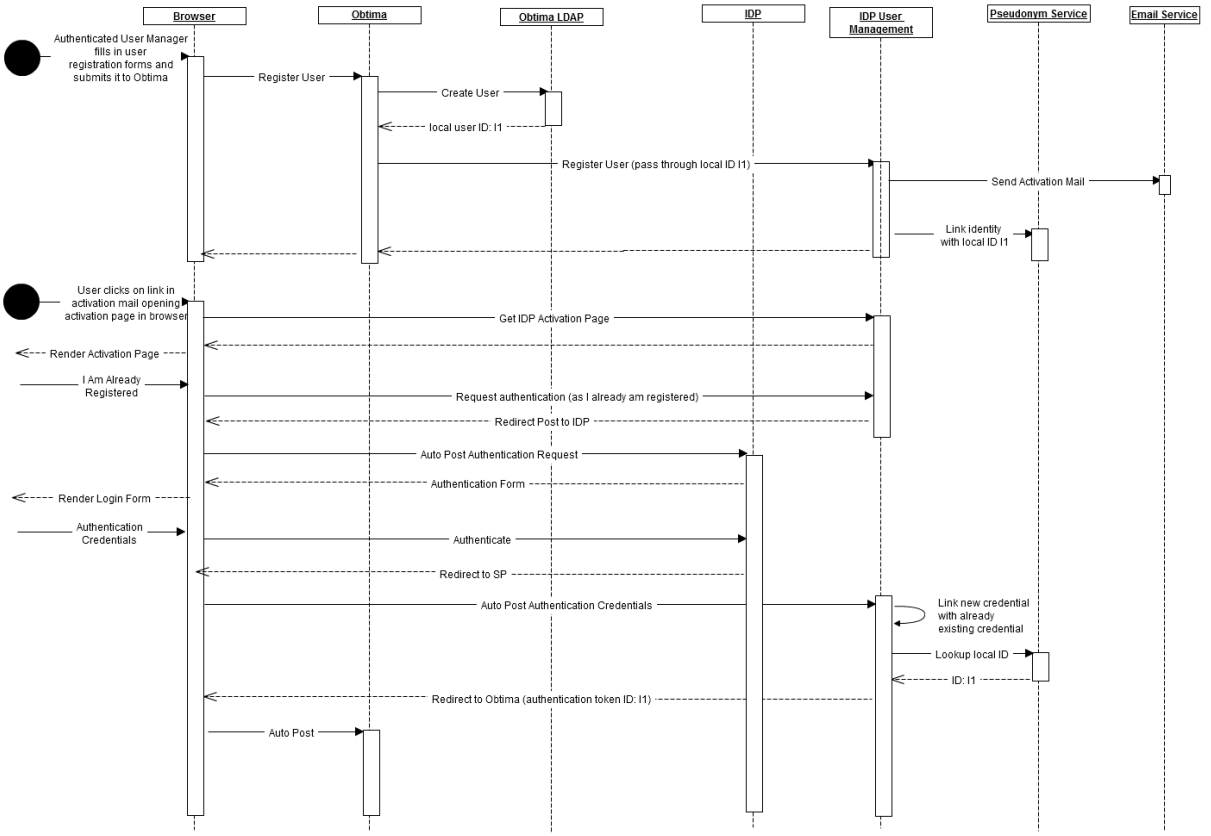
Figure 30 Enrolment of patients through the p-medicine user management service

Three important steps can be distinguished during user enrolment.

1. In the registration step a user visits the p-medicine user registration page to register himself. Upon registration, the user management service creates an inactive user account on LDAP and sends the user an activation mail. Optionally, before the activation mail is sent, a p-medicine administrator could be requested to confirm the user's registration requests.
   Alternatively it can also be a p-medicine administrator who registers the user.
2. Through the activation mail, the user can activate his account (activation step). The user initiates activation by clicking on the "activate" link in the activation mail he received. An activation page will then be rendered where the user can provide all missing required information. After submittal of the activation page, he user is activated an a unique p-medicine identifier is assigned to him.
3. Finally in the last step (local linkage) the activated user can visit any p-medicine service provider (SP). According to the SSO use case, such an SP would redirect the user to the p-medicine IdP to fetch the user's identity assertion (which amongst other attributes will contain the unique p-medicine identifier).
   If this service provider requires a local identity (e.g. Obtima or Portal), during a user's first visit the SP should enrol the user locally. The local identity can then be linked to the central identity by either storing the unique p-medicine identifier in the local database, or by using the central pseudonym service to link the local identifier with the central identifier. By using the pseudonym service, the IdP can directly provide the local identifier upon a later visit.

3.3.6.2.4.2.2.2 Enrolment triggered from a service provider

Alternatively to the central enrolment, a local service provider's administrator can also initiate user enrolment.



The same three steps (registration, activation and local linkage) can be distinguished here.

1. The local administrator creates a user on the service provider. This service provider will then call the p-medicine registration REST service which will register the user within p-medicine and send the user an activation mail. Through this REST call the service provider should also pass the user's local ID so that the (temporary) central identity can be linked with the local one through the central pseudonym service.
2. Through the activation mail, the user can activate his account. The user initiates activation by clicking on the "activate" link in the activation mail he received. An activation page will then be rendered where the user can choose whether he wishes to create a new central identity or link with an already existing one.
    a. If the user chooses to create a new identity an activation form will be rendered where the user can provide all missing required information. After submittal of the activation page, the user is activated and a unique p-medicine identifier is assigned to him.
    b. If the user already has a p-medicine identity he might choose to not create a new one but instead link the newly created identity with his existing one effectively merging both into one.
    For this the user has to authenticate with his existing account to prove it is actually his after which both will be merged together effectively linking the local SP's identity with the already existing central identity.
3. When the user then revisits the service provider he'll be redirected to the Idp. After successful authentication the IdP will return the user's identity assertion. This assertion contains the p-medicine identifier and the local identifier of that user for the

service which was previously provided through the REST registration call. This way the service provider can link the identity assertion with the local identity.

3.3.6.2.5 Knowledge Discovery and Decision Support

3.3.6.2.5.1 Data Mining "patterns"

Data Mining patterns are described in detail in D11.1 [5] we will only provide a short description of the corresponding functional and component view, and the architecture components that should be in place for the benefit of the overall p-medicine architecture.

A data-mining task in p-medicine can be described as workflow or as workflow pattern:

- A *data-mining workflow* is executable in the p-medicine and can invoke any of the p-medicine data mining services and computational execution environments.
- A *workflow pattern* is not executable and serves as a template data mining workflows. It contains steps labeled as "manual step" Each manual step contains a human readable description of required inputs, outputs, and the purpose of this step (e.g. a quality assurance step, in which a user has to assess the quality of inputs and to decide, whether the quality of the input is sufficient to proceed with the next step of the workflow or not). Apart from "manual steps" a user can edit a workflow pattern by replacing manual steps with workflow steps that can be executed automatically. At the end of the editing process, the user must provide valid workflow pattern, this is, all steps of the pattern are either workflow steps that can be executed automatically or are labeled as manual step.

The sequence diagram for the Data-Mining Patterns is shown on Figure 31.

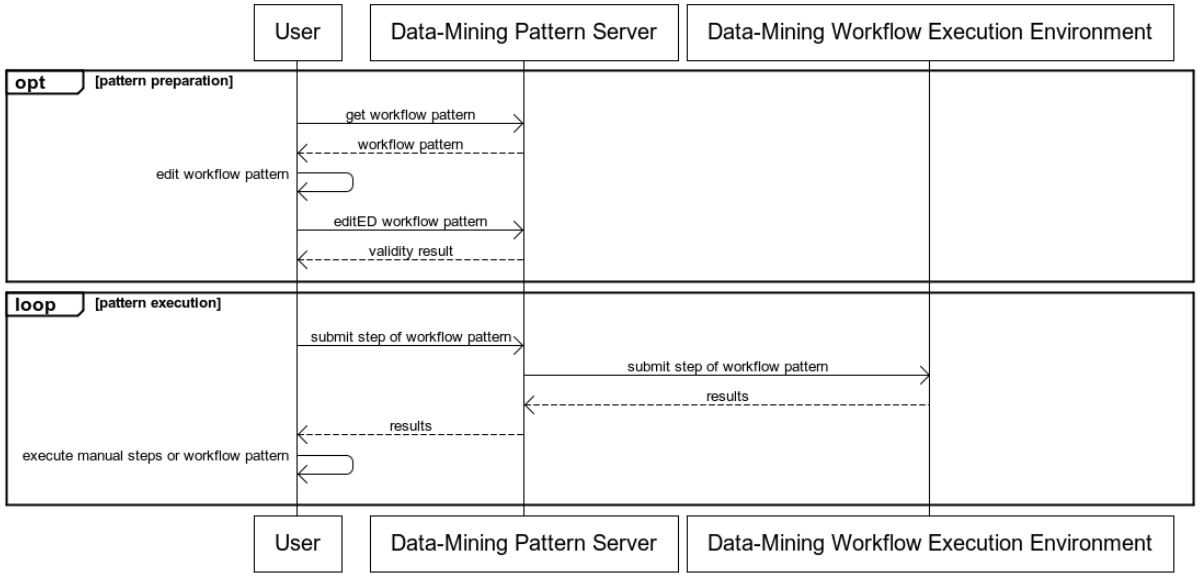


Figure 31 Sequence diagram of the *"Data-Mining Patterns"*

The corresponding diagram is shown on the following Figure.

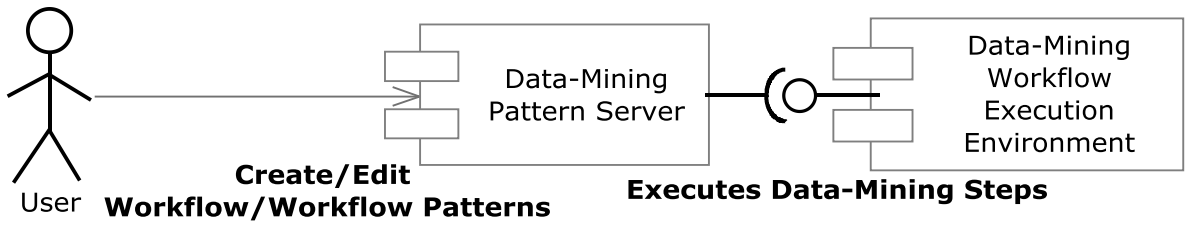


Figure 32 Component Diagrams of the "*Data-Mining Patterns*"

Component Name	Data-Mining Pattern Server
Responsibilities	To store, retrieve, and validate workflow patterns.
Collaborators	Data-Mining Workflow Execution Environment
Rationale	There should be a central repository for the access patterns.
Issues and notes	Data-Mining Patterns should be properly visualized.

Component Name	Data-Mining Workflow Execution Environment
Responsibilities	To execute Data-Mining executable steps
Collaborators	Data-Mining Pattern Server
Rationale	There should be an execution engine for the Data-Mining executable steps
Issues and notes	The executable steps should be first validated from the Data-Mining Pattern Server, and there should be proper messaging system in case runtime errors occur.

An overview of the overall data-mining service architecture, as described in D11.1 [5], is shown also on Figure 33. The idea is that there is a collection of services that can be accessed uniformly from the user-portal interface. The services will actually reuse algorithms encoded in the statistical language R, and a workflow engine will allow the users to profit from the plentitude of the existing workflows shared on public workflow repositories.

