MyHealthAvatar

A Demonstration of 4D Digital Avatar Infrastructure for Access of Complete Patient Information

Project acronym: MyHealthAvatar

Deliverable No. D3.2
Architecture Design

Grant agreement no: 600929

SEVENTH FRAMEWORK PROGRAMME

Dissemination Level		
PU	Public	**X**
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)	

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	MyHealthAvatar
Project Full Name:	A Demonstration of 4D Digital Avatar Infrastructure for Access of Complete Patient Information
Deliverable No.:	D3.1
Document name:	User Requirements
Nature (R, P, D, O)[1]	R
Dissemination Level (PU, PP, RE, CO)[2]	PU
Version:	1.0
Actual Submission Date:	30/06/2013
Editor: Institution: E-Mail:	Manolis Tsiknakis TEI-C tsiknaki@epp.teicrete.gr

ABSTRACT:

This deliverable will describe MyHealthAvatar architectural blueprint and implementation activities. The emphasis, in order to meet project objectives, for the definition and design of MyHealthAvatar's architectural scheme is on the selected standards and the technologies user and functional requirements. We followed the IEEE 1471 recommendation that 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. The steps were on capturing stakeholder needs; making a series of architectural design decisions that resulted in a solution that meet these needs, assessing it against the stakeholder needs, and refining the solution until it is adequate and captures the architectural design decisions in a complete *"Architectural Description"*. These iterative activities formed the core of the architecture definition process. The aim of this deliverable is to feed the subsequent tasks in the building of the system.

KEYWORD LIST:

User Requirements, Standards, Guidelines, Protocols, Formats, IT, State-Of-the-Art, Review

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n^o 600929.

[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**
1.0	09/13/2013	Draft	Manolis Tsiknakis Emmanouil G. Spanakis Haridimos Kondylakis
1.1	10/10/2013	Draft	Manolis Tsiknakis Emmanouil G. Spanakis Haridimos Kondylakis
1.2	10/21/2013	Draft	Manolis Tsiknakis Emmanouil G. Spanakis Haridimos Kondylakis Sfakianakis Stelios (FORTH)
1.4	10/23/2013	Draft	Manolis Tsiknakis Emmanouil G. Spanakis Haridimos Kondylakis Stelios Sfakianakis
2.0	10/23/2013	Draft	Manolis Tsiknakis Emmanouil G. Spanakis Haridimos Kondylakis Stelios Sfakianakis Feng Dong

List of contributors

- Tsiknakis Manolis (TEI-CRETE)
- Kostas Marias (FORTH-ICS)
- Emmanouil G. Spanakis (FORTH-ICS)
- Haridimos Kondylakis (FORTH-ICS)
- Sfakianakis Stelios (FORTH)
- Feng Dong (BED)

Contents

Table of Contents

1 Executive Summary

MyHealthAvatar proposes a solution for access, collection and sharing of long term and consistent personal health status data through an integrated environment, which will allow more sophisticated clinical data analysis, prediction, prevention and *in silico* treatment simulations tailored to the individual citizen. The proposed solution will be able to support: Information collection and access (Internal data repositories to store individual data for the avatars; links to external sources; model repositories, information extraction from the web and data collection using mobile apps; semantics and linked data to support the data/model searching and reasoning.), Data management and sharing, as well as Information analysis using integrated toolboxes.

MyHealthAvatar follows recommendations from relevant VPH activities on “Digital Patient”. MyHealthAvatar architectural platform will be designed as an integrated facility that allows multiple functionalities rather than just a data storage facility as in the previous attempts. It’s distinctive features include:

- ICT utilities to support data collection with minimal user input, including web information extraction, mobile apps, etc.
- ICT toolbox to support clinical decisions by using simulation models and by using visual analytics.
- Data and model repositories to provide rich resources of data and models
- Ontology and RDF repositories to support data search and reasoning.
- A cloud based ICT architecture that allows the access of data from a range of different sources, and integration of the repositories, the toolbox and the ICT utilities.
- A local cloud solution to support the computing requirement for the avatars without remote data transfer.
- A proof of market on open sources for MyHealthAvatar APIs

In this deliverable we describe MyHealthAvatar architectural blueprint and implementation activities. The emphasis, in order to meet project objectives, for the definition and design of MyHealthAvatar’s architectural scheme is on the selected standards and the technologies user and functional requirements. We followed the IEEE 1471 recommendation that 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. The steps were on capturing stakeholder needs; making a series of architectural design decisions that resulted in a solution that meet these needs, assessing it against the stakeholder needs, and refining the solution until it is adequate and captures the architectural design decisions in a complete “Architectural Description”. These iterative activities formed the core of the architecture definition process. The aim of this deliverable is to feed the subsequent tasks in the building of the system.

2 Introduction

This deliverable aims to define the architecture of the MyHealthAvatar 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. 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 platform.

2.1 Purpose of this document

Over the past few decades, the complexity of software systems has increased substantially. Complexity is a key concern that we would like software architecture to address. This complexity presents itself in two primary guises:

- 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 abstractions that hide unnecessary detail, providing unifying and simplifying concepts, decomposing the system, 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. 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.

Given that we need to decompose the system to address complexity, what new problems emerge that have to be dealt with by the architecture?

- 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 behavior 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.

MyHealthAvatar architectural definition aims to allow the access of information from a range of external sources, e.g. individual hospital records, and other data/model repositories, and also supports data management, transfer, security and sharing controlled by individuals.

3 Methodology

In recent years a realization has grown of the importance of software architecture. According to Bass et al 0the 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 [Bass 1997]. The IEEE recommendation 0defines 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 [IEEE 2000]. 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 a system architect proceed to design the architecture? A proposed architecture definition process is shown in the figure below:

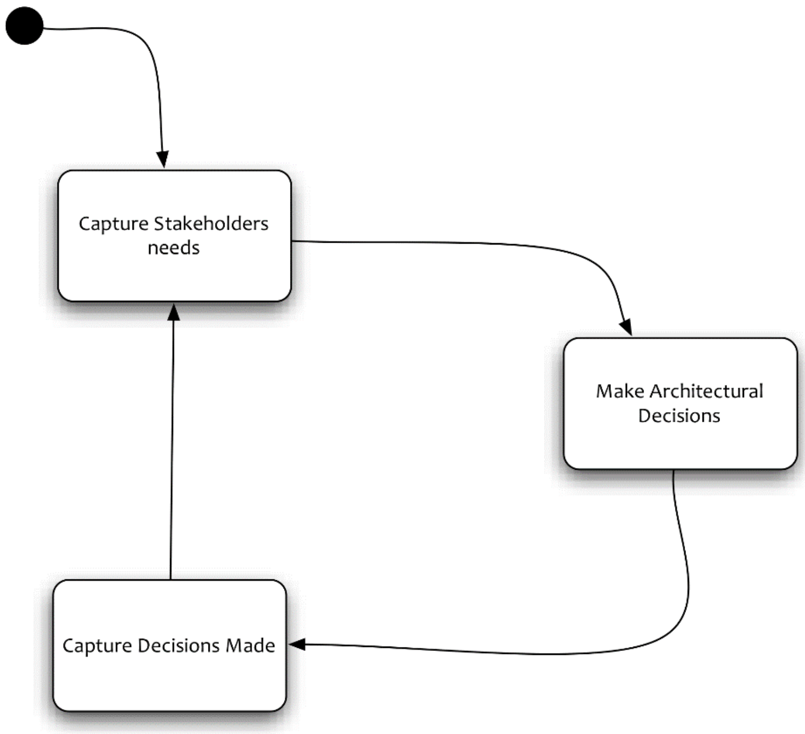

Figure 1: The architecture definition process

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.

3.1 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.

3.1.1 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.

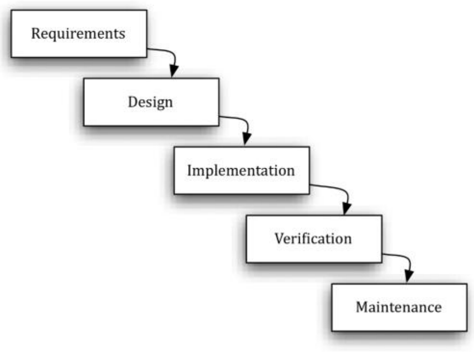

Figure 2: The waterfall model

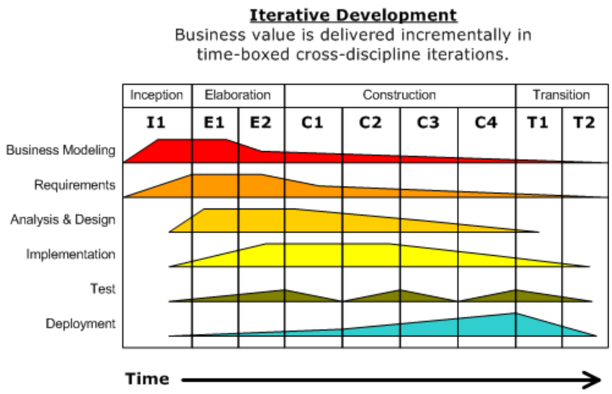

Figure 3 The iterative process of RUP

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).

3.1.2 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.

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.

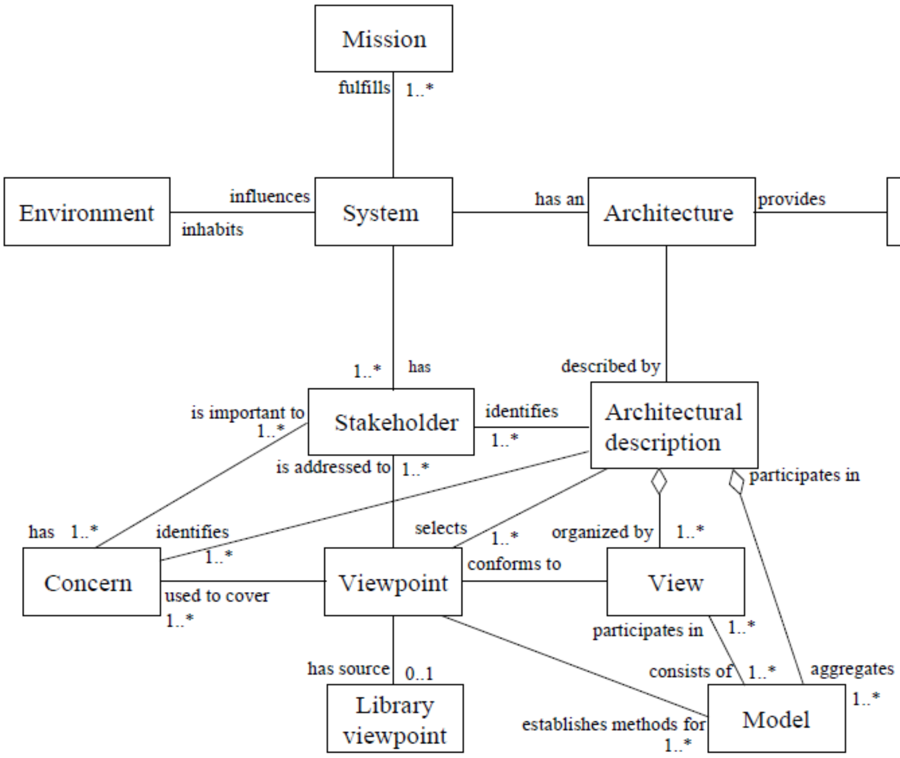

Figure 4 Conceptual model of architectural description from IEEE1471

IEEE 1471 defines a set of requirements for conforming ADs which 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

3.1.3 4+1 Views Model

The "4+1" views model was originally developed in 1987 by Phillipe Kruchten of Rational Software **Error! Reference source not found.**. 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

3.1.4 Rozanski and Woods Viewpoint Set

Rozanski and Woods [Rozanski 2011] 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.

3.2 The selected approach

The approaches that are centered on the notion of stakeholders, views, and viewpoints are in conformance to the ISO/IEEE 42010. The Rozanski and Woods selection of the viewpoints but also their introduction of the perspectives (i.e. nonfunctional, quality attributes) in the discussion seems to be the most appropriate for describing the MyHealthAvatar 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”, which means to continue until the basic requirements are met and the identified risks are addressed [Fairbanks 2010].

4 MyHealthAvatar Architecture definition and design

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 architecture definition process on the set of viewpoints proposed by Rozanski and Woods [Rozanski 2011]. 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.

4.1 Stakeholders

A stakeholder is anyone who has an interest in or concerns about the system that we actually building.

In MyHealthAvatar the most important stakeholders are:

- Researchers/Medical personnel expert domain Users. These can be further classified in bioinformaticians, clinicians, users of clinical trials management systems, etc.
- Patients. In MyHealthAvatar where personalized provision of health and patient empowerment services are to be delivered, patients represent a prominent 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 expert 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.

4.2 MyHealthAvatar Use cases (to include Table HK)

4.2.1 MyHealthAvatar Use Cases (UCs)

In this section we present the various use case that are described in WP2. We try to identify the various requirements that will be used to describe and design the architecture structure of MyHealthAvatar platform.

4.2.1.1 UC1: MyHealthAvatar User accounts

Users (citizens) will be able to log onto the system using their username and password. New users will be able to sign up to the system by creating basic personal information including security questions. Informed consent and privacy: Users will need to accept the privacy policy and the "terms and conditions" of using the MyHealthAvatar platform. Upon log into the system, users will be able to enter, browse their data, explore medical information, and communicate with other patients. Users will be able to view and interact with an avatar

- a 3D representation of the human body. It will allow the End User to click with the computer mouse on a particular part of the avatar "body" to trigger a search of medical records to retrieve relevant information

MyHealthAvatar is a platform for End-Users who want to share their health information to create collective knowledge about disease, health, and treatments. In order to achieve this goal advanced Informed Consent and Privacy Policy Scenario / Use Case should be implemented. End User has the GUIs, functionalities and tools in the frames of MyHealthAvatar platform to accept, reject, print or revise at any time the Privacy and Informed Consent settings. There will be two types of users, the first type includes patients and citizens for their life time data collection, the second type includes doctors who will be linked to the avatars from citizens/patients for clinical practices and medical research purpose

4.2.1.2 UC2: 3D avatar visualization

MyHealthAvatar platform would propose an avatar - a 3D representation of the human body - to allow End Users (e.g. patients, doctors) to visualize patient medical records in a new way. Similar to IBM's Anatomic and Symbolic Mapper Engine (ASME), this visualization method would allow the End User to click with the computer mouse on a particular part of the avatar "body" to trigger a search of medical records to retrieve relevant information.

Figure 5 The ASME system will allow doctors to "click" on different parts of the 3-D avatar of the human body[3]

The avatar will be used as a means for presenting general medical knowledge to the citizen users. Users will be able to select individual parts and see related medical information such as anatomy. The information may also include medicine and food.

4.2.1.3 UC3: MyHealthAvatar data browse

Upon log in to their own account, users will be able to browse their own data, including all the personal health status data collected through the avatar system, plus medical records and clinical data from the hospitals. The avatar system will need to offer tools that support effective data query and search, such as filtering. The 4D avatar will play an important role in presenting the data. Users will be able to select individual parts of the avatar body to view the data associated to the selected parts. Different colors or textures will be assigned to individual parts of the 4D avatar to represent their health status. For example, if the heart has a serious problem it will be highlighted using a unique color or texture. *The avatar system will also need tools which will help users to analyze medical images.* In this context, v*isual representation in multi-layer geometries and colors to support a body*

[3] http://www-03.ibm.com/press/us/en/pressrelease/22375.wss (June 2013)

(anatomy) centered visualization of health status data for the clinical case of Alzheimer disease will be studied. MyHealthAvatar will provide related tools for data browsing, *diagnosis, monitoring and treatment.*

4.2.1.4 UC4: MyHealthAvatar virtual community

This case describes the search framework from end-users' perspective and it is focused on listing all MHA registered end-users with ability to apply advanced search filters:

- Age
- Gender
- Votes (Likes)
- Treatment
- Symptom
- Interests
- Country
- City
- etc.

It is important to mention that every end-user should confirm the possibility to visualize his/her profile publically or privately. Only public profiles should be visible in search results. Additionally, the search function is suggested to be accessible only for end-users with public profiles. This case also provides a social media that allows patients to build up a virtual community by sharing their daily activities (e.g. how many exercises they have done), exchanging their experiences. It should also provide a link to Facebook/Twitter.

4.2.1.5 UC5: Self data collection for citizens

This case seeks new solutions for increasing the quality and sustainability of future healthcare systems by actively engaging citizens in monitoring their own health through self-collection of life-logging data. The development and treatment of many diseases are affected by our life styles and environment. A long term monitoring of these factors, especially through the self-involvement of patients, is extremely valuable in supporting individualized health prediction and treatment. Many studies have shown compelling needs in self-life logging and self-monitoring of patients, which has great potential in leading to preventive medicine, cost saving and enhanced quality in future healthcare. We aim to create a symbiotic relationship of available technology today and MyHealthAvatar platform. The goal is to respond to the fast growing demand for developing new technologies and services for self-monitoring for supporting wellness, fitness and prevention of the most common chronic diseases (i.e. cardio-vascular and stroke, diabetes, rheumatic problems, respiratory problems and COPD, etc.). Mobile applications will monitor user's "health-status", "lifestyle" and "wellness" and upload data to the MyHealthAvatar system for close monitoring of health conditions and prevention of many diseases. The system then will be able to analyze user's lifestyle and medical data. Special "alerts" will be applied to support end users with feedback supporting and assisting their daily activities and well-being.

An interface for patients writing a diary is very helpful to collect patient specific data related to their disease. This can be partly structured: e.g. body weight, heart rate, blood pressure, temperature, medicine taken, etc. It can also include structured data of scoring systems, e.g. physical and/or psychological and/or emotional status. In addition free text entry needs to be allowed. More specifically, we explore various ways for the data collection in the avatar to monitor users' health-status, lifestyle and wellness.

These include:

- Web interface for data entry
- Sensors (e.g. blood glucose, blood pressure, heart rate, locations, steps, sleep)
- Mobile apps

Mobile apps will be used to monitor the health status of the users (e.g. mood, food). We will also explore the possibility to extract health related information from electronic cards (e.g. purchase of food and drink, daily exercises in gyms), as well as from social network. We will also look into the possibility of implementing an advanced Patient Devices Software Development Kit (SDK or "devkit"). Also we will try to create mobile application that will use tools provided by MyHealthAvatar platform allowing the capturing of biomedical signals using only sensing devices attached already to smartphones commercially available today.

4.2.1.6 UC6: MyHealthAvatar Toolbox

Remote Monitoring: To allow for remote monitoring from the doctors using the avatar system, we need to link the avatar system to the hospital information system (which is again another use case). This will subsequently allow the transfer of the avatar data into the hospital records. The Remote Monitoring tool/frame collects and processes patient care information from supported healthcare devices that conform to standards.

Remote Consultation: The MyHealthAvatar system can be used for direct interaction between the patient and the physician. Such an interaction might provide the following functionalities: a) Making appointments with the physician; b) Asking questions to the physician; c) Giving advice to the patient by the physician

Knowledge discovery: Patients interested in the most recent and personalized information about their disease, treatment and prognosis. MHA platform could contain an ontology-based Knowledge Discovery (KD) module able to connect highly heterogeneous data and textual information. The semantic framework could be based on gene, tissue, disease and compound ontologies (important for drugs and clinical research frames). This framework could contain information from different organisms, platforms, data types and research areas that is integrated into and correlated within a single searchable environment using search algorithms. It could provide a unified interface for all MHA users to formulate, explore and identify new information (according to specific preferences and needs) across vast collections of available experimental and research data. KD module could combine classical keyword-based search with text-mining and ontologies to navigate large results sets (internal & external) and facilitate information and/or knowledge discovery. End users could be provided with an advanced ontology based (Gene Ontology (GO) and Medical Subject Headings (MeSH)) 'Table of Contents' in order to access, explore, structure (quickly) the millions of available resources (PubMed abstracts, news, clinical trials info) according to the predefined topics of interest (Allergy, Cancer, etc.).

4.2.1.7 UC7: Link MyHealthAvatar to hospitals

End User has the GUIs, functionalities and tools in the frames of MyHealthAvatar platform to enter, import, store and export personal medical data with hospital information systems. We will optionally use ObTiMA as a dummy system to mimic external hospital system. ObTiMA, an ontology-based clinical trial management system, has been developed as a proof-of-concept application to highlight the possibilities of ontology based creation and managing of clinical trials within the ACGT (Advancing Clinico-Genomic Trials on Cancer) project. ObTiMA has a modular architecture with a core basic module for data management of clinical trials. Different other modules are under development in the frames of p-medicine project. The data stored in ObTiMA are relevant for the Health Avatar to enhance the system with relevant clinical trial data. On the other hand the info stored in MHA might be of relevance for a clinical trial. As result, the bidirectional data upload from MHA to ObTiMA is needed. This Scenario / Use Case describes the bilateral linkage between ObTiMA and MHA by being focused on the Operational Data Model (ODM). Similar work will be carried out using other external hospital information systems and personal health records.

4.2.1.8 UC8: MyHealthAvatar simulation models

In this particular use case end users, through a usable graphical user interface, using MyHealthAvatar platform tools and functionalities will be able to create and execute biological simulations scenarios. The user will be able to select one of the biological simulation models available in the Model Repository and one of the sets of clinical data available in the Clinical Data Repository (or uploads a set from his computer). Afterwards he/she executes a biological simulation. Finally he/she retrieves the results of the simulation and proceeds to their evaluation. End-User also has the GUIs, functionalities to manage the content of the Model Repository and the Clinical Data Repository (related to simulation models).

4.2.1.9 UC9: Utilization of personal genomic information for the individualization of MHA platform

Tis use case aims at the **development of MHA technology as an individualized medicine platform by the utilization, interpretation and integration of personal genomic information into health medical history record**. This technically high-level and complex use case involves a number of health and lifestyle related processes, tools and services which translate genomic data to genetic predisposition evaluation and health risk estimation, pharmacogenomic predictions, histology and pathway visualizations etc. in order to support and facilitate advanced individualized medical decision making (i.e. integrative individual patient case view, specification of simulation models, therapy selection etc) and provide with guidelines for preventive medicine.

4.2.1.10 UC10: Building personalized CHF related risk profiles and “real-time monitoring” services

A major challenge related to caring for patients with chronic conditions is the early detection of exacerbations of the disease that may be of great significance. Generally, cardiovascular disorders as chronic diseases require a continuous record for patient’s status. In addition, there are many cases where more than one medications are prescribed due to disease progression or due to the appearance of co-morbidities (respiratory comorbidities, renal dysfunction, cognitive dysfunction, depression and in some cases arthritis). To this respect, there is a generally need for providing information in both the treating physicians but also to the patient regarding disease status and treatment outcome. In addition to the above the dedicated clinical personnel should be contacted immediately and possibly intervene in time before an acute state is reached, by changing medication, or any other interventions, in order to ensure patient safety. There is a need to support real-time remote monitoring of patients diagnosed with congestive heart failure and MHA, enhanced with semantic technologies, may host personalized, accurate and up-to-date clinical information.