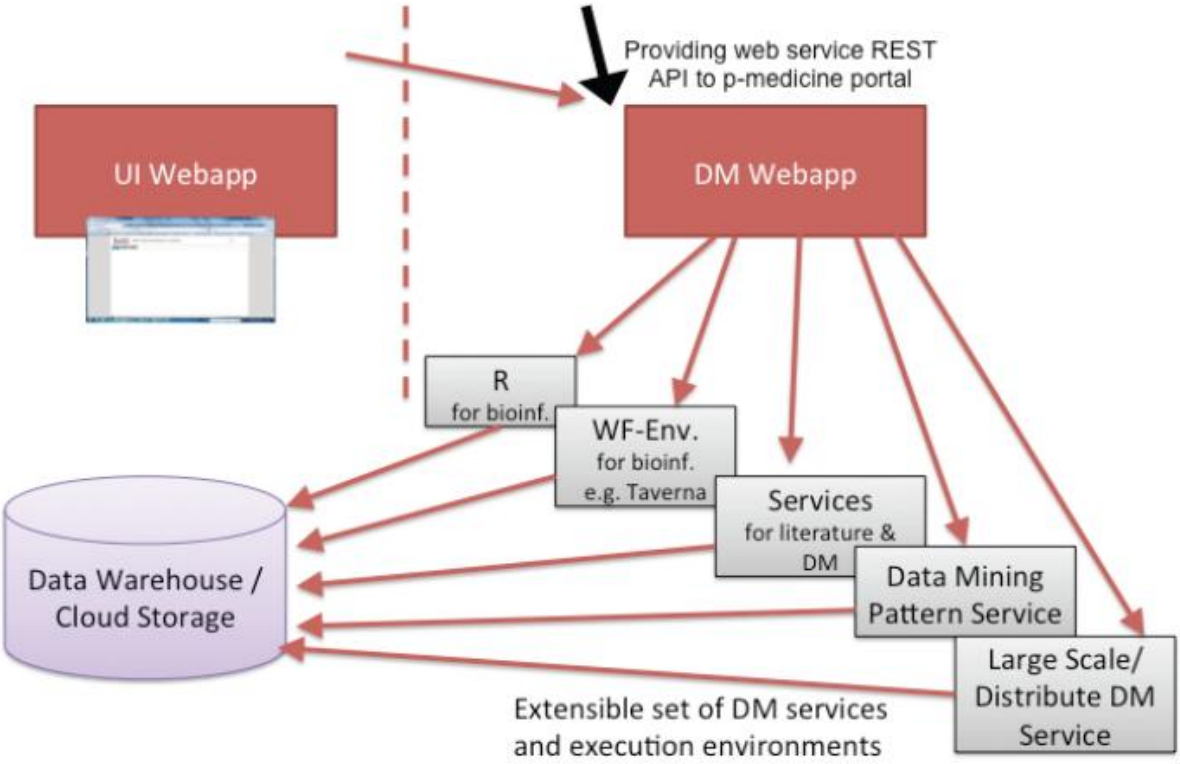


Figure 33 Overview of the data-mining service architecture as shown on D11.1

Besides standard execution environment for data mining in the architecture will be integrated also dedicated data-mining services developed in p-medicine, such as the literature mining service developed in WP11. Further data-mining services can be added, given that they provide a web-based service interface.

3.3.6.2.5.2 Clinical Decision support

Clinical Decision Support is a critical component for organizations seeking to improve the health of the healthcare delivery system. Hospitals, health systems and medical groups already realize that increased patient volume requires more than simply adding staff. It means leveraging technology to improve care quality, access, effectiveness, efficiency and safety, the result of which is better care at lower costs. Many healthcare organizations have implemented CPOE (computerized physician order entry) systems and EHR (electronic health record) systems. Still, challenges remain in system selection, adoption, implementation and use.

Developed together with the P-Medicine clinical users and making use of the latest medical evidence, the p-medicine CDS applications will aim to support the transition from empirical medicine to personalized treatment. The P-Medicine project defined several clinical scenarios in which CDS would be highly beneficial. These will be further refined into technical use cases and user requirements, implemented and evaluated together with the clinical users.

The general requirements are as follows:

- ✓ Patient stratification according to the St.Gallen subtypes
  - Stratification is based on molecular subtypes and is useful in choosing the patient-specific optimal care as well as for risk analysis and prevention
  - Molecular data for patient stratification is not always available

- St.Gallen also provides a good approximation of molecular sub-types using clinico- pathological information, therefore without the need for gene expression analysis data

✓ Prediction, detection and management of severe adverse events
- Prediction based on existing models and on data mining of research data
- Early identification of safety risks
- Efficient reporting of serious adverse events
- Efficient management of the adverse events that have occurred

✓ Evidence-based treatment recommendations
- Linking to relevant knowledge including clinical trials, and published literature
- Finding the appropriate clinical trials for a patients according to their condition
- Access to updated clinical guidelines (such as NCCN, ASCO,etc.) and protocols efficiently represented

In order to be able to provide recommendations, a CDS system first needs to extract the needed data and knowledge with semantics. Therefore, the following challenges need to be overcome:

- Representation and elicitation of medical knowledge. Medical knowledge needs to be automatically extracted from literature, clinical trials and guidelines.
- Linkage to machine-processable semantics, to automatically combine data from multiple sources the understanding of the semantics is essential.
- Structuring the patient data, such as images, free-text reports, and multiple formats used by multiple sites. Standardization of data from multiple sources is therefore needed.
- Integration into the clinical workflow and semantic linkage to EHR. Seamless integration within the care workflow is a key success factor.

A meaningful CDS application therefore integrates multiple sources of data and knowledge. The figure below depicts the complexity of the environment in the case of a CDS tool for prediction and early detection of severe adverse events (AEs).

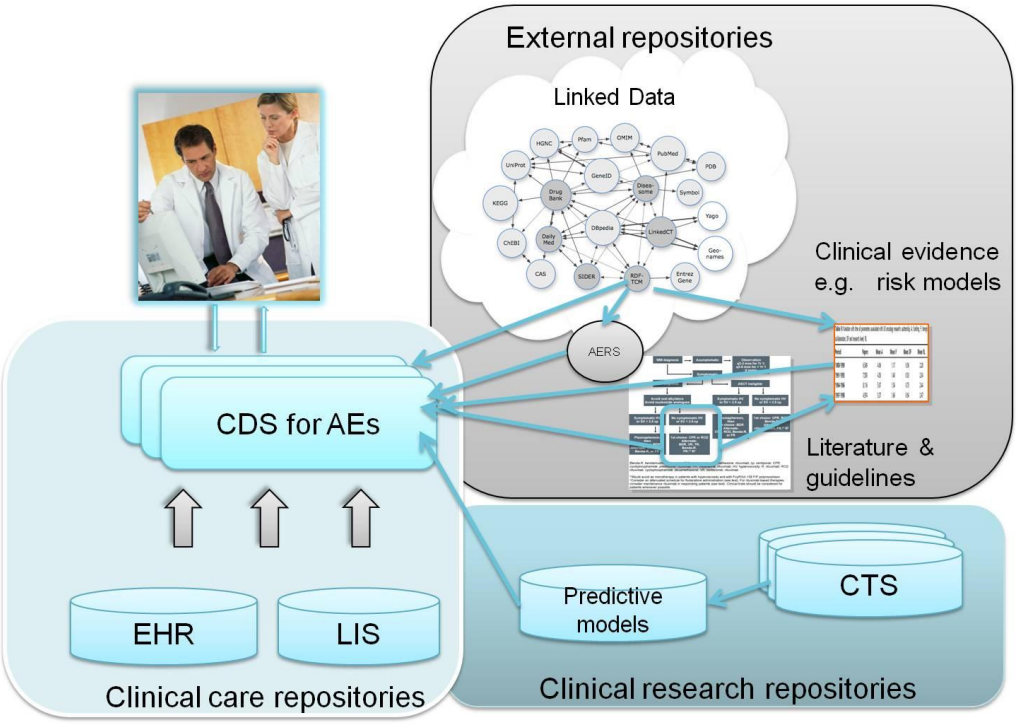


Figure 34 The data integration to support CDS for adverse events

3.3.6.2.6 Biobank Access

Biobanks represent key resources for clinico-genomic research and advances in personalized medicine. Therefore, the sharing of biomaterial is an important functionality of the p-medicine platform. For this purpose an integrated service framework, the Biobank Access Framework, will be developed that will support researchers' growing demand to access and share high quality biomaterial and related data for their research projects. The framework enables and simplifies access to existing biobanks but also supports to offer own biomaterial collections to research communities and encompasses both technical and legal aspects. It harmonizes biomaterial data according to a standard biobank data set and will be integrated seamlessly into the p-medicine platform. The Biobank Access Framework will support the main intended use of biomaterial sharing while aspects like data sharing and linkage of different data resources are covered by the data warehouse and its corresponding data push services and data annotation resources. The development of the Biobank Access Framework will base on four use cases that are described in detail in D2.2 and D10.1. To describe the functional requirements, we have derived three main scenarios from the use cases, which are described in the following sections. We will furthermore describe the main components of the framework that are relevant for the p-medicine architecture.

Offering biomaterial to closed or open research community

A biomaterial owner is supported in offering his biomaterial and related data to open or closed research communities, according to legal aspects. The offered data can be stored in any arbitrary biobank management system. The owner has the possibility to select which of his biomaterial he wants to offer to which research communities. Furthermore, biomaterial owners can push their biomaterial data into the p-medicine data warehouse in order to link the data to other biomedical data sources. To support this scenario a tool will be developed that is called p-BioBank Wrapper. This tool will support a biomaterial owner to upload his biomaterial data into p-BioSPRE, the metabiobank of the Biobank Access Framework (s. below for a more detailed description of p-BioSPRE).

The main interaction of a biobank owner, who wants to offer biomaterial, with the components of the Biobank Access Framework are shown in Figure 35. A biobank owner can upload biomaterial data from one or more of his biobank management systems that need to implement export interfaces according to the standard biobank data set. The uploaded data is pseudonymized according to the p-medicine concept with the CATS service (s. above). The biobank owner can then select the imported data that he wants to share with certain research communities in the p-BioBank Wrapper, specify access restrictions, and upload the data to p-BioSPRE. During this process the data is anonymized. Furthermore, the p-BioBank Wrapper enables a biobank owner to push selected pseudonymized biomaterial data into the p-medicine data warehouse to share it for integration with other biomedical data sources.

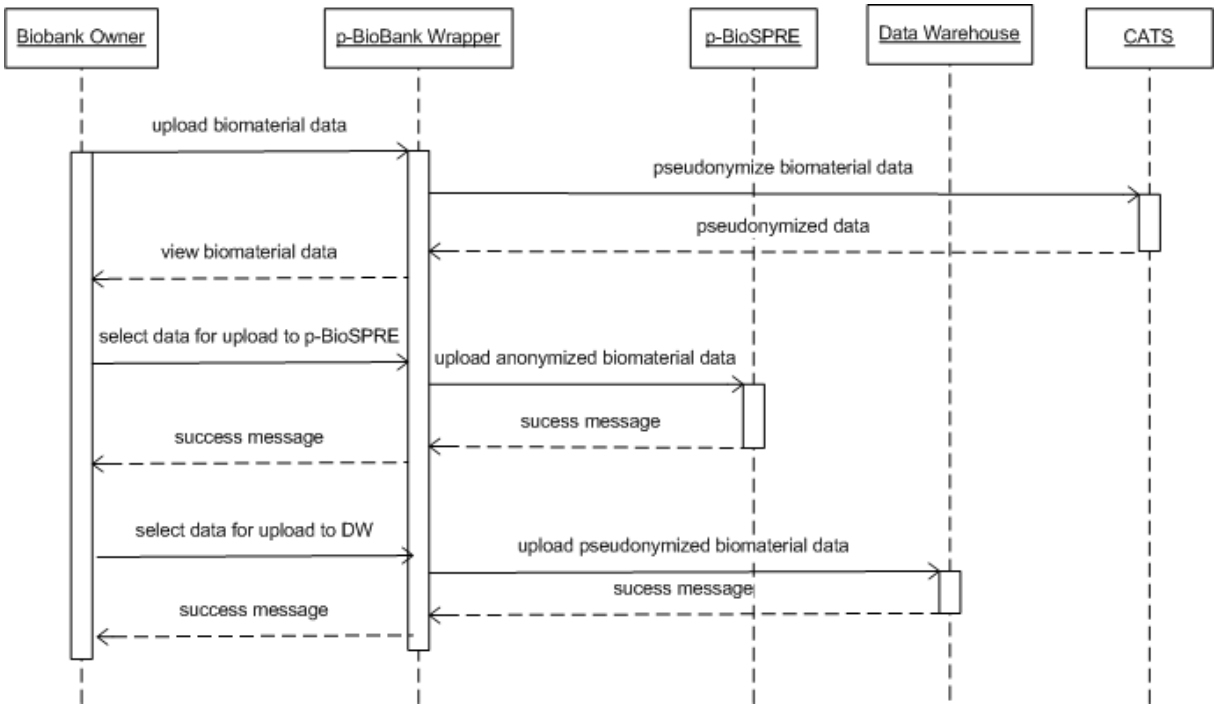


Figure 35 Sequence diagram for main biobank scenario “Offering biomaterial to closed or open research community”.

Searching and requesting biomaterial for research

A researcher is enabled to search the biomaterial that is offered within his communities. He can get information about the available quantity and data that is related to the material. It is furthermore possible for him to request biomaterial for a research project. For this purpose the project needs to be described in detail. The Biobank Access Framework forwards the request to the biomaterial owner. This scenario is mainly supported by p-BioSPRE, the metabiobank of the Biobank Access Framework.