In order to tailor the proposed system to the patient’s profile and assist physicians in selecting people who are predisposed by coronary disease, hypertension, or valvular heart disease; we build a CHF related risk profile based on a risk appraisal function that is based on the diagnostic criteria [i.e. the Framingham Heart Study (486 heart failure cases during 38 years of follow-up)]. The predictors used are based on Age, Coronary heart disease and Valve disease status provided by the patient Electronic Health Record (EHR), as well as on HR, on blood pressure and on Body Mass Index (BMI) provided by the pulse oximeter, the blood pressure monitor and the weight scale respectively. The calculated risk probability may be used to alter the default threshold values (higher risk probability adds more constraint on the physiological patterns). Furthermore we present what else data regarding patients health status could be embed into the platform towards the creation of a profile with necessary information for both patient and treating physicians. To this respect an approach of presenting data regarding demographic, physiology, diagnostic test results and disease management (i.e. prescribed drugs) is provided.

4.2.1.11 UC11: Decision making tools regarding emergency situations in clinical practice. The example of anti-platelet & anticoagulation therapy in the pre-operative patients

Hemostasis disorders can develop due to a deficiency or defect in an individual's platelets or clotting mechanisms. Dysfunctions can lead either in bleeding disorders (hemophilia) or in over-clotting disorders such as thrombosis. Dysfunctions that lead in thrombus formation can be related with morbidities such as cardiovascular disorders (coronary disease, heart attack, angina, congestive heart failure and valve disease), pulmonary embolism, stroke and transient ischemic attacks, deep vein thrombosis, peripheral vascular disease (PVD), phlebitis and in some cases obesity. Patients that are diagnosed with over-clotting deficiencies are treated with anticoagulant or anti-platelet therapies as a preventive care. Several single nucleotide polymorpshisms (SNPs) are known regarding drug-targets or metabolizing enzymes (mainly of Cytochrome P450 family) of anti-platelet and anticoagulant therapies. Some well-known examples are the Vitamin K epoxide reductase complex subunit 1 (VKORC1) where specific gene mutations have been related with deficiencies in Vitamin-K-depedent clotting factors and the response to anticoagulant therapies of warfarin and acenocoumarol. In addition, regarding the metabolizing enzymes of P450 family, CYP2C19 is the main metabolic enzyme for the activation of the anti-platelet agent clopidogrel. The latter, is a pro-drug activated in the liver by cytochrome P450 enzymes, mainly CYP2C19. Genetic polymorphism (CYP2C19*2, CYP2C19*3 and CYP2C19*17) exists for CYP2C19 expression, with approximately 5% of Caucasian and 20% of Asian populations being poor metabolizers with no CYP2C19 function. Due to the above, anti-platelet and anticoagulant agents that are administered in clinical practice, appear to have a wide inter-subject variability in their pharmacokinetics and thus in pharmacodynamics. Antiplatelet and anticoagulation therapies are typical examples where therapeutic drug monitoring is applied for every patient as well as pharmacogenomics information are taken into account and several algorithms have been created in order to integrate data and improve pharmacotherapy. Moreover, there are emergency cases such as pre-operative status, where an adjustment in dose should be applied for patients following anti-coagulation and anti-platelet therapies in order to avoid bleeding problems during surgery or during recovery.

Summarizing the above information there are cases where additional information are needed but not easily attainable due to lack of clinical data. To this respect in silico approaches (such as Physiologically-based pharmacokinetic/pharmacodynamic modeling) seem capable in providing evidence regarding possible treatment outcomes and organized in order treating physicians will be able to avoid as much as possible “guesswork” for a specific patient. The availability of creating “virtual cohorts” of patients and in silico approaches can assist in generating predictive approaches towards improved personalized medicine.

4.2.1.12 UC12: Brain Trauma

A pre-injury clinical profile of patient is a critical aide that can help the clinicians by providing a better insight and possibly improve the clinical outcomes by circumventing the barriers imposed by the heterogeniety of traumatic brain injuries. Individualized treatment and targeted therapies based on patients' data are imperative both from the patients' perspective and also from the clinicians point of view and can ensure more promising outcomes and better prediction and prevention. Such a profile can be used to establish models of pathphysiologic mechanisms significant to pathoanatomic presentations of brain injuries.

A clinical phenotype of the patient has to be developed based on pre-injury characteristics. The patients' past medical history, drug history, demographic information, family history, socioeconomic status and life style and habits contribute significantly towards accurate assessment and management in case of brain trauma or cerebrovascular accident. In cases requiring surgical intervention, pertinent medical history can provide information about any co-existing brain lesions or any medical condition that contra-indicates general anesthesia or surgery e.g., patients on anticoagulation therapy. Studies show that prognosis after TBI is strongly correlated to the medical history of the patient and characteristics like age, alcoholism, drugs, cardiac problems, hypertension, liver dysfunction, diabetes and renal impairment etc. can affect the treatment regimen and morbidity and mortality. For example, subdural haematomas (SDH)can be difficult to differentiate from extradural haematomas (EDH) if the size is small, however, the aetiology, management and outcomes can be significantly different. History of repeated brain injury, e.g., sports related, can exacerbate the symptoms. Small children and old people are prone to falls and history of head trauma may be difficult to identify, while road traffic accidents are more common in young and middle aged people with associated skull fractures. Family history of brain aneurysms is an important consideration for suspected subarachnoid haemorrhage (SAH).

Junior doctors in A&E who do not have adequate experience in interpreting ct scans can sometime misdiagnose and this undertriage or overtriage can lead to a treatment which is different than what is required. An automated system providing second opinion to the radiologist can help improve the sensitivity and specificity of diagnoses and reduce inter-observer variability.

The data repository available within MyHealthAvatar can allow researchers develop mathematical and computational models based on gender, race, ethnicity categories, age, lifestyle, education and medical data and this can significantly contribute to innovative healthcare practices. These data from MyHealthAvatar combined with presenting complaints at the time of admission, history of presenting illness, clinical and neurological findings such as the Glasgow Coma Scale, pupil reactivity, loss of consciousness, vomiting episodes, ENT bleeding, fits, headaches and vital signs can be combined with image features from CT scans and Marshall CT Classification to develop a prognostic model for traumatic brain injuries (TBI).

4.2.2 Overview of requirements

The project has carried out analysis of detailed end-user requirements and needs by collecting an initial set of Scenarios / Use Cases. These cases were collected by consortium members through interaction with stakeholders of MyHealthAvatar system, including citizens/patients, clinical doctors and clinical and IT researchers. These exercises allow us to describe in details the Scenarios / Use Cases proposed for implementation. Each of these cases addresses a use scenario from a particular user perspective, either as a patient, or as a doctor, or as a clinical or IT researcher.

In MyHealthAvatar two general categories of scenarios will be investigated:

- System use cases: these are the cases that describe the functionalities of the MyHealthAvatar system from the perspectives of both clinicians and citizens/patients.
- Clinical use cases: these are the cases that describe how to use the data from the MyHealthAvatar system in real clinical scenarios.

The Appendix of this document shows the descriptions of the use cases that have been agreed within the consortium so far.

Notably, the team of MHA project is continuously working on the definition of the use cases before they are finalized in the next project's task Nr 7.1, named "Scenarios and use cases for MyHealthAvatar: PM10=>PM18 (Task Leader: USAAR)".

4.3 Architectural Views

The definition of the architecture entails always the risk that some important stakeholders concerns are not captured that has implications in the implementation and the deployment of the system. The most important such concern relates to the protection of the sensitive patient data, the patients' anonymity, and the access control. The general contingency plan in these concerns is to follow an iterative, agile approach in the system definition and building, where the requirements are revisited and appropriate correcting actions are taken if needed. The consortium as a whole has a lot of experience from different perspectives (technical, clinical, legal), which guarantees the close monitoring of the realization of the MyHealthAvatar platform. In this section we describe the architecture of the MyHealthAvatar platform from the various viewpoints of Rozanski and Woods.

4.3.1 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 system, when considered a single, unified system, can be seen in Figure 6. The main feature of this type of diagram is that it shows clearly the channels of communication with all sub-systems and external systems. It clearly presents the key components that must be built during our work plan.

The key components of the related working task include: 1) user requirements – objective 1; 2) Internal data repositories and an internal model repository – Objective 2.1-2.2; 3) ICT architecture that support data access to internal and external resources and data management – Objective 2.3; 4) Data collection utilities – Objective 2.4; 5) Semantics and RDF repository to support data search and reasoning – Objective 2.5; 6) Simulation and data analysis toolbox – Objective 2.6, 2.7 and 2.8; 6) Demo & evaluation – Objective 2.9; 7) Investigation of the legal and IPR aspects of the avatars – Objective 3; 8) Understanding of clinical acceptability – Objective 4; 9) Recommendations for the future work – Objective 5 ; 10) Dissemination and exploitation of the results to influence the future healthcare system – Objective 6

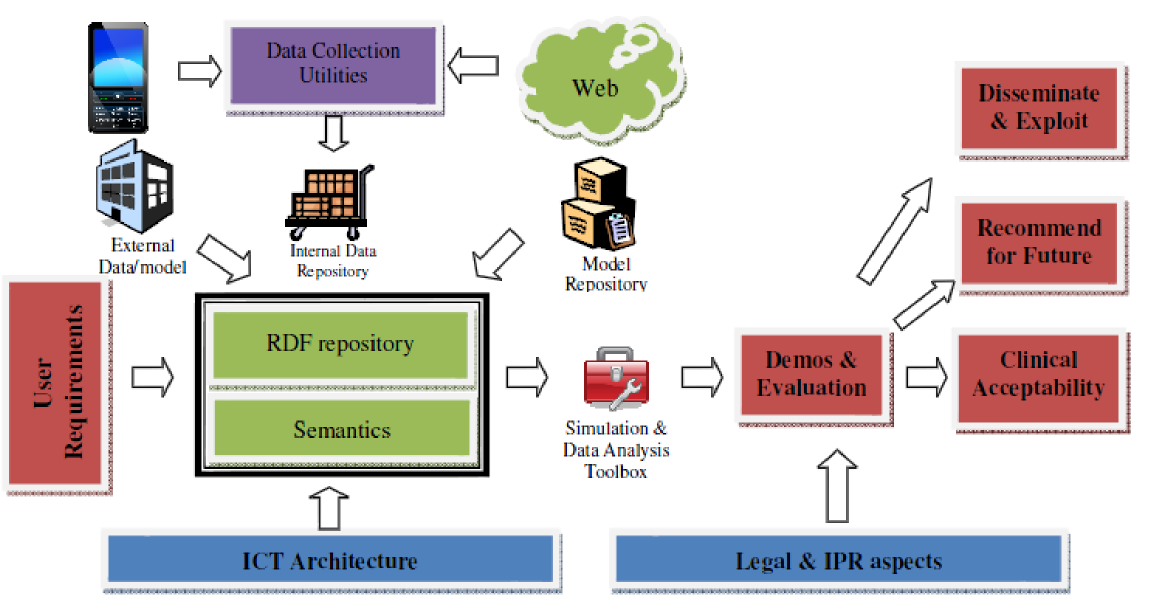

Figure 6 MyHealthAvatar platform as a unified system and its interactions with external entities

The major stakeholders of the system, as collected from the requirement analysis phase include:

- Citizens
- Patients
- Clinicians and
- Medical researchers

It is our goal that the requirement analysis of MyHealthAvatar will clearly delineate the borders of the system architecture. According to these requirements, the following are external entities that are currently considered for the system under development:

- HIS, EHR and PHR systems
- Clinical processed data and data from internal Warehouse (lab results, images etc.)
- eCRF with filed in data from ObTiMA. Health Avatar with clinical trial related data (i.e. laboratory results, pre-operative state, etc.)
- Drug data
  - Pharmacokinetic and Pharmacodynamic properties
  - Population data
  - Demographic
  - Genetic
  - Physiology
  - Pathology

- PubMed Repository, Clinical Trials information, news articles, etc.
- Social Networks and other sources of online activity of the users
- Model repositories that contains simulation models

4.3.2 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 7. 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.

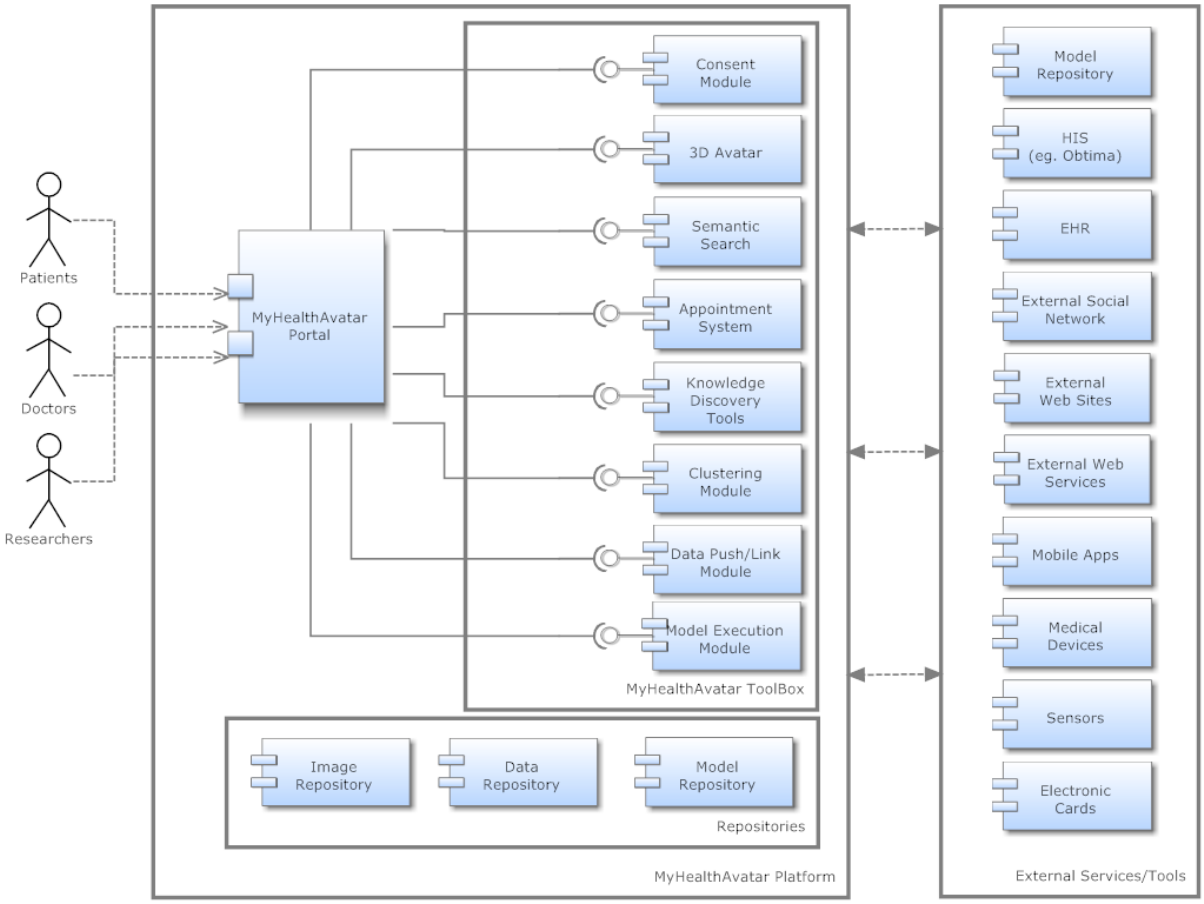

Figure 7 The main components of the system and their interactions

4.3.2.1 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[4]. An actor may be a class of users, roles users can play, or other systems.

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 used an updated version of the use cases as described in D2.2 in order to elicit the MyHealthAvatar functional requirements.

[4] 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

4.3.2.1.1 The Avatar System

4.3.2.1.1.1 UC-1 MyHealthAvatar User Accounts

In this use case the end-user is being able to be registered to the system. If the user is already registered he can log-in into the MyHealthAvatar platform.

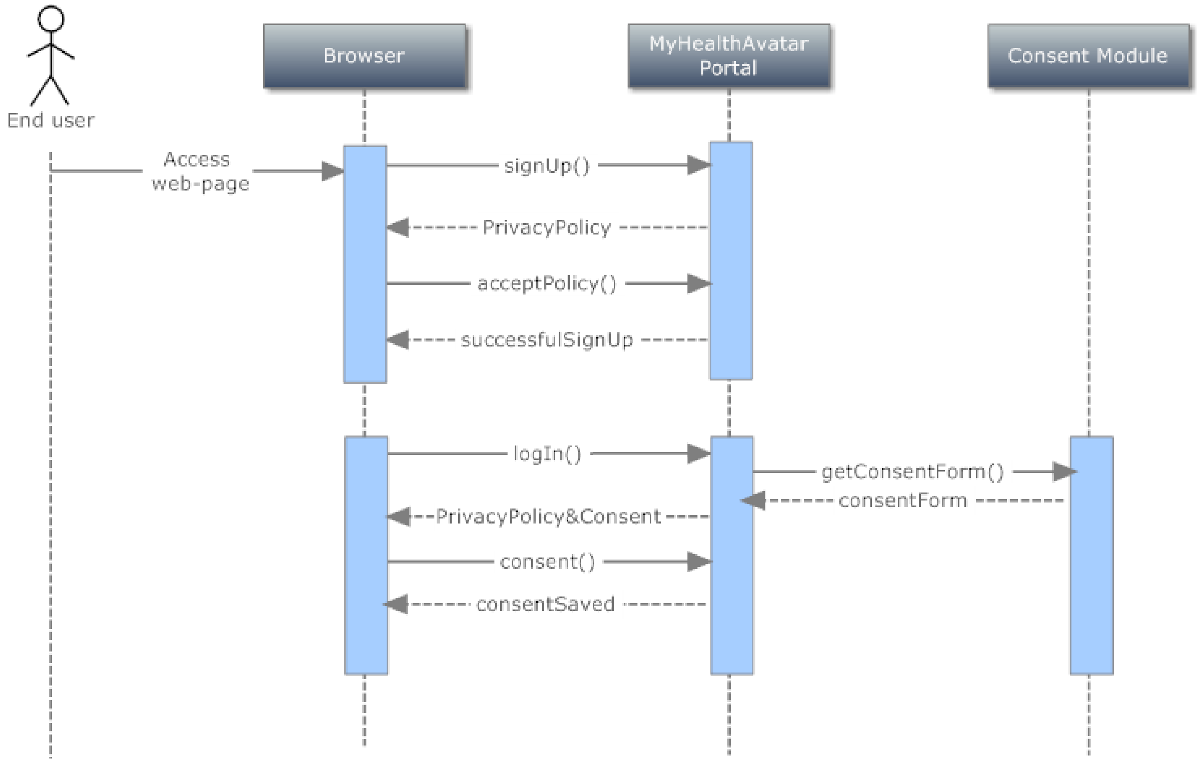

Figure 8. Sequence diagram for sign-up and log-in into the MyHealthAvatar platform.

The component diagram of this scenario is shown in the following Figure.

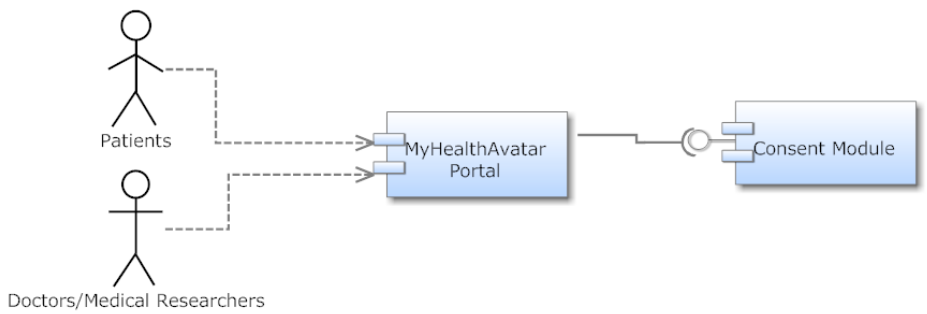

Figure 9. The component diagram for UC1

The description of each component is provided bellow

Component Name	**MyHealthAvatar Portal**
Responsibilities	To allow citizens and doctors log-in into the system, visualize their results and browse their information.
Collaborators	Consent Module
Rationale	There should be a component with a user friendly interface to allow interacting with the end-users
Issues and notes	No issues and notes

Component Name	Consent Module
Responsibilities	To allow patients decide the information that they should share with the platform and the other users
Collaborators	MyHealthAvatar portal
Rationale	The patients should decide how their information is being used.
Issues and notes	No issues and notes.

4.3.2.1.1.2 UC-2 3D avatar visualization

This use case focuses on the 3D representation of the human body. The sequence diagram for this use case is shown in Figure 10 whereas the component diagram is shown in Figure 11.