The main interaction of a researcher with p-BioSPRE is shown in Figure 36. P-BioSPRE provides a search interface that enables authorized users to search for biomaterial according to the standard biobank data set. A user is authorized to search biomaterial and related data if he has a p-medicine user account and is a member of the research community the biomaterial is offered to. According to the legal requirements described in D10.1, the biomaterial data that is provided in p-BioSPRE is anonymized. When the user has found appropriate biomaterial for his research, he can request the material. For this purpose p-BioSPRE provides request forms that enable the user to specify the amount of biomaterial he needs and to define his research project in detail. Legal aspects will be presented to the reasearcher (i.e. template of a material transfer agreement, privacy protection guidelines,

responsibility to report about research outcome, etc). P-BioSPRE will then forward the request to the biomaterial owner, who can contact the researcher and decide about the request.

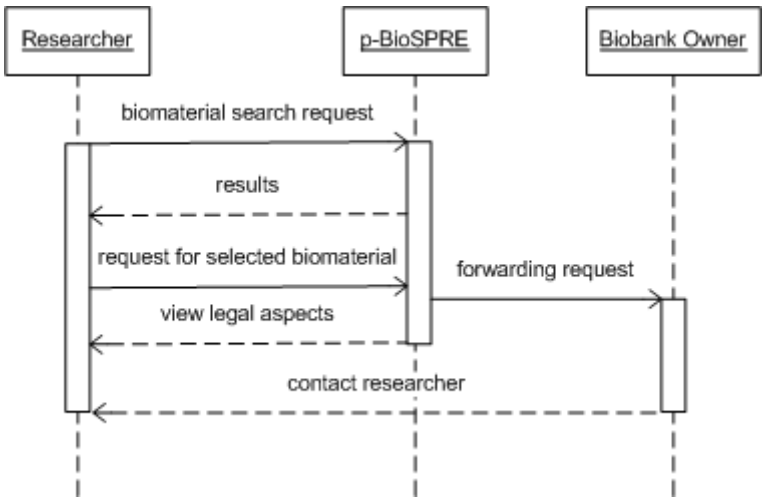


Figure 36 Sequence diagram for main biobank scenario “Searching and requesting biomaterial for research”.

Managing Biomaterial Data in ObTiMA

Users of ObTiMA, the p-medicine’s ontology based trial management system, can manage their biomaterial data within clinical trials. For this purpose pre-defined but adjustable case record forms for patient’s biomaterial according to a standard biobank dataset are provided in ObTiMA. The biomaterial data can be integrated with clinical data within a trial or across several trials for further analysis. Legacy biomaterial data can be imported into ObTiMA from excel files that comply with the standard biobank dataset.

The main interaction of an ObTiMA user with the Trial Biomaterial Manager is shown in Figure 37. In order to set up the biobank management component in a trial the user selects and adjusts predefined biobank CRFs for his trial. For this purpose predefined biomaterial CRFs are provided according to the standard biobank data set. The user can then import legacy biomaterial data according to the standard biobank dataset and/or fill in and edit data on the CRFs. Selected biomaterial data can be uploaded to p-BioSPRE. During this process the data is anonymized. Last, the user has the possibility to upload the biomaterial data into the p-medicine data warehouse. Furthermore, the Trial Biomaterial Manager will allow to link clinical data and biomaterial data within clinical trials and across trials for further analysis.

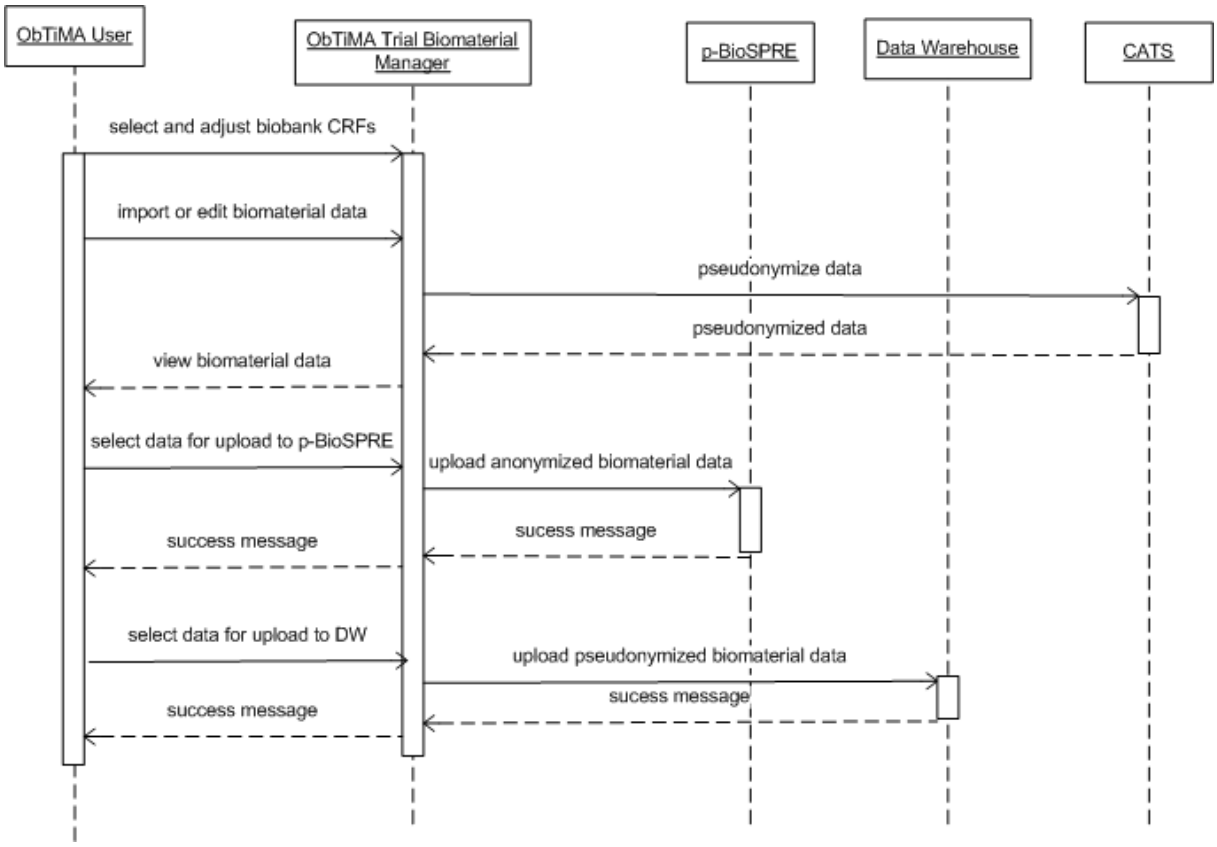


Figure 37 Sequence diagram for main biobank scenario “Managing biomaterial data in ObTiMA”.

A component diagram that depicts the components of the Biobank Access Framework (depicted in orange) and their interaction with other p-medicine components is shown in Figure 38.

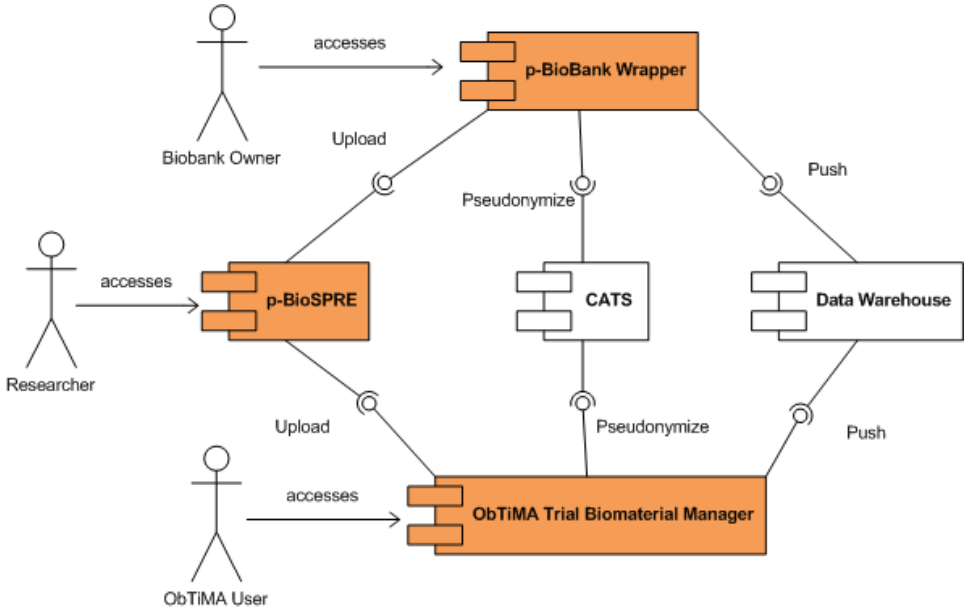


Figure 38 Component Diagram for Biobank Access Framework.

The components of the Biobank Access Framework are described in more detail below:

Component Name	p-BioSPRE (p-medicine Biomaterial Search and Project Request Engine)
Responsibilities	P-BioSPRE is a metabiobank that provides researchers the possibility to search for and request biomaterial that fits their research purposes.
Collaborators	p-Biobank Wrapper, ObTiMA Trial Biomaterial Manager
Rationale	A metabiobank is needed that provides a harmonized search interface to biomaterial data.
Issues and notes	Technically p-BioSPRE will base on the CRIP metabiobank. It will be integrated into the p-medicine Portal

Component Name	p-Biobank Wrapper
Responsibilities	Enables a biobank owner to share his biomaterial and related data from legacy biobank management systems in p-BioSPRE within an open or closed research community.
Collaborators	p-BioSPRE
Rationale	A tool is needed that supports a biobank owner to share his biomaterial data.
Issues and notes	Technically a p-Biobank Wrapper is based on the Integrative Research Database from the CRIP toolbox. It is a local server that is installed at the site of a biomaterial owner

Component Name	ObTiMA Trial Biomaterial Manager
Responsibilities	Enables management of biomaterial data in clinical trials and sharing of selected biomaterial data in p-BioSPRE.
Collaborators	p-BioSPRE
Rationale	An ObTiMA component is needed that supports management of biomaterial data within a running trial and uploading of relevant data to p-BioSPRE.
Issues and notes	The trial biomaterial manager is developed as a component of the web based trial management system ObTiMA.

The initial architecture of the biobank access framework is shown in Figure 39 and explained in detail in D10.1.

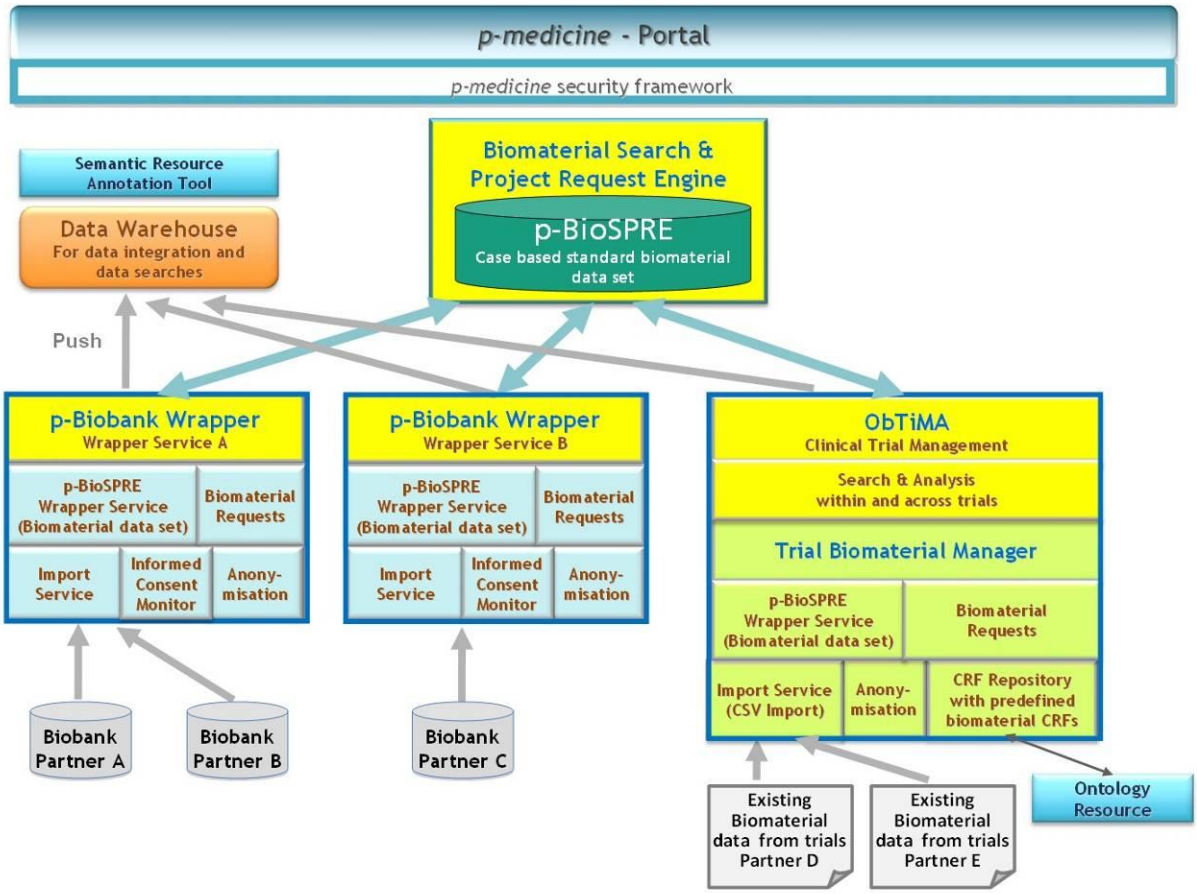


Figure 39 Initial Architecture of the Biobank Access Framework

3.3.6.3 Information View

The Information view of the system defines the structure of the system's stored and transient information (e.g. databases and message schemas) and how related aspects such as information ownership, flow, currency, latency and retention will be addressed.