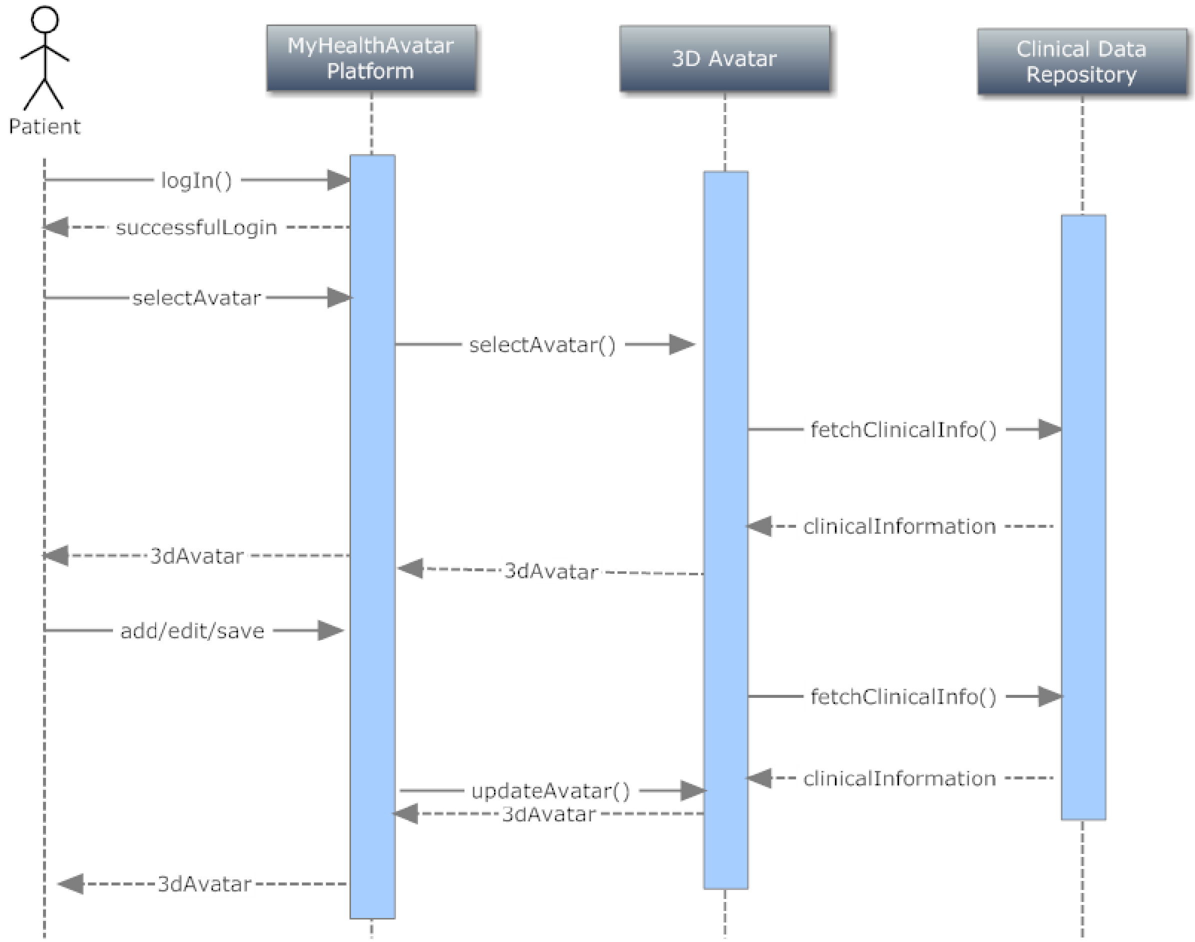

Figure 10. The sequence diagram of UC-2

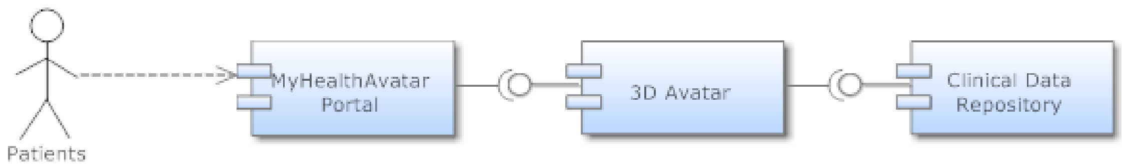

Figure 11. The component diagram of UC2

Bellow we describe in detail the different components.

Component Name	**MyHealthAvatar Portal**
Responsibilities	To allow citizens and doctors log-in into the system, visualize their results and browse their information.
Collaborators	3D Avatar

Rationale	There should be a component with a user friendly interface to allow interacting with the end-users
Issues and notes	No issues and notes

Component Name	**3D Avatar**
Responsibilities	Users will be able to select individual parts of the avatar body to view the data associated to the selected parts.
Collaborators	MyHealthAvatar Portal, Data Repository, EHR
Rationale	Each hospital has its own information system to store patient records.
Issues and notes	3D avatar should be available in multiple platforms.

Component Name	**Clinical Data Repository**
Responsibilities	To allow the storage of data in a secure, distributed, highly accessible environment.
Collaborators	3D avatar
Rationale	There should be a central repository of data, that will allow further elaboration on the collected data
Issues and notes	Probably there should be defined an ontology based data access layer to allow access to heterogeneous data.

4.3.2.1.1.3 UC-3 MyHealthAvatar data browse

This use case focuses on providing to the users a mechanism to browse their own data through the avatar system. The sequence diagram of this use case is shown in Figure 12 whereas the component diagram is presented in Figure 13.

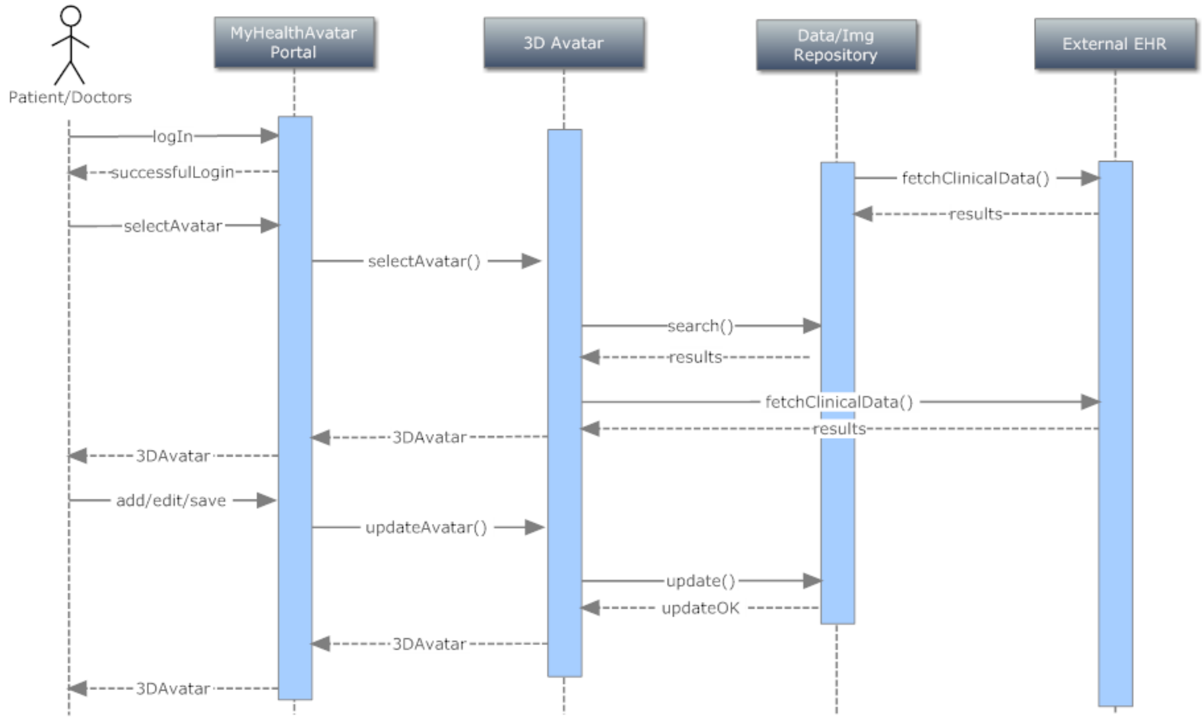

Figure 12. The sequence diagram of UC-3

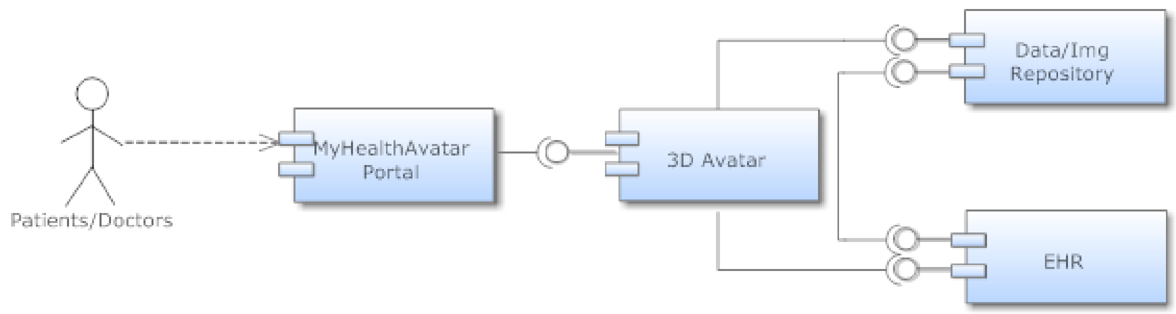

Figure 13. The component diagram of UC-3

Bellow we describe in detail the different components.

Component Name	**MyHealthAvatar Portal**
Responsibilities	To allow citizens and doctors log-in into the system, visualize their results and browse their information.
Collaborators	3D Avatar
Rationale	There should be a component with a user friendly interface to allow interacting with the end-users

Issues and notes	No issues and notes

Component Name	**3D Avatar**
Responsibilities	Users will be able to select individual parts of the avatar body to view the data associated to the selected parts.
Collaborators	MyHealthAvatar Portal, Data Repository, EHR
Rationale	Each hospital has its own information system to store patient records.
Issues and notes	3D avatar should be available in multiple platforms.

Component Name	**Data Repository**
Responsibilities	To allow the storage of data in a secure, distributed, highly accessible environment.
Collaborators	3D avatar, EHR
Rationale	There should be a central repository of data, that will allow further elaboration on the collected data
Issues and notes	Probably there should be defined an ontology based data access layer to allow access to heterogeneous data.

Component Name	**EHR**
Responsibilities	To store the patient records in hospitals
Collaborators	3D avatar, Data Repository
Rationale	Each hospital has its own information system to store patient records.
Issues and notes	Ontology-based mediation of information could be exploited to achieve data translation on realtime.

4.3.2.1.1.4 UC-4 MyHealthAvatar virtual community

This use case describes the search framework in order to search for similar users and exploit external social networks. The sequence diagram of this use case is shown in Figure 14 whereas the component diagram is shown in Figure 15.

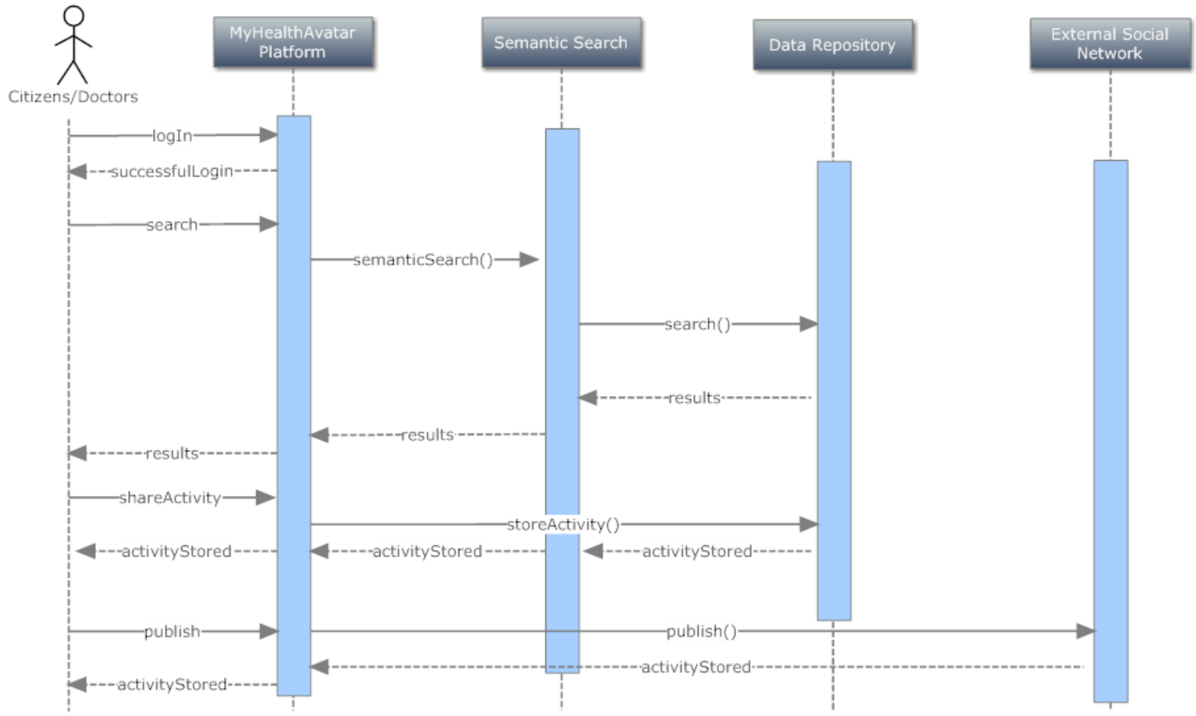

Figure 14. The sequence diagram for UC-4

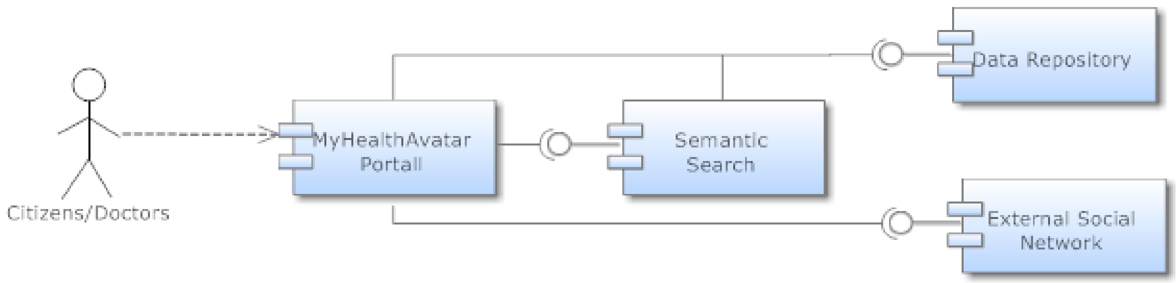

Figure 15. The component diagram for UC-4

Bellow we explain the different components.

Component Name	**MyHealthAvatar Portal**
Responsibilities	To allow citizens and doctors log-in into the system, visualize their results etc.

Collaborators	Semantic Search, Data Repository, External Social Network
Rationale	There should be a component with a user friendly interface to allow interacting with the end-users
Issues and notes	No issues and notes

Component Name	**Data Repository**
Responsibilities	To allow the storage of data in a secure, distributed, highly accessible environment.
Collaborators	MyHealthAvatar Portal, Semantic Search
Rationale	There should be a central repository of data, that will allow further elaboration on the collected data
Issues and notes	Probably there should be defined an ontology based data access layer to allow access to heterogeneous data.

Component Name	**External Social Network**
Responsibilities	To provide social information related with citizens health
Collaborators	MyHealthAvatar Portal
Rationale	The idea is to extract health related information from the citizens social network
Issues and notes	The API from external social network should be used.

Component Name	**Semantic Search**
Responsibilities	To enable searching over patients and data.
Collaborators	MyHealthAvatar Portal, Data repository
Rationale	Patients and doctors should be able to search information. More specifically, the following should be possible using this search engine. 1. Find patients with similar condition, symptom and treatments

	2. Find out symptoms and treatment for their conditions by looking at other fellow patients 3. Find out possible conditions for their symptoms by looking at other fellow patients 4. Find out possible treatments for their conditions by looking at other fellow patients 5. Find out "friends" and allow "followers" as in Facebook/Twitter 6. 6) Share activities, exercise experiences etc with friends and followers.
Issues and notes	The semantic search should be facilitated by a proper ontology.

4.3.2.1.1.5 UC-5 Self-data collection for citizens

This use case describes the data flow for collecting information by citizens and storing it to the MyHealthAvatar data warehouse. The sequence diagram is shown in Figure 16 whereas the component diagrams of this use-case is presented in Figure 17.

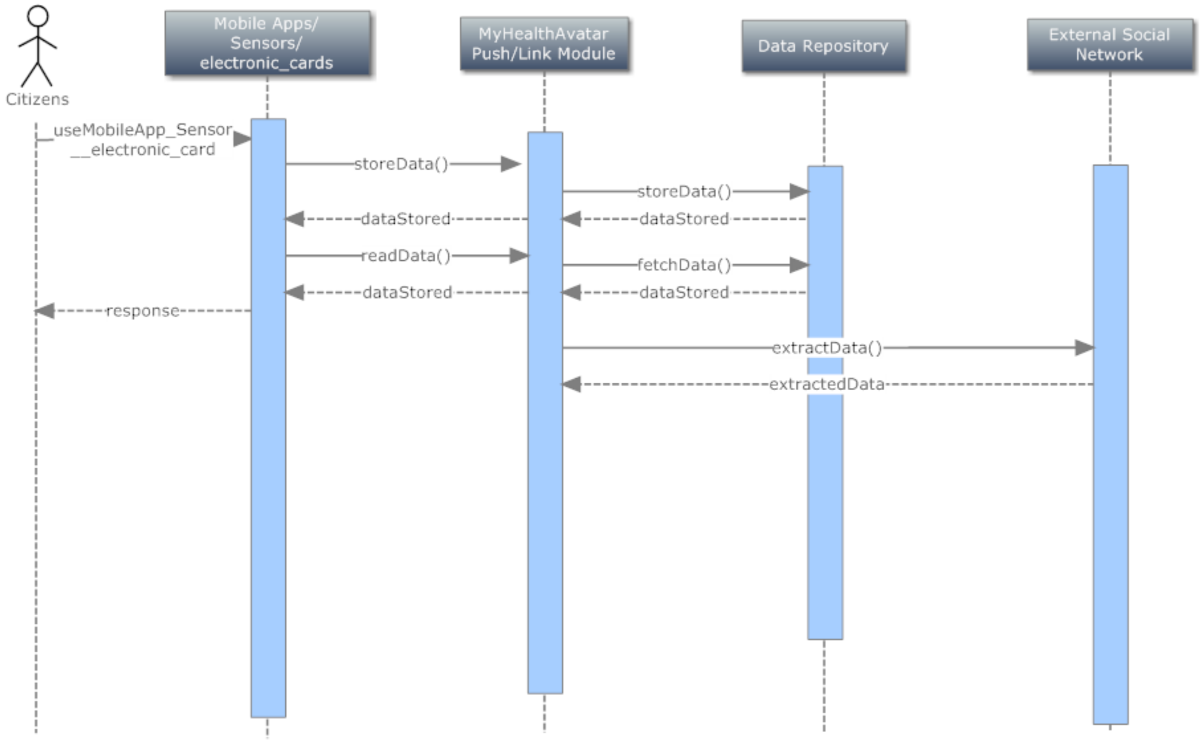

Figure 16. The sequence diagram for UC-5

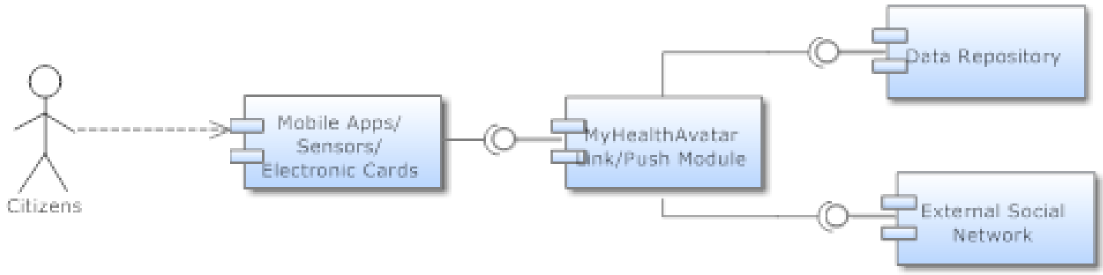

Figure 17. The component diagram for UC-5

The components shown in Figure 17 are explained bellow

Component Name	**Mobile Apps/Sensors/Electronic Cards**
Responsibilities	To allow actively engaging citizens in monitoring their own health through self-collection of life-logging data
Collaborators	**MyHealthAvatar Push/Link Module**
Rationale	We aim to create a symbiotic relationship of available technology today and MyHealthAvatar platform. Mobile applications, sensors etc. will monitor user's "health-status", "lifestyle" and "wellness" and upload data to the MyHealthAvatar system for close monitoring of health conditions and prevention of many diseases.
Issues and notes	Probably there should be defined a proper API in order to allow apps & sensors to interact with the MHA platform.

Component Name	**MyHealthAvatar Push/Link Module**
Responsibilities	To allow citizens and doctors log-in into the system, visualize their results etc.
Collaborators	Mobile Apps/Sensors/Electronic Cards, Data repository, Social Networks
Rationale	There should be a component with a user friendly interface to allow interacting with the end-users
Issues and notes	Probably there should be defined a proper API in order to allow apps & sensors to interact with the MHA platform.

Component Name	**Data Repository**

Responsibilities	To allow the storage of data in a secure, distributed, highly accessible environment.
Collaborators	**MyHealthAvatar Push/Link Module**
Rationale	There should be a central repository of data, that will allow further elaboration on the collected data
Issues and notes	Probably there should be defined a proper API in order to allow apps & sensors to interact with the MHA platform.

Component Name	**External Social Network**
Responsibilities	To provide social information related with citizens health
Collaborators	**MyHealthAvatar Push/Link Module**
Rationale	The idea is to extract health related information from the citizens social network
Issues and notes	The API from external social network should be used.

4.3.2.1.1.6 UC-6 MyHealthAvatar Toolbox

This use case has to do with remote monitor tools similar to the previous use case. The addition to the previous use case however is the remote consultation module in order for a patient to book appointments with the physician and the knowledge discovery module that processes data to provide personalized information. The sequence diagram of the aforementioned use-case is presented in Figure 18 and the corresponding component diagram in Figure 19.