In the p-medicine platform the main component responsible for storing, querying, and retrieving data is the Data Warehouse.

3.3.6.3.1 Data structure

Three main types of data types will be stored in p-medicine as shown also in Figure 40.

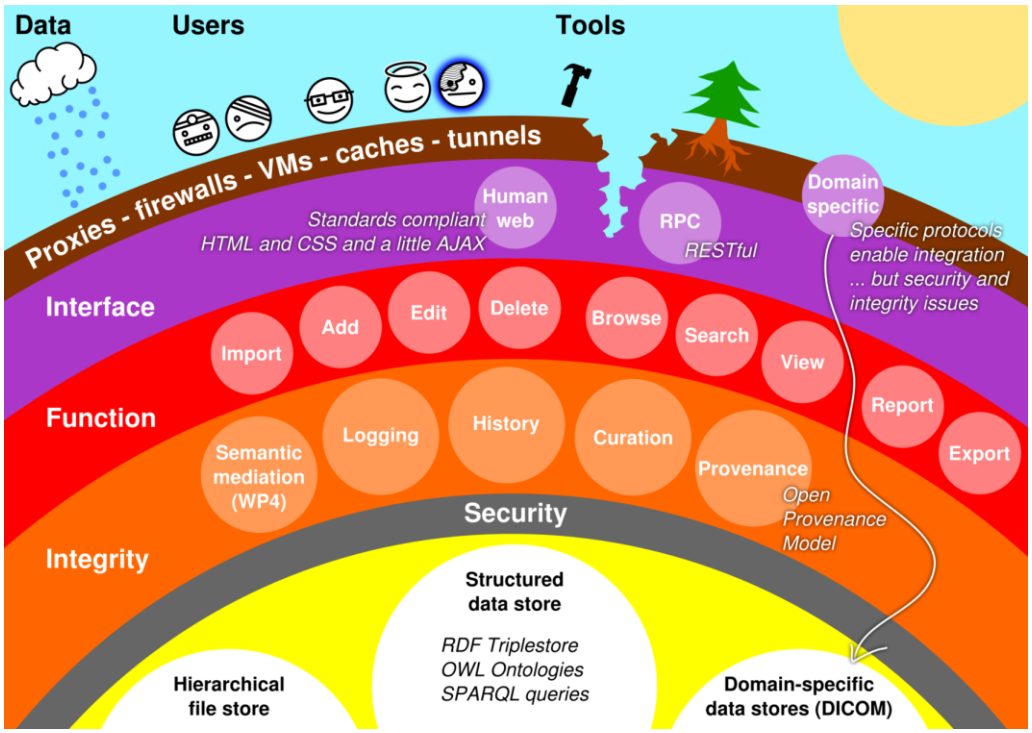


Figure 40. A layered view of the p-medicine warehouse

Binary Files: Generic files will be stored in the data warehouse. Files in the store will be referred to by URI, since many federated file stores may exist and they may be referred to by the structured data in other data warehouses. There is no specific need to keep any information about the file beyond its name, and the content of the file. File metadata and relationships between files (resembling a hierarchy) will be stored in the structures data store.

DICOM Images: DICOM (Digital Imaging and Communications in Medicine) is the de facto standard for handling, storing, printing, and transmitting information in medical imaging. images must be referred to by URI, since many federated image stores may exist. The image store should offer direct, secure access to images through the standard DICOM image access protocols.

Structured Data: The core data that will be stored in the warehouse is structured data. Since ontologies play a key part in the p-medicine project, the structured data should be transformed into the HDOT format to be saved at the data warehouse. Structured data should be saved using RDF representation and the corresponding data store should be a compatible triple-store system such as OWLIM for example.

3.3.6.3.2 Data flow

The data flow, as a consequence of the functional view of the data flow use-cases is shown on Figure 41.

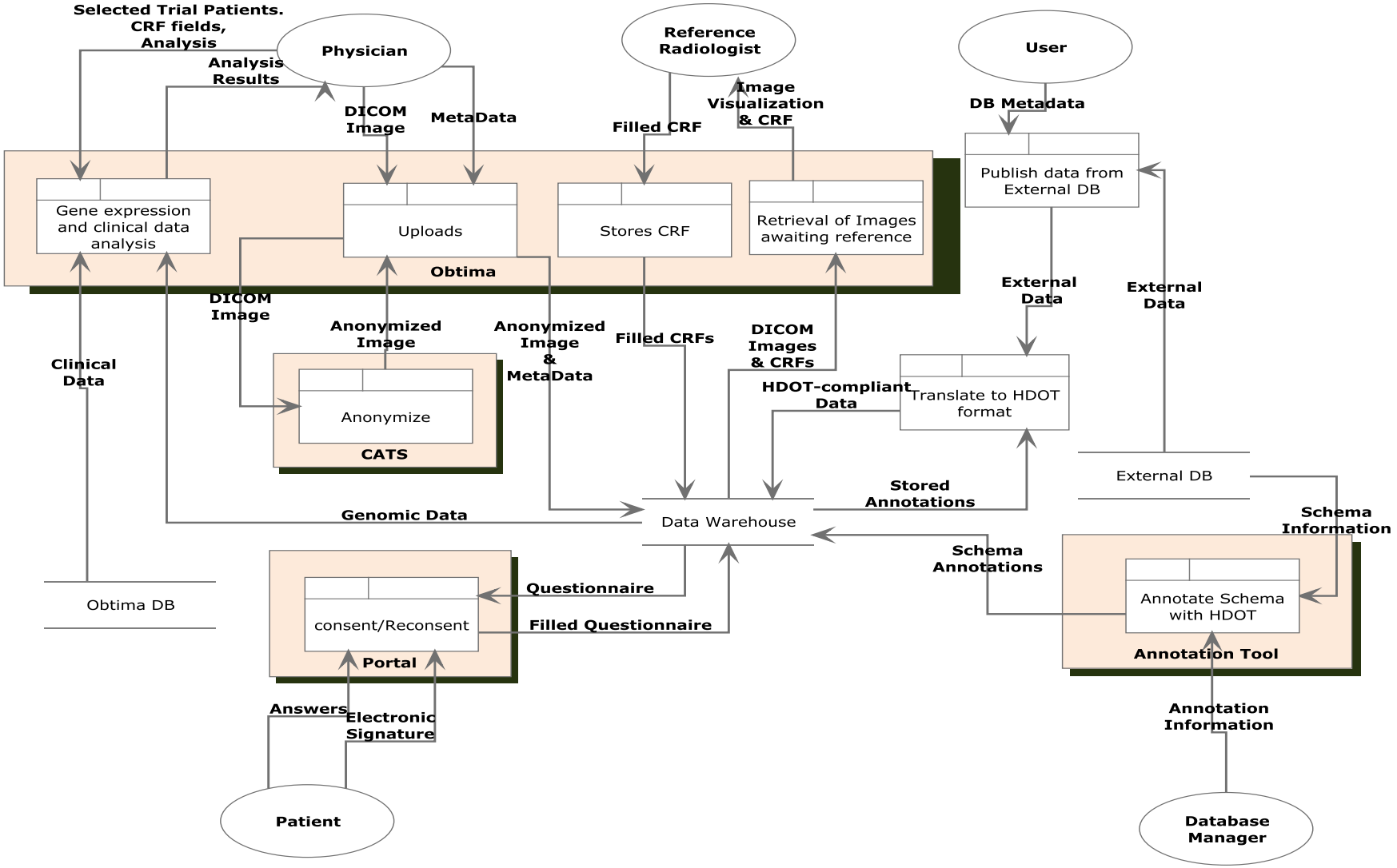


Figure 41. Data flow Diagram

We will now describe in detail the top level of the processes shown on Figure 41.

Obtima Processes

Process Name	Gene Expression and Clinical Data Analysis
Input	Clinical Data (Obtima DB), Genomic Data(DW), Parameters (Physician)
Output	Analysis Results (Physician)
Rationale	This process takes as input Clinical Data stored in Obtima CRFs and Genomic Data stored in DW and performs an analysis based on the parameters passed by the Physician

Process Name	Uploads
Input	DICOM Image (Physician), Meta-Data (Physician),
Output	Anonymized Image & Meta-Data (DW)
Rationale	This process takes as input the DICOM Image and some parameters passed by the physicians and returns the anonymized Image to be stored at the DW. Of course this Process has to call another process from CATS in order to anonymize the image.

Process Name	Retrieval of Images Awaiting Reference
Input	Radiologist id (Reference Radiologist)
Output	Visualized Images & CRFs
Rationale	This process retrieves the Radiologist id and retrieves the images awaiting reference. In order to be able to retrieve the relevant images the radiologist should have been authenticated first. However, we will not focus on security on this view and we will omit security processes involved. After the images have been retrieved the proper CRFs are presented to the user as well to be filled.

Process Name	Stores CRF
Input	Filled CRF (Reference Radiologist)
Output	Storage Notification (Reference Radiologist)
Rationale	As soon as the radiologist views the images, he can fill-in the proper fields in the CRFs. He submits the forms and they are stored in the

	DW.

Annotation Tool Processes

Process Name	Annotate Schema with HDOT
Input	Schema Information (External DB), Annotation Information (DB Manager)
Output	Schema Annotations
Rationale	This process retrieves, the schema information from external DBs, and the annotation information provided by the DB Manager and stores the annotation in the DW. This annotation will be used later by the data translation service to translate data to an HDOT-compliant format.

Portal Processes

Process Name	Consent/Reconsent
Input	Questionnaire (DW), Answers (Patient), Electronic Signature(Patient)
Output	Filled Questionnaire (DW)
Rationale	This process gets as input the questionnaire provided to the user, the corresponding answers and his electronic signature and stores the answers to the DW.

CATS Processes

Process Name	Anonymize
Input	DICOM Image (Obtima)
Output	Anonymized DICOM Image (Obtima)
Rationale	This process gets as input a DICOM image from Obtima and returns an Anonymized DICOM Image to be stored in the DW

Other Processes

Process Name	Publish Data from External DB
Input	DB MetaData (User), External Data (External DB)
Output	External Data (HDOT Translation Process)

Rationale	This process gets as input the connection parameters to an external DB provided by a user, loads the data from the specific DB and sends them to the HDOT translation process to be translated into HDOT format

Process Name	Translate to HDOT format
Input	Stored annotations (DW), External Data (External DB)
Output	HDOT-Compliant Data (DW)
Rationale	This process gets data from an external DB, and the corresponding annotations and transforms them into HDOT-compliant format to be stored at the DW.

3.3.6.3.3 Data ownership

The legal and ethical requirements of the project impose several restrictions on hoe the data are managed, shared, and maintained. First of all, the data should be (pseudo)anonymized prior to their upload as described in paragraph 3.3.6.2.2.1.4. After the upload, the anonymized data can be shared and read by the p-medicine users (physicians, bioinformaticians, clinical trial managers, etc). In any case no one can get access or even links to the initial non-anonymized dataset that was initially used. That is also the case for the user who initially uploaded the data! The legal framework allows the reidentification of the patient data in special extreme cases (e.g. when the patient should be notified for some important finding that relates to her health status), and this requires going through the Trusted Third Party (TTP) that is sole owner of the mappings between the pseudonyms and the real patient identities. The details of this process are to be further defined in the context of work package 5.

3.3.6.4 Concurrency View

The Concurrency view of the system defines the set of runtime system elements (such as operating system processes) into which the system's functional elements are packaged. The p-medicine platform is a distributed set of components accessed over the network so at first sight there's a lot of concurrency but in the inter-component communication so there's not much to propose for this view.

On the other hand the race conditions that are potentially introduced when the data in the warehouse are accessed and modified are an issue. The Data Warehouse should therefore provide all the internals mechanisms for eliminating those race conditions at the database level by adopting an appropriate modeling and update mechanisms. For example, instead of modifying existing data all new uploads will create additional *versions* of the data. This is similar to "Multiversion Concurrency Control" (MVCC) used by database management systems or the software transactional memory in programming environments to increase the concurrency of the underlying system.

3.3.6.5 Deployment View

The p-medicine platform will be distributed along many computational nodes due to its complexity, functionality, and heterogeneity of components. An initial deployment diagram is shown below:

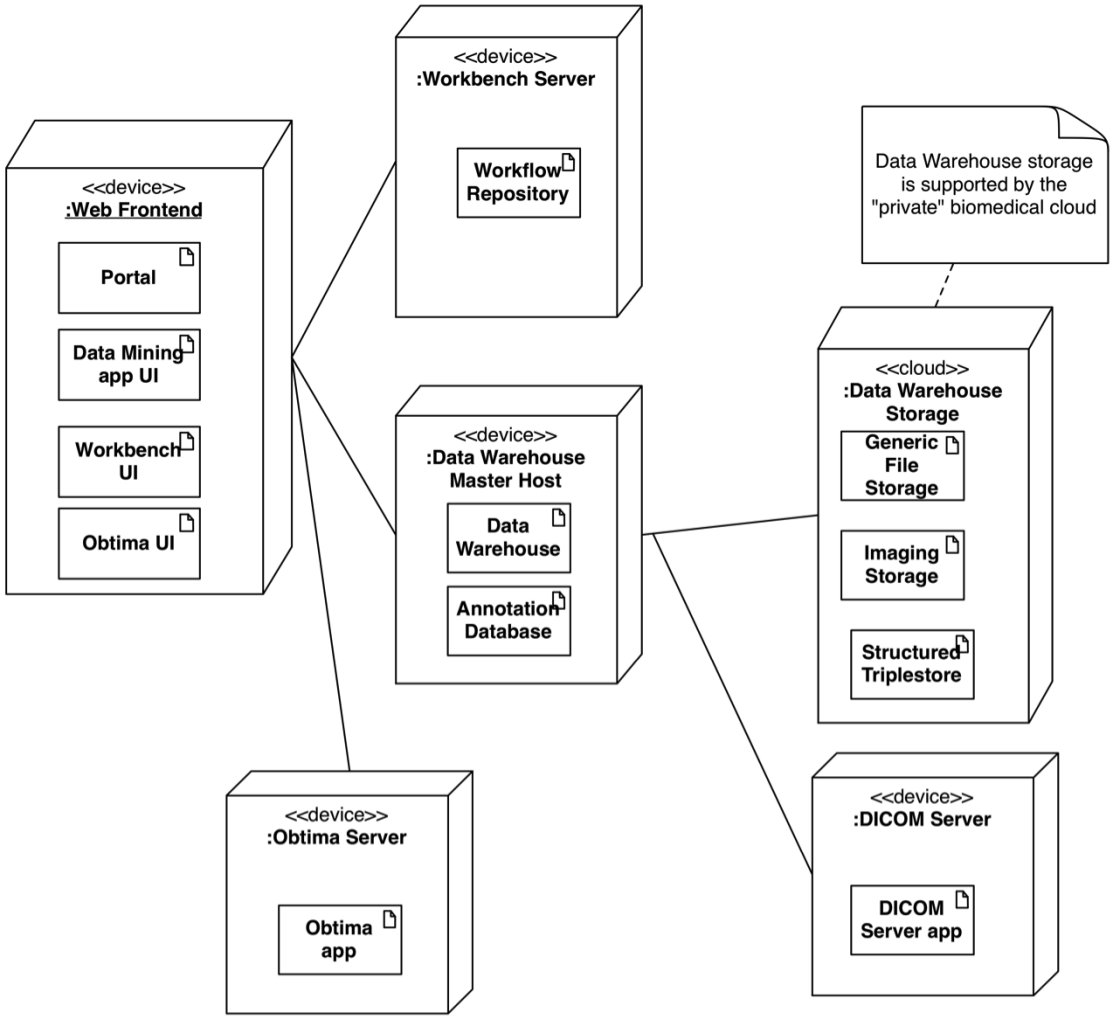


Figure 42 An initial deployment diagram for the system

The data warehouse will be designed to use a private cloud for storing:

- Generic files, i.e. “unstructured” (binary) data from the data warehouse point of view
- DICOM Images

The deployment aspects of this cloud infrastructure are described next.

3.3.6.5.1 Cloud infrastructure

P-medicine Cloud Storage System is the lowest level component in the data management architecture of p-medicine. It provides REST interfaces for managing file storing in the cloud environment and is built based on OpenStack technology. It provides access to reliable storage space taking into account requirements from different end user scenarios: long term data preservation on the one hand, as well as fast access to application data in the workflow execution.

It can use local disks or production level storage system to achieve higher level of availability and reliability for the most important data.

Based on the application profile data can be treated in a different way in a context of storage device used, or replication strategy applied.

Cloud storage in p-medicine environment is intended to be used by Data Warehouse as a storage backend for files. The other scenarios are related to data mining workflows that can

use cloud storage for intermediate computation results, and oncosimulator application executed in a dedicated cluster environment.

P-medicine project it was decided to use open source cloud storage technology OpenStack Object Storage (Swift). According to the official deployment guidelines Swift is designed to run on commodity hardware what allows to set up efficient cloud storage service at reasonable price. The primary deployment plan assumes to instantiate project wide storage service using hardware and software infrastructure provided by PSNC. PSNC has access to the highly efficient and available storage services provided by National Data Storage (NDS) which is part of the EUDAT (European Data Infrastructre).

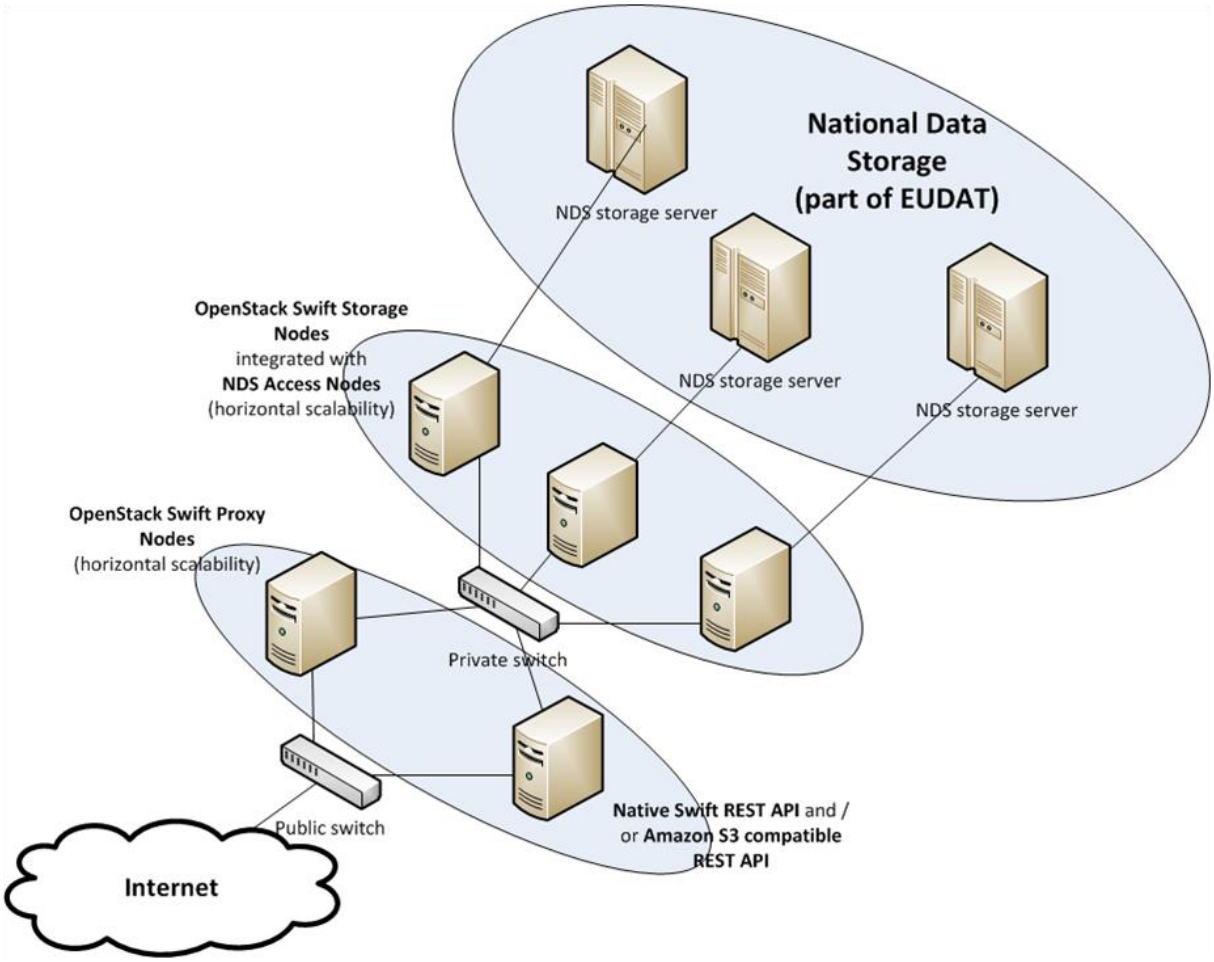


Figure 43 The deployment of the p-medicine's private cloud

This way we can provide stable and production ready resources to support project needs regarding medical data preservation. The swift services are highly autonomous so the whole architecture is flexible enough to allow different deployment scenarios. The four main services are: Proxy Services, Object Services, Container Services and Account Services. Proxy Service plays role of the contact point (API) with users and 3rd party services, while other three kind of services manage files, containers and accounts so are used to manage physical data and logical structure.

At the beginning it is assumed to have only one Proxy Service node which should be enough to carry on the initial load. The horizontal scalability of all Swift services allows to add another instances and balance load so later we will be able to extend overall API throughput. All other services will be deployed on three different nodes and create a balance and replication ring to allow suitable level of redundancy to secure all services' operational databases.

We assume to have a cyclic deployment plan starting from initial infrastructure. From the beginning all resources will be monitored and in case of higher resource usage another nodes with additional services will be added. The whole p-medicine architecture is still under definition and will be developed continuously so cyclic deployment strategy will fit the overall direction of project development.

3.3.6.6 Development view

The Development view of the system defines any constraints on the software development process that are required by the architecture. This includes the system's module organisation, common processing that all modules must implement, any required standardisation of design, coding and testing and the organisation of the system's code base.

Our opinion is that in this stage of the P-Medicine project, where the architecture is still under definition, the definition in detail of the development processes, technologies and constraints is premature. After the definition of the platform's functionalities and context properties, and taking into account the interactions between the various components, it is appropriate first to define the interfaces of these interactions and based on these interfaces to conclude in specific development decisions.

Nevertheless, in the context of Task T3.1 (monitoring of standards) and the deliverable D3.1, some development decisions have been taken or at least seem to be the most appropriate, based on the evaluations of the various standards and the best practice techniques. This documentation can be found in the Deliverable 3.1 in full extent, and we copy here only a selected set of guidelines, technologies and architectural styles that seem to be the best choices amongst the various alternatives. However, we stress out the fact that we do not exclude the usage of any technology, we only encourage the usage of specific technologies for easier integration and interoperability reasons as preferable, whenever the ability to select the development technology is given.

3.3.6.6.1 Open standards and technologies

Due to the distributed nature of the platform, which is composed by many different tools and services, the most logical choice for the development of the platform is by using open standards and open technologies. This way, the platform can be able to easily adopt externally provided solutions and interoperate with other projects, organizations, data providers and end users. Examples of such proposed open technologies are:

- The usage of HTML5 for the development of web interfaces, instead of proprietary techniques and tools.
- The usage of XML for data exchange instead of proprietary or non-standardized data formats.
- The usage of HTTP as the transfer protocol.
- The usage of LAMP/LAPP (Linux, Apache, MySQL/PostgreSQL, Perl/PHP/Python) solutions instead of proprietary server solutions.

3.3.6.7 Operational View

The Operational view defines how the system will be installed into its production environment and how it will be configured, managed, monitored, controlled and maintained.

The details of this view will be provided as the development of the platform moves forward.

System Qualities

Non-functional requirements are usually orthogonal to functionality and are observable properties of the system. They are usually "systemic" in the sense that there's no a single

place that has the responsibility for each of them. Instead these non-functional requirements or quality properties of the system emerge from the architecture and the design.