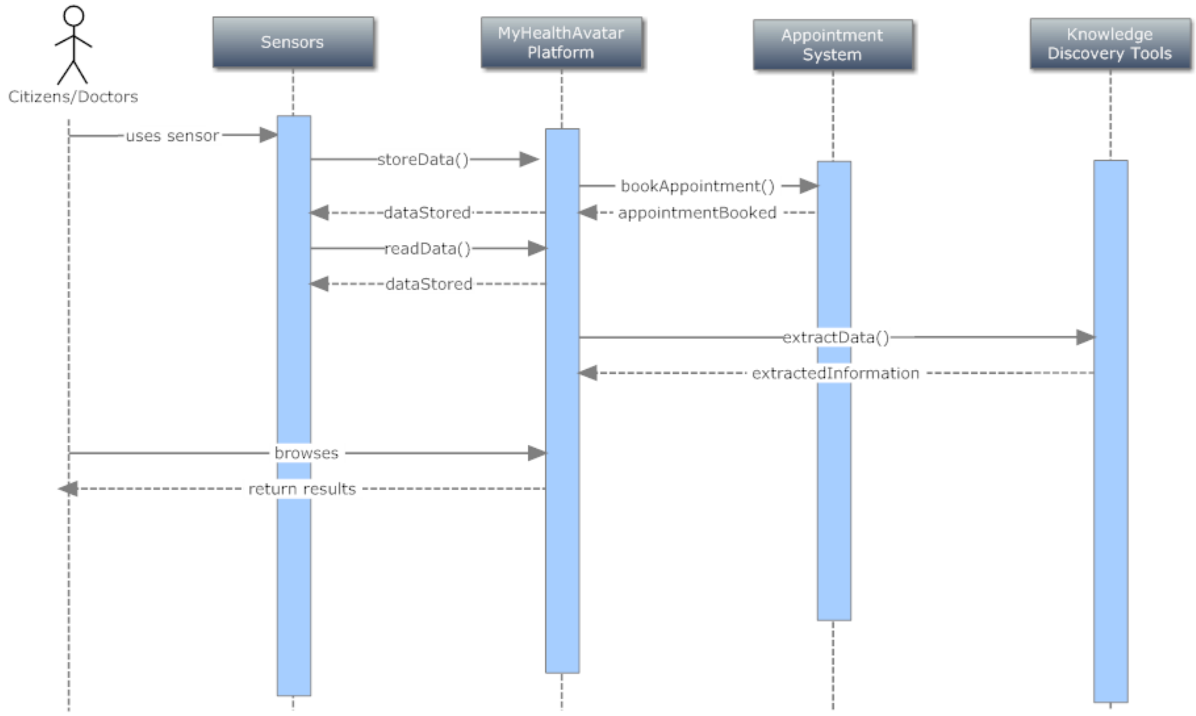

Figure 18. The sequence diagram for UC-6

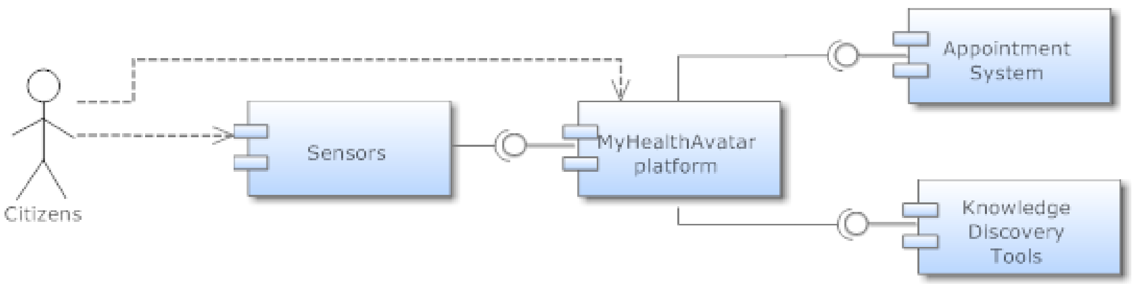

Figure 19. The component diagram for UC-6

Bellow we describe in detail the different components

Component Name	**Sensors**
Responsibilities	To allow actively engaging citizens in monitoring their own health through self-collection of life-logging data
Collaborators	MyHealthAvatar Platform
Rationale	We aim to create a symbiotic relationship of available technology today and MyHealthAvatar platform. Mobile applications, sensors etc. will monitor user's "health-status", "lifestyle" and "wellness" and upload data

	to the MyHealthAvatar system for close monitoring of health conditions and prevention of many diseases.
Issues and notes	Probably there should be defined a proper API in order to allow apps & sensors to interact with the MHA platform.

Component Name	**MyHealthAvatar Platform**
Responsibilities	To allow citizens and doctors log-in into the system, visualize their results etc.
Collaborators	Sensors, Appointment system, Knowledge discovery module
Rationale	There should be a component with a user friendly interface to allow interacting with the end-users
Issues and notes	Probably there should be defined a proper API in order to allow apps & sensors to interact with the MHA platform.

Component Name	**Appointment System**
Responsibilities	To allow booking an appointment with a doctor
Collaborators	MyHealthAvatar Platform
Rationale	The MyHealthAvatar system can be used for direct interaction between the patient and the physician. Such an interaction might provide the following functionalities: 1. Making appointments with the physician 2. Asking questions to the physician 3. Giving advice to the patient by the physician
Issues and notes	Both patients and doctors should use this module to book an appointment.

Component Name	**Knowledge discovery tools**
Responsibilities	To provide the most recent and personalized information about their disease, treatment and prognosis.
Collaborators	MyHealthAvatar Platform

Rationale	KD module could combine classical keyword-based search with text-mining and ontologies to navigate large results sets (internal & external) and facilitate information and/or knowledge discovery.
Issues and notes	An ontology-based Knowledge Discovery (KD) module should be used to connect highly heterogeneous fata and textual information.

4.3.2.1.1.7 UC-7 Link MyHealthAvatar to hospitals

This use case describes how MyHealthAvatar platform interacts with hospital information systems. The sequence diagram of this use case is shown on Figure 20 and the component diagram in Figure 21.

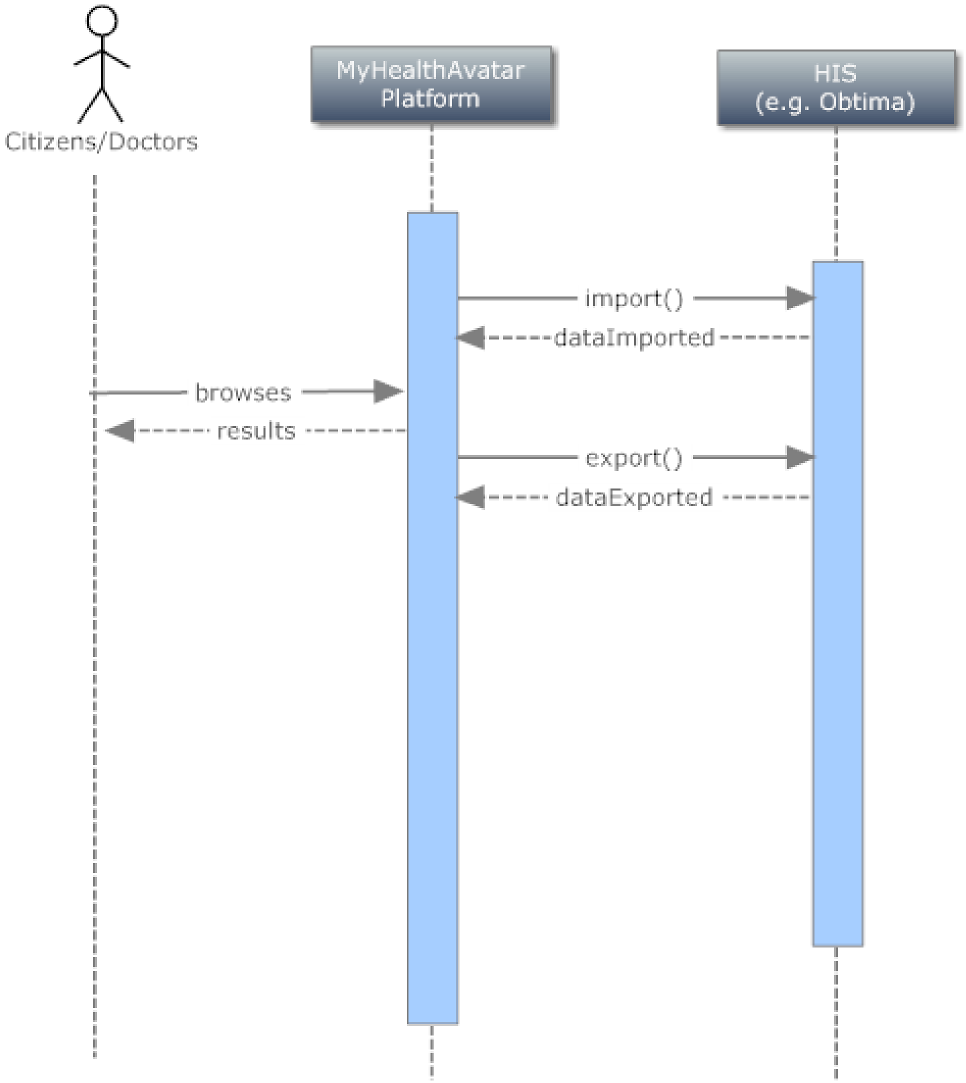

Figure 20. The sequence diagram for UC-7

Figure 21. The component diagram for UC-7

Bellow we describe in detail the different components.

Component Name	**MyHealthAvatar Platform**
Responsibilities	To allow citizens and doctors log-in into the system, visualize their results etc.
Collaborators	HIS
Rationale	There should be a component with a user friendly interface to allow interacting with the end-users
Issues and notes	Probably there should be defined a proper API in order to allow apps & sensors to interact with the MHA platform. Moreover ontology based export and import mechanisms should be devised.

Component Name	**HIS**
Responsibilities	To store the patient records in hospitals
Collaborators	MyHealthAvatar platform
Rationale	Each hospital has its own information system to store patient records.
Issues and notes	The Operational Data Model (ODM) could be exploited. It is designed to facilitate the archive and interchange of the metadata and data for clinical research, its power being fully unleashed when data are collected from multiple sources.

4.3.2.1.1.8 UC-8 Compile and perform a simulation using a biological model

This use case describes the GUIs, functionalities and tools in the frames of MyHealthAvatar platform to create and execute a biological simulation scenario. The sequence diagram is shown in Figure 22 whereas the component diagram in Figure 23.