3.3.6.8 Performance and scalability

The requirements regarding such non-functional system qualities have not yet been specified. In subsequent versions of the architecture we will focus on defining the main such performance requirements and the architectural decisionsenabling the p-medicine technical platform to meet those requirements.

3.3.6.9 Security

The security requirements and how they are addressed are shown in the next table.

Requirement	How Met
Authentication	Single Sign On and Single Sign Out
Authorization	Non-critical access requests, where no patient data is involved, can be authorised locally (i.e. by the portal, obtima, etc.) by using identity information retrieved from the identity provider (through the authentication token), possibly extended with locally stored user information. On the other hand, critical access control decisions, where patient data is involved, will be taken centrally by the p-medicine policy decision point.

3.3.6.10 Usability

Usability plays an essential role in the whole development process of the project p-medicine. The main objective of the usability methodology in the beginning of a project is to describe the task with the whole context of use of the end users. To assure that the software used in p-medicine will meet the high demands of the end users and that the platform fulfils the requirements for usability of the main target groups, the software has to be evaluated by the users throughout the development period. Taking user needs into account early in the project development can reduce implementation costs and avoid loss of time.

The usability process we will use is described in the Deliverable 2.1 and it is based on the interviews taken with the representative of each group of users (e.g. clinicians, trial managers, bioinformaticians, etc.). The interviews were documented in five context scenarios in Appendix 2 of Deliverable 2.2 that have been sent first to the interviewees themselves for validation before the usability engineer derives the system requirements. Achieving a common understanding of the requirements is indeed a necessary step to enable the developer of a platform supporting efficient user activities, and user satisfaction. The next step is to consolidate the implementation of the software tools in accordance to the requirement specification defined by the context scenarios. After the first prototypes were implemented real prospective users will have the opportunity to test the software. The first prototypes need not to have the complete functionality of the tool. It should give the user a first view of the interface and what is possible. The resulting use scenarios are documented and will be described in detail in WP 15.

4 Conclusion

In this document we have started the process of documenting the p-medicine's architecture. We have made a selection for the design process and proceeded to the identification of the major views etc.

We have followed a more or less strict approach here in compliance with the terminology of the IEEE standard 1471 and some of the current best practices. Nevertheless we have not delved too much deep in the details and the meticulous specification of all the possible aspects and characteristics of the p-medicine architecture. The reason is twofold. On one hand we are just initiating the process and we expect that we revisit and enhance this document in the course of the project. On the other hand we are very fond of the "Big Up Front Design". Instead we plan to follow a more *agile* software development process. The Agile Manifesto values the efficient delivery and change in the software by focusing on the continuous communication with the stakeholders, the iterative design, and the frequent release cycle.

Such an incremental and iterative approach is the one proposed by the "Twin Peaks" model of Nuseibeh [9] shown in Figure 44. Using the author's own words "*the spiral life-cycle model addresses many drawbacks of a waterfall model by providing an incremental development process, in which developers repeatedly evaluate changing project risks to manage unstable requirements and funding*".

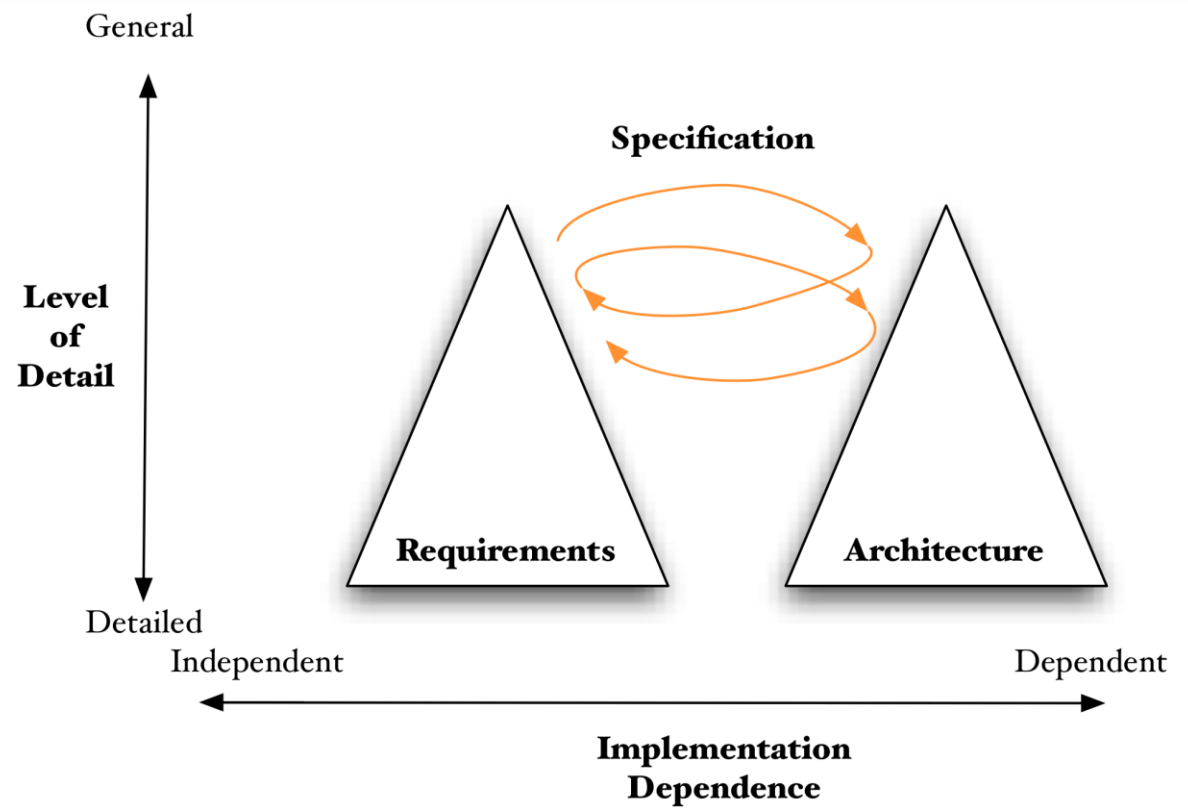


Figure 44 Architecture definition context

This interplay between requirements and architecture is justified by the following observations:

- Requirements analysis provides the context for architecture definition by defining the scope and the system's desired functionality and quality properties.

- Architecture definition often reveals inconsistent and missing requirements and also helps stakeholders understand the relative costs and complexities of meeting their concerns. This feeds back into requirements analysis to clarify and add requirements and to prioritize these when tradeoffs are made between stakeholders' aspirations and what can be achieved given time and budget constraints.

Therefore this spiral type of architecture definition is the approach we aim to follow in p-medicine. This document has presented the initial requirements and how these are mapped into architectural decisions and subsequent versions will elaborate more on the specifics of each architectural view.

5 References

[1]. Bass L., Clements P. and Kazman R., Software Architecture in Practice, Addison-Wesley, 1997.
[2]. Malan Ruth, Bredemeyer Dana: Software Architecture Action Guide, Bredemeyer Consulting, 2004
[3]. Clements, Paul, et al. Documenting Software Architectures. Boston, MA: Addison-Wesley, 2003.
[4]. Deliverable 2.2 – Definition on scenarios and use cases and report on scenario based user needs and requirements, p-medicine consortium
[5]. Deliverable 11.1 – Initial definition of data mining patterns, p-medicine consortium
[6]. Rozanski, Nick, and Woods, Eoin. Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives, 2nd Edition. Addison Wesley, 2011.
[7]. IEEE Computer Society. Recommended Practice for Architectural Description. IEEE Std-1471-2000. October 9, 2000. http://standards.ieee.org/reading/ieee/std_public/description/se/1471-2000_desc.html
[8]. Kruchten, Philippe. "Architectural Blueprints - The 4+1 View Model of Software Architecture." IEEE Software, 12(6):42–50, November 1995.
[9]. Nuseibeh, Bashar. "Weaving Together Requirements and Architectures." IEEE Computer, 34(3): 115–117, March 2001.
[10]. George H. Fairbanks, Just Enough Software Architecture: A Risk-Driven Approach Marshall & Brainerd. 2010

6 Glossary

- The **actor** is an active entity (human user or external system) that is in the environment of the system and that interacts with the system. An actor represents a coherent set of roles – one user can perform several roles and several users can play the same role.
- The ***architecture*** of a system defines four different aspects: its *static structure,* its *dynamic structure,* its *externally visible behavior,* and its *quality properties.*
- **Architectural overview document**: An architectural document giving an overview of the architecture at a high level of abstraction. The document is targeted at a broad range of audiences including developers, marketing, management and possibly potential end-users.
- **Architectural style**: Defines a family of systems in terms of a pattern of structural organization. Thus it is a set of rules, which determines a set of components and the manner in which they should be connected together. A style or pattern describes a generic solution to a specific class of problems appearing typically in a specific context.
- **Architecture pattern**: see architectural style
- **Component**: A unit of responsibility and functionality on a specific abstraction level. A component may correspond to a single class or a group of implementation classes. Components may merely serve as a high-level grouping mechanism for classes and not be reflected in the actual code (white-box component). Or components may be encapsulations of classes having an interface or façade class that is part of the component and hides the internal structure of the component (black-box component). Such interface or façade classes often have the name of the component they belong to. Components can be passive or active (have their own thread of control), be created at system startup or be created and deleted any time at runtime, be singletons or have several instances, and they can be system specific or be reusable library
- A ***stakeholder*** is a person, group, or entity with an interest in or concerns about the realization of the architecture. Stakeholders include users but also many people, such as developers, operators, and acquirers. Architectures are created solely to meet stakeholder needs
- A **Policy Administration Point** (PAP) is an endpoint that manages policies. It will provide a PDP with all policies required to produce an authorization decision.
- A **Policy Decision Point** (PDP) is an entity that makes authorization decisions. A PDP accepts authorization requests and will make a decision based on policies fetched from a Policy Administration Point (PAP).
- A **Policy Enforcement Point** (PEP) is a software component which requests and enforces authorization decisions.
- A **Policy Information Point** (PIP) is an endpoint that provides missing information to a Policy Decision Point (PDP), i.e. attribute information. For example, if a policy requires information on a specific attribute that has not been provided with the authorization request, a Policy Decision Point (PDP) might request a PIP for information on that attribute.

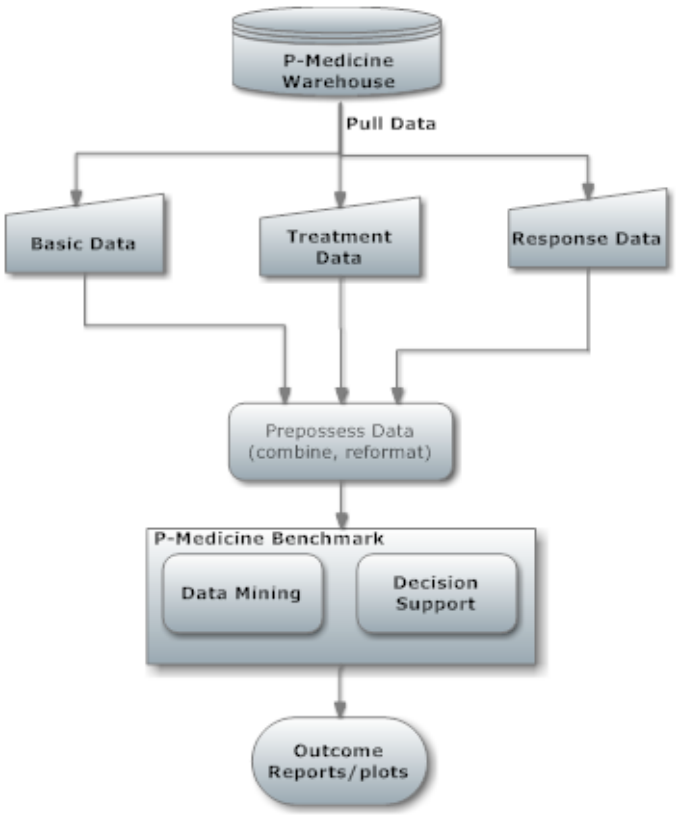
7 Appendix 1 - Abbreviations and acronyms

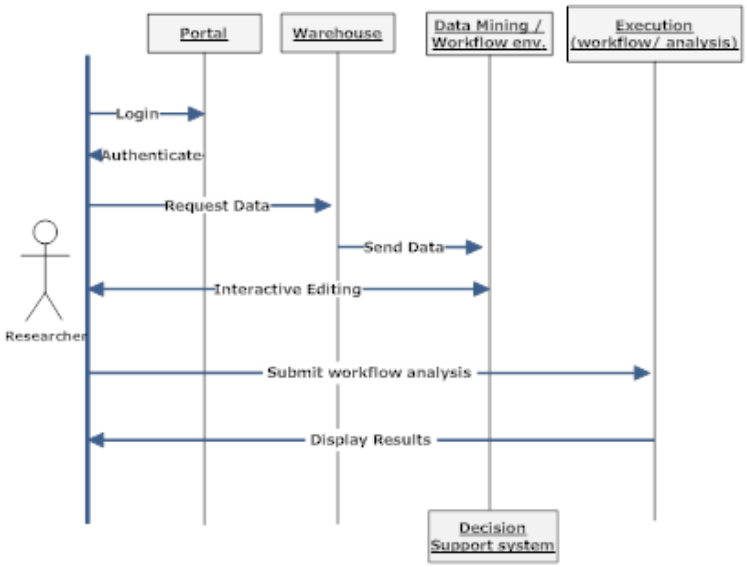
AE	Adverse Events
CAT	Custodix Anonymization Tool
CATS	CAT Service
CDS	Clinical Decision Support
DSS	Decision Support System
DICOM	Digital Imaging and Communications in Medicine
EHR	Electronic Health Record
HTTP	Hypertext Transfer Protocol
HMAC	Hash-based Message Authentication Code
IdP	Identity Provider
PAP	Policy Administration Point
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIMS	Personal Information Management System
PIP	Policy Information Point
REST	REpresentational State Transfer
SLO	Single Log Out
SSO	Single Sign On
TTP	Trusted Third Party
UI	User Interface

8 Appendix 2 – ALL and Breast Cancer Use Cases

Item	Description
Identifier	ALL_1
Version	0.1
Name	Relapse or Minimal Residual Disease on Acute Lymphoblastic Leukaemia
Description of the use case (end-user perspective)	Minimal residual disease (MRD) is the name given, to small numbers of leukaemic cells that remain in the patient during treatment or after treatment when the patient is in remission (no symptoms or signs of disease). It is the major risk factor for treatment failure or relapse leukaemia. Data of a representative cohort of 2000 patients will be used. Data should be accessed through the system, data analysis and data mining can be performed and the results are presented in a clearly structured way. In future results can be used for decision support.
Problem(s) to solve	To find indicative patterns within basic, treatment and response data that lead to relapse or minimal residual disease for Acute Lymphoblastic Leukaemia
Challenges	High variability in predicting variables
Risks	
Expected benefits	support to clinical decisions
Characterization	○ fundamental ○ general ◉ specific - (this scenario is meant for ALL data)
If specific, please give the Domain	◉ Acute lymphoblastic leukaemia ○ Breast Cancer ○ Nephroblastoma ○ other Cancer, please specify ○ Non-Cancer Domain, please specify:
End-user	○ system ◉ person ◉ basic scientist ◉ clinician ○ computer scientist ○ regulatory body, lawyer, ethicist ○ patient ○ other, please specify:
Pre-condition(s)/pre-requisite(s)	Decision support tools/libraries, such as R weka matlab, for the analysis of the data

Requisite(s)	
Post-condition(s)/post-requisite(s)	
Constraints	
External sources needed from outside p-medicine	○ data, please specify: ○ tools, please specify: ○ services, please specify: ○ models, please specify: ○ other, please specify:
Data used	◉ personal ○ only non-personal ○ target population, please specify:
Input data	◉ internal database: At P-medicine warehouse will be • Basic data: gender, age at diagnosis, white blood cell count at diagnosis, blood blast count,hemoglobin levels and platelet counts at diagnosis, FAB classification, complete immunophenotyping data, ploidy status, status for prognostic relevant chromosomal translocations (ETV6/RUNX1, BCR/ABL, MLL/AF4, E2A/PBX1), percentage of bone marrow blasts, extramedullary disease (CNS, testis, and others). • Treatment data: risk group stratification, cumulative drug doses, information on HSCT and cranial irradiation, information on time frame for the application of treatment phases. • Response data: prednisone response, blast percentages in the bone marrow on treatment days 15 and 33, MRD analyses on treatment days 33 and 78. ○ external database, please specify: ○ online input:
Output data	○ database, please specify: ○ variables for use, please specify: ◉ structured document: Predictive values and summaries that assist in clinical decision support for relapse, treatment-related mortality, secondary malignancy. ◉ graphic, please specify: plots if available from the Knowledge discovery tools
Data volume	~100 MB

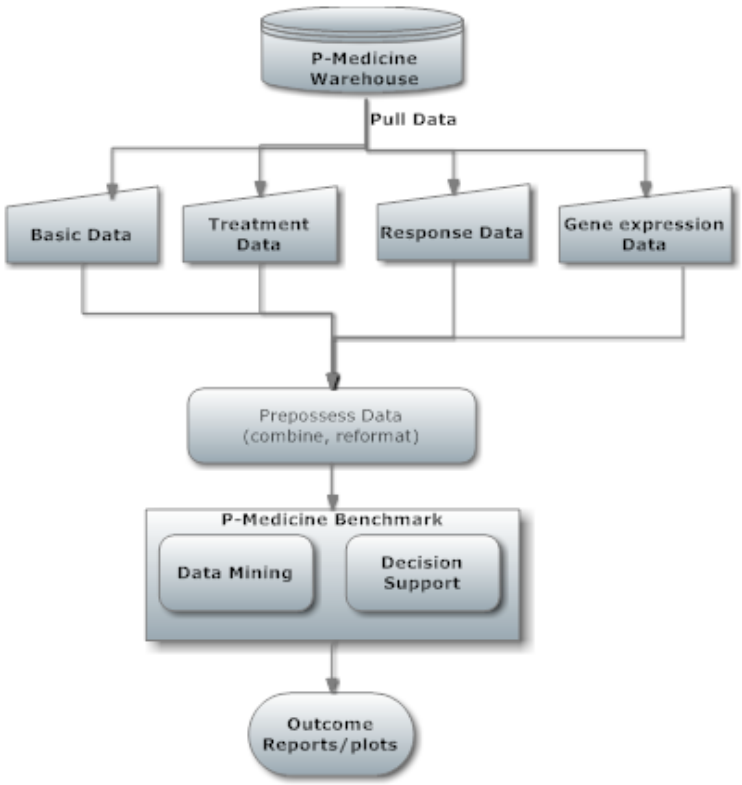
<table>
<tr><td>Dataflow</td><td colspan="2"> 
</td></tr>
<tr><td>Data storage</td><td colspan="2">All the data will be available from the p-medicine data warehouse.</td></tr>
<tr><td>Successful End Condition</td><td colspan="2">Report and plot that assist to the decision support.</td></tr>
<tr><td>Fail End Condition</td><td colspan="2">the analysis stops with error messages</td></tr>
<tr><td rowspan="4">Basic workflow*</td><td>Actor Action (Researcher)</td><td>System response</td></tr>
<tr><td>Login to portal</td><td>Authentication of the user.</td></tr>
<tr><td>Request data</td><td>Retrieve data (basic, treatment, response) data for ALL.</td></tr>
<tr><td>Create/edit Data mining workflow</td><td>Interactive GUI for the editing (workflow editing environment)</td></tr>
<tr><td></td><td>Submit workflow</td><td>Execute workflow/ Return results</td></tr>
</table>

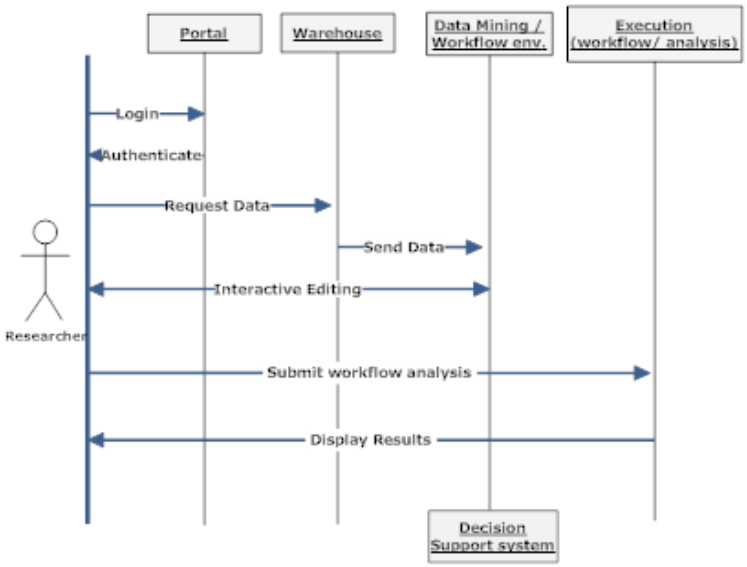
	
Expected usage frequency	Low
Needed for DSS	◉ yes ○ no
Needs HPC	○ yes ◉ no
Needs Grid	○ yes ◉ no
Priority for development	
Responsible for development	
Mock-up needed	○ yes ◉ no
Responsible for Mock-up	
Who is building the tool	WP11 &WP13 partners with CAU
Open Source tool	◉ yes ○ no, please specify why:

Item	**Description**
Identifier	ALL_2
Version	0.1
Name	Relapse or Minimal Residual Disease on Acute Lymphoblastic Leukaemia
Description of the use case (end-user perspective)	Minimal residual disease (MRD) is the name given, to small numbers of leukaemic cells that remain in the patient during treatment or after treatment when the patient is in remission (no symptoms or signs of disease). It is the major risk factor for treatment failure

	or relapse leukaemia. Data of a representative cohort of 664 patients will be used. Data should be accessed through the system, data analysis and data mining can be performed and the results are presented in a clearly structured way. In future results can be used for decision support.
Problem(s) to solve	To find indicative patterns within basic, treatment, response and gene expression data that lead to relapse or minimal residual disease for Acute Lymphoblastic Leukaemia
Challenges	High variability in predicting variables
Risks	
Expected benefits	support to clinical decisions
Characterization	○ fundamental ○ general ◉ specific - (this scenario is meant for ALL data)
If specific, please give the Domain	◉ Acute lymphoblastic leukaemia ○ Breast Cancer ○ Nephroblastoma ○ other Cancer, please specify ○ Non-Cancer Domain, please specify:
End-user	○ system ◉ person ◉ basic scientist ◉ clinician ○ computer scientist ○ regulatory body, lawyer, ethicist ○ patient ○ other, please specify:
Pre-condition(s)/pre-requisite(s)	Decision support tools/libraries, such as R weka matlab, for the analysis of the data
Requisite(s)	
Post-condition(s)/post-requisite(s)	
Constraints	
External sources needed from outside p-medicine	○ data, please specify: ○ tools, please specify: ○ services, please specify: ○ models, please specify: ○ other, please specify:

Data used	⦿ personal O only non-personal O target population, please specify:
Input data	⦿ internal database: At P-medicine warehouse will be • Basic data: gender, age at diagnosis, white blood cell count at diagnosis, blood blast count, hemoglobin levels and platelet counts at diagnosis, FAB classification, complete immunophenotyping data, ploidy status, status for prognostic relevant chromosomal translocations (ETV6/RUNX1, BCR/ABL, MLL/AF4, E2A/PBX1), percentage of bone marrow blasts, extramedullary disease (CNS, testis, and others). • Treatment data: risk group stratification, cumulative drug doses, information on HSCT and cranial irradiation, information on time frame for the application of treatment phases. • Response data: prednisone response, blast percentages in the bone marrow on treatment days 15 and 33, MRD analyses on treatment days 33 and 78. • Gene expression data: low-density array of 95 genes previously associated with treatment response and/or outcome. O external database, please specify: O online input:
Output data	O database, please specify: O variables for use, please specify: ⦿ structured document: Predictive values and summaries that assist in clinical decision support for relapse, treatment-related mortality, secondary malignancy. ⦿ graphic, please specify: plots if available from the Knowledge discovery tools
Data volume	~100 MB