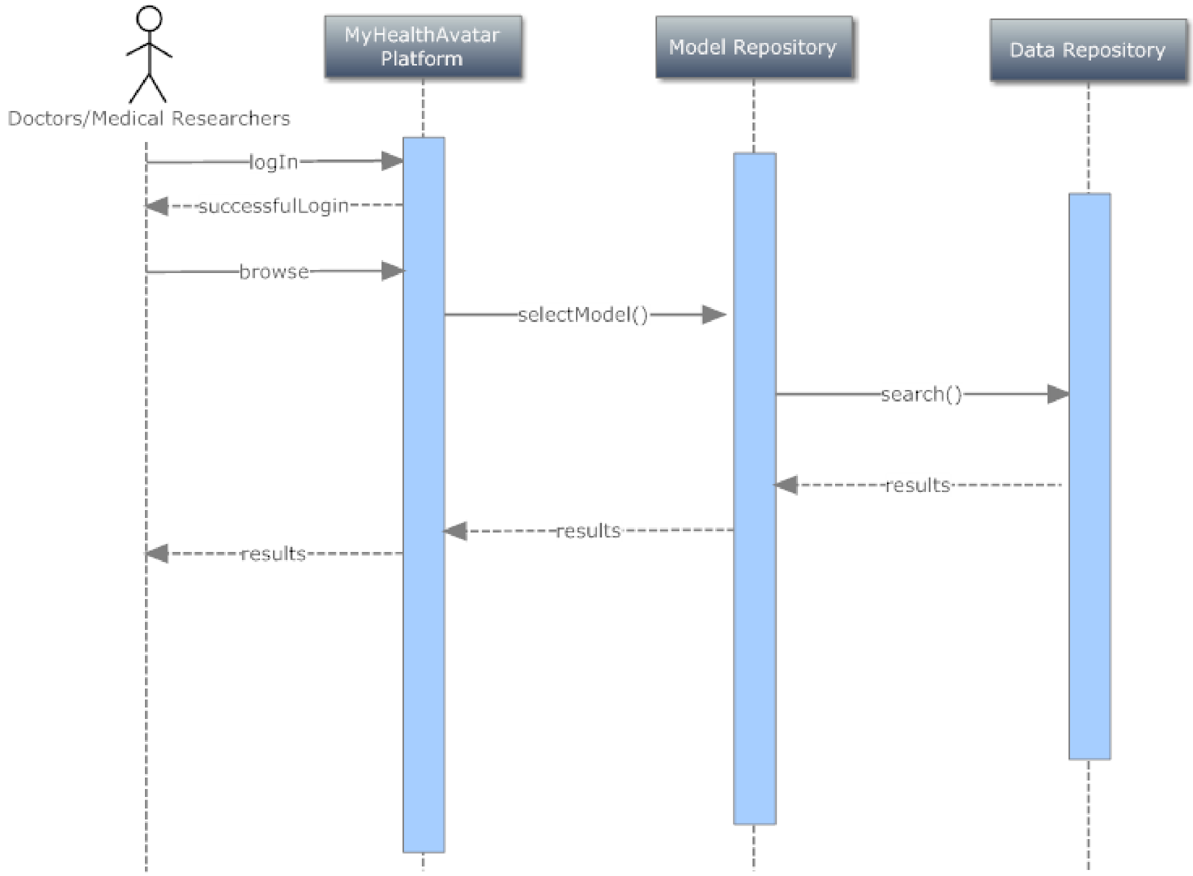

Figure 22. The sequence diagram of UC-8

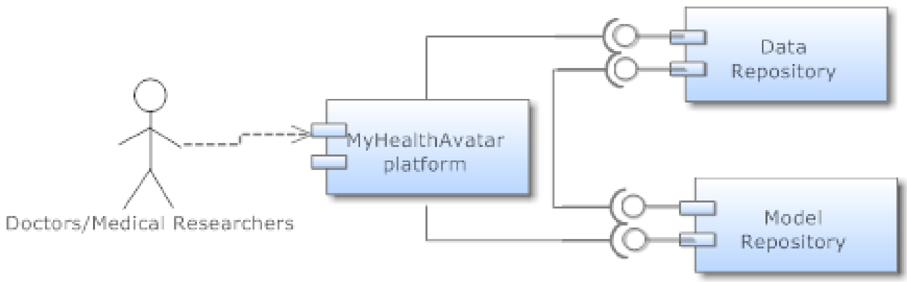

Figure 23. The component diagram of UC-8

The different components are being explained bellow:

Component Name	**MyHealthAvatar Platform**
Responsibilities	To allow citizens and doctors log-in into the system, visualize their results etc.

Collaborators	Data Repository, Model Repository
Rationale	There should be a component with a user friendly interface to allow interacting with the end-users
Issues and notes	No issues and notes

Component Name	**Data Repository**
Responsibilities	To allow the storage of data in a secure, distributed, highly accessible environment.
Collaborators	MyHealthAvatar Platform, Model repository
Rationale	There should be a central repository of data, that will allow further elaboration on the collected data
Issues and notes	Probably an ontology based data access mechanism should be established to allow access to heterogeneous data.

Component Name	**Model Repository**
Responsibilities	To allow the storage of biological models in a secure, distributed, highly accessible environment.
Collaborators	MyHealthAvatar Platform, Model repository
Rationale	There should be a central repository of models, that will allow further elaboration on the collected data
Issues and notes	There should be in place the proper facilities for these models to be executed.

4.3.2.1.2 Clinical Scenarios

4.3.2.1.2.1 UC-9 Personal Genomics

4.3.2.1.2.2 UC-10 Personal CHF

This scenario has to do with building a CHF related risk profile of a patient and using this profile to issue several alerts and recommendations. The sequence diagram of the aforementioned use case is shown in whereas the component diagram in

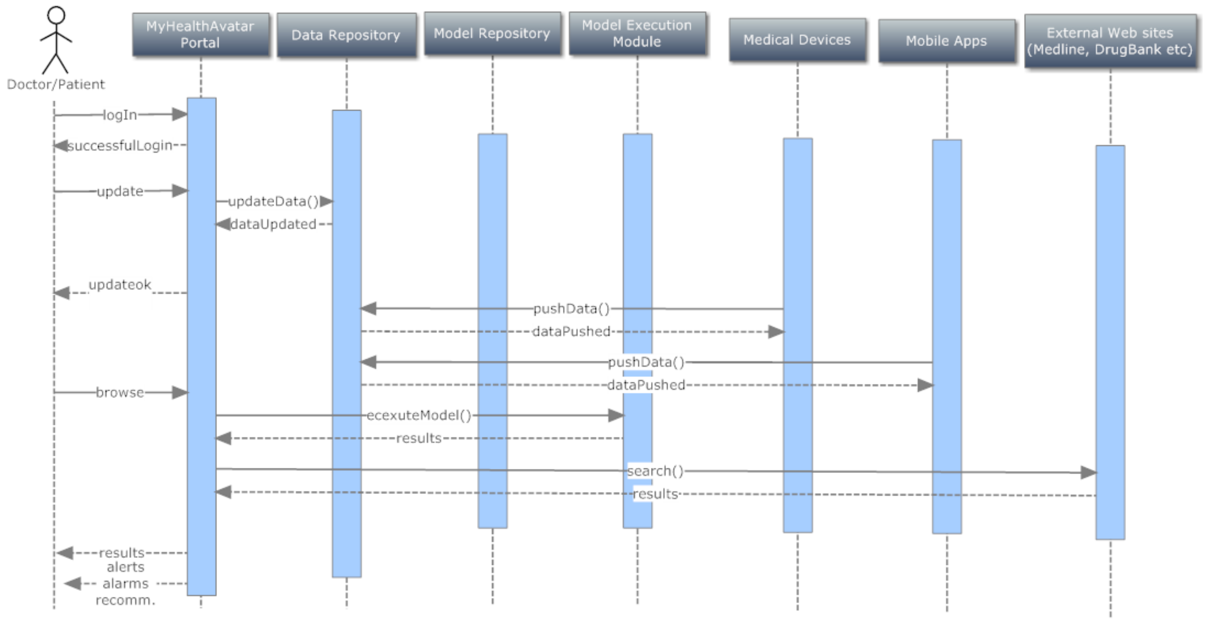

Figure 24. The sequence diagram of UC-10

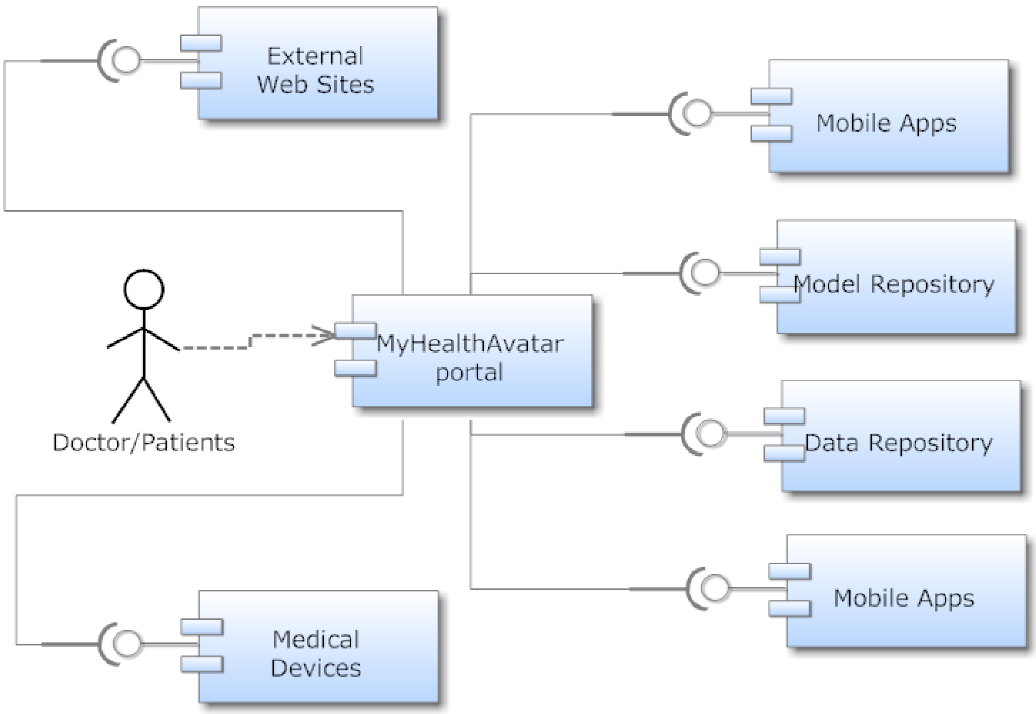

Figure 25. The component diagram of UC-10

The different components are explained bellow.

Component Name	**MyHealthAvatar Portal**

Responsibilities	To allow citizens and doctors log-in into the system, visualize their results etc.
Collaborators	Data Repository, Model Repository, Mobile Apps, Medical devices, External web sites
Rationale	There should be a component with a user friendly interface to allow interacting with the end-users
Issues and notes	No issues and notes

Component Name	**Data Repository**
Responsibilities	To allow the storage of data in a secure, distributed, highly accessible environment.
Collaborators	MyHealthAvatar Portal
Rationale	There should be a central repository of data, that will allow further elaboration on the collected data
Issues and notes	Probably an ontology based data access mechanism should be established to allow access to heterogeneous data.

Component Name	**Model Repository**
Responsibilities	To allow the storage of biological models in a secure, distributed, highly accessible environment.
Collaborators	MyHealthAvatar Portal
Rationale	There should be a central repository of models, that will allow further elaboration on the collected data
Issues and notes	There should be in place the proper facilities for these models to be executed.

Component Name	**Model Execution Module**
Responsibilities	To allow the execution of biological models

Collaborators	MyHealthAvatar Portal
Rationale	There should be a central place where the models will be executed.
Issues and notes	There should be in place the proper facilities for these models to be executed.

Component Name	**Medical Devices**
Responsibilities	To allow actively engaging citizens in monitoring their own health through self-collection of life-logging data.
Collaborators	MyHealthAvatar Portal
Rationale	We aim to create a symbiotic relationship of available technology today and MyHealthAvatar platform. Medical devices will monitor user's "health-status" and other vital signs (e.g. blood pressure, SpO2, Heart Rate, ECG)and upload data to the MyHealthAvatar system for close monitoring of health conditions and prevention of many diseases.
Issues and notes	Probably there should be defined a proper API in order to allow apps & sensors to interact with the MHA platform.

Component Name	**Mobile Apps**
Responsibilities	To allow actively engaging citizens in monitoring their own health through self-collection of life-logging data.
Collaborators	MyHealthAvatar Portal
Rationale	We aim to create a symbiotic relationship of available technology today and MyHealthAvatar platform. Mobile applications will monitor user's "health-status", "lifestyle" and "wellness" and upload data to the MyHealthAvatar system for close monitoring of health conditions and prevention of many diseases.
Issues and notes	Probably there should be defined a proper API in order to allow apps & sensors to interact with the MHA platform.

Component Name	**External Web Sites**

Responsibilities	To allow actively engaging citizens in monitoring their own health through self-collection of life-logging data
Collaborators	MyHealthAvatar Portal
Rationale	We aim to create a symbiotic relationship of available technology today and MyHealthAvatar platform. Mobile applications, sensors etc. will monitor user's "health-status", "lifestyle" and "wellness" and upload data to the MyHealthAvatar system for close monitoring of health conditions and prevention of many diseases.
Issues and notes	Probably there should be defined a proper API in order to allow apps & sensors to interact with the MHA platform.

4.3.2.1.2.3 UC-11 Antithrombotic therapy in preoperative period

This use case has to do with decision making tools regarding emergency situations in clinical practice. The example that is presented has to do with anti-platelet therapy in pre-operative patients. The sequence diagram is shown at whereas the component diagram is shown at.