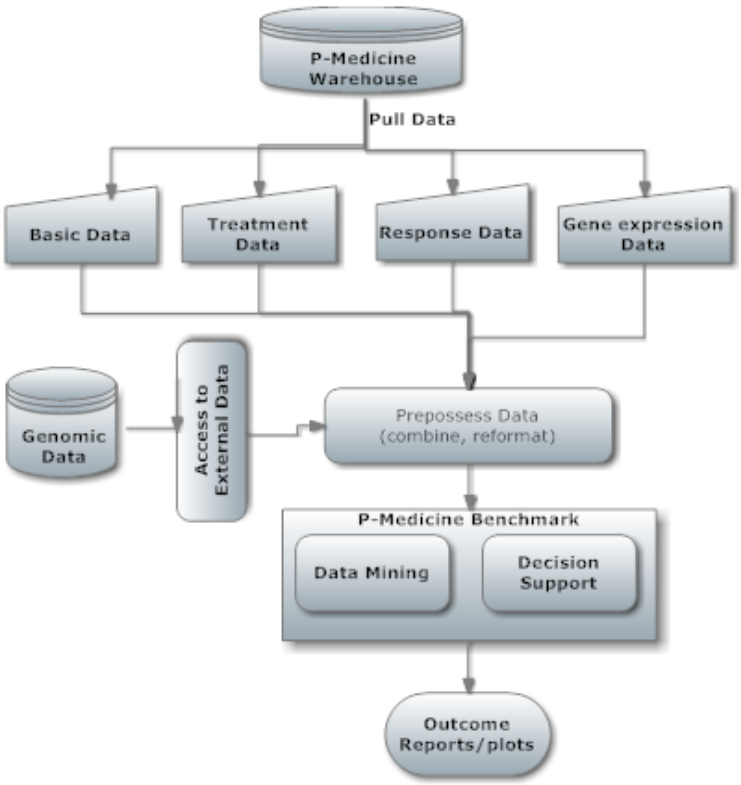
<table>
<tr><td>Dataflow</td><td> 
</td></tr>
<tr><td>Data storage</td><td>All the data will be available from the p-medicine data warehouse.</td></tr>
<tr><td>Successful End Condition</td><td>Report and plot that assist to the decision support.</td></tr>
<tr><td>Fail End Condition</td><td>the analysis stops with error messages</td></tr>
<tr><td>Basic workflow*</td><td>
<table>
<tr><td>Actor Action (Researcher)</td><td>System response</td></tr>
<tr><td>Login to portal</td><td>Authentication of the user.</td></tr>
<tr><td>Request data</td><td>Retrieve data (basic, treatment, response) data for ALL.</td></tr>
<tr><td>Create/edit Data mining workflow</td><td>Interactive GUI for the editing (workflow editing environment)</td></tr>
<tr><td>Submit workflow</td><td>Execute workflow/ Return results</td></tr>
</table>
</td></tr>
</table>

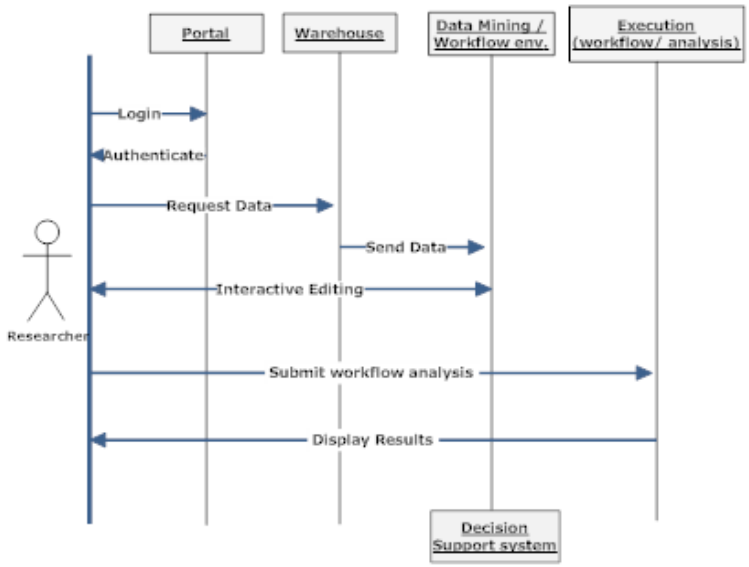
	
Expected usage frequency	Low
Needed for DSS	⦿ yes O no
Needs HPC	O yes ⦿ no
Needs Grid	O yes ⦿ no
Priority for development	
Responsible for development	
Mock-up needed	O yes ⦿ no
Responsible for Mock-up	
Who is building the tool	WP11 &WP13 partners with CAU
Open Source tool	⦿ yes O no, please specify why:

Item	**Description**
Identifier	ALL_3
Version	0.1
Name	very high risk leukaemia
Description of the use case (end-user perspective)	Data of a representative cohort of 100 patients will be used divided into two categories. Case control of 50 VHRL (very high risk leukaemia) and 50 non VHRL patients. Data should be accessed through the system, data analysis and data mining can be performed and the

	results are presented in a clearly structured way. In future results can be used for decision support.
Problem(s) to solve	To find indicative patterns within basic, treatment, response, gene expression and genomic data that can discriminate the VHRL and non VHRL patients.
Challenges	High variability in predicting variables
Risks	
Expected benefits	support to clinical decisions
Characterization	○ fundamental ○ general ◉ specific - (this scenario is meant for ALL data)
If specific, please give the Domain	◉ Acute lymphoblastic leukaemia ○ Breast Cancer ○ Nephroblastoma ○ other Cancer, please specify ○ Non-Cancer Domain, please specify:
End-user	○ system ◉ person ◉ basic scientist ◉ clinician ○ computer scientist ○ regulatory body, lawyer, ethicist ○ patient ○ other, please specify:
Pre-condition(s)/pre-requisite(s)	Decision support tools/libraries, such as R weka matlab, for the analysis of the data
Requisite(s)	
Post-condition(s)/post-requisite(s)	
Constraints	
External sources needed from outside p-medicine	○ data, please specify: ○ tools, please specify: ○ services, please specify: ○ models, please specify: ○ other, please specify:
Data used	◉ personal ○ only non-personal ○ target population, please specify:
Input data	◉ internal database: At P-medicine warehouse will be

	• Basic data: gender, age at diagnosis, white blood cell count at diagnosis, blood blast count, hemoglobin levels and platelet counts at diagnosis, FAB classification, complete immunophenotyping data, ploidy status, status for prognostic relevant chromosomal translocations (ETV6/RUNX1, BCR/ABL, MLL/AF4, E2A/PBX1), percentage of bone marrow blasts, extramedullary disease (CNS, testis, and others). • Treatment data: risk group stratification, cumulative drug doses, information on HSCT and cranial irradiation, information on time frame for the application of treatment phases. • Response data: prednisone response, blast percentages in the bone marrow on treatment days 15 and 33, MRD analyses on treatment days 33 and 78. • Gene expression data: low-density array of 95 genes previously associated with treatment response and/or outcome. O external database: • Genomic data: high density gene expression data (40000 datapoints), SNP array data mainly Affimetrix 6.0, genomwide information on CNV/LOH (Copy Number Variation, Loss Of Heterozygozity) O online input:
Output data	O database, please specify: O variables for use, please specify: ⊙ structured document: Predictive values and summaries that assist in clinical decision support for relapse, treatment-related mortality, secondary malignancy. ⊙ graphic, please specify: plots if available from the Knowledge discovery tools
Data volume	~100Mb

<table>
<tr><td>Dataflow</td><td colspan="2"> 
</td></tr>
<tr><td>Data storage</td><td colspan="2">Most of the data will be available from the p-medicine data warehouse. Also external data (genomic data) are required for this scenario.</td></tr>
<tr><td>Successful End Condition</td><td colspan="2">Report and plot that assist to the decision support.</td></tr>
<tr><td>Fail End Condition</td><td colspan="2">the analysis stops with error messages</td></tr>
<tr><td rowspan="4">Basic workflow*</td><td>Actor Action (Researcher)</td><td>System response</td></tr>
<tr><td>Login to portal</td><td>Authentication of the user.</td></tr>
<tr><td>Request data</td><td>Retrieve data (basic, treatment, response) data for ALL.</td></tr>
<tr><td>Create/edit Data mining workflow</td><td>Interactive GUI for the editing (workflow editing environment)</td></tr>
<tr><td></td><td>Submit workflow</td><td>Execute workflow/ Return results</td></tr>
</table>

	
Expected usage frequency	Low
Needed for DSS	◉ yes ○ no
Needs HPC	○ yes ◉ no
Needs Grid	○ yes ◉ no
Priority for development	
Responsible for development	
Mock-up needed	○ yes ◉ no
Responsible for Mock-up	
Who is building the tool	WP11 &WP13 partners with CAU
Open Source tool	◉ yes ○ no, please specify why:

Item	Description
Identifier*	PSB_1
Version	1.0
Name	**Pathway Scenario for Breast Cancer**
Description of the use case (enduser perspective)	Input: Immunohistochemistry (IHC), gene expression and clinical data from breast cancer Action:

	These data needs to be: 1) Processed and QC checked (some manual steps and some automated steps) 2) analyzed for associations between gene expression, IHC and clinical data using statistical tools (e.g. R) 3) finally, correlated to pathway data using for example the KEEG pathway database (http://www.genome.jp/kegg/pathway.html), MetaCore™ (http://www.genego.com/trial) or Ingenuity (http://www.ingenuity.com/). Output: Specific cohorts of patients with breast cancer will be produced as a result.
Problem(s) to solve	To find disrupted pathways in breast cancer
Challenges	Smooth link of databases. To make the tool domain independent for usage in other cancer domains.
Risks	Incorrect match of databases can generate wrong hypotheses which can be extremely costly.
Expected benefits	Hypotheses generated are fed back to biologist to plan validation studies and clinicians to plan trials and new clinical studies.
Characterization	O fundamental ⊙ general ⊙ specific
If specific, please give the Domain	O Acute lymphoblastic leukaemia ⊙ Breast Cancer O Nephroblastoma ⊙ other Cancer, please specify: this scenario could be extended to most solid cancers O Non-Cancer Domain, please specify:
Enduser	⊙system (this could be part of a larger scenario) ⊙person ⊙ basic scientist ⊙ clinician O computer scientist O regulatory body, lawyer, ethicist O patient ⊙ other, please specify: biostatistician, epidemiologist, bioinformatician

Pre-condition(s)/pre-requisite(s)	Availability of gene expression, immunohistocemistry and clinical data, availability of pathway databases. Anonymization of personal data is needed (although data are pre-anonymized by UOXF).
Requisite(s)	If used as clinical decision support service (DSS)
Post-condition(s)/post-requisite(s)	If used as DSS the result in individual patients needs to be delivered on time.
Constraints	If used as DSS the data from gene expression analysis, their normalisation, as well as the clinical data needs to be available on time. These logistics have to be solved otherwise (if data are coming late) the patient will not benefit from this use case as a DSS. This risk is independent of the IT and the data generation (laboratory work for gene expression and immunohistochemistry).
External sources needed from outside p-medicine	⊙data, please specify: KEGG or other pathway database ⊙ tools, please specify: R or other statistical tool to perform analysis of samples ⊙ services, please specify: Access to local clinical and pathology database O models, please specify: O other, please specify:
Data used	⊙ personal O only non-personal ⊙ target population, please specify: Retrospective series of breast cancer patients treated in Oxford
Input data	⊙internal database, please specify: a) clinical database: description: The clinical data will be provided by ObTiMA b) gene array expression data: The gene array data will be provided as CEL files. They need to be further specified. The data need to be normalized. ⊙ external database, please specify: Pathology database

	KEGG database ⊙ online input Selection of patients from the clinical database Selection of variables to correlate	
Output data	⊙ database, please specify: Results should be stored in database O variables for use, please specify: ⊙structured document, please specify: Document content should be: Results from the association analysis Results from the pathway analysis (Tables with all statistics) ⊙graphic, please specify: Heatmap of gene expression data Scatter plots, Box plots Kaplan-Meier curves	
Data volume	Large, depending on the number of cases and the number of genes analysed in the gene array experiments	
Dataflow	Please specify: The data flow needs to be specified during the development of the tool. Data should be stored in the data warehouse.	
Data storage	Please specify: Data will be stored in the data warehouse after anonymization	
Successful End Condition	Delivering disrupted pathways in breast cancer for a single patient or a cohort of patients	
Fail End Condition		
Basic workflow	Actor Action	System response
	Selection of databases	View of databases and/or variables
	Check QC for all data	QC visualized – ok from the user needed to proceed
	Selection of cases	Only these data are used in the scenario

	Check specific QC of cases	QC visualized – ok from the user needed to proceed
	Selection of variables	Only these variables are used
		The system automatically matches the clinical, IHC and gene expression data
		The workflow for analysis is defined previously by the Tool builder
		The analysis is run and significant association are flagged In case of single patient: disrupted pathways are shown, with list of relevant genes In case of cohort: all analyses results are shown, with relevant plots
		Daily for historical cohort studies at present. Eventually prospectically for new trials. For all breast cancer patients where gene expression and IHC data are present.
	Download results	
Expected usage frequency		
Needed for DSS	⊙ yes O no	
Needs HPC	⊙yes O no	
Needs Grid	O yes ⊙ no (not strictly necessary in this configuration, but yes if gene expression data are replaced by sequencing data)	
Priority for development	high	

Responsible for development	technical group
Mockup needed	⊙ yes O no
Responsible for Mockup	technical group
Who is building the tool	technical group
Open Source tool	⊙ yes O no, please specify why: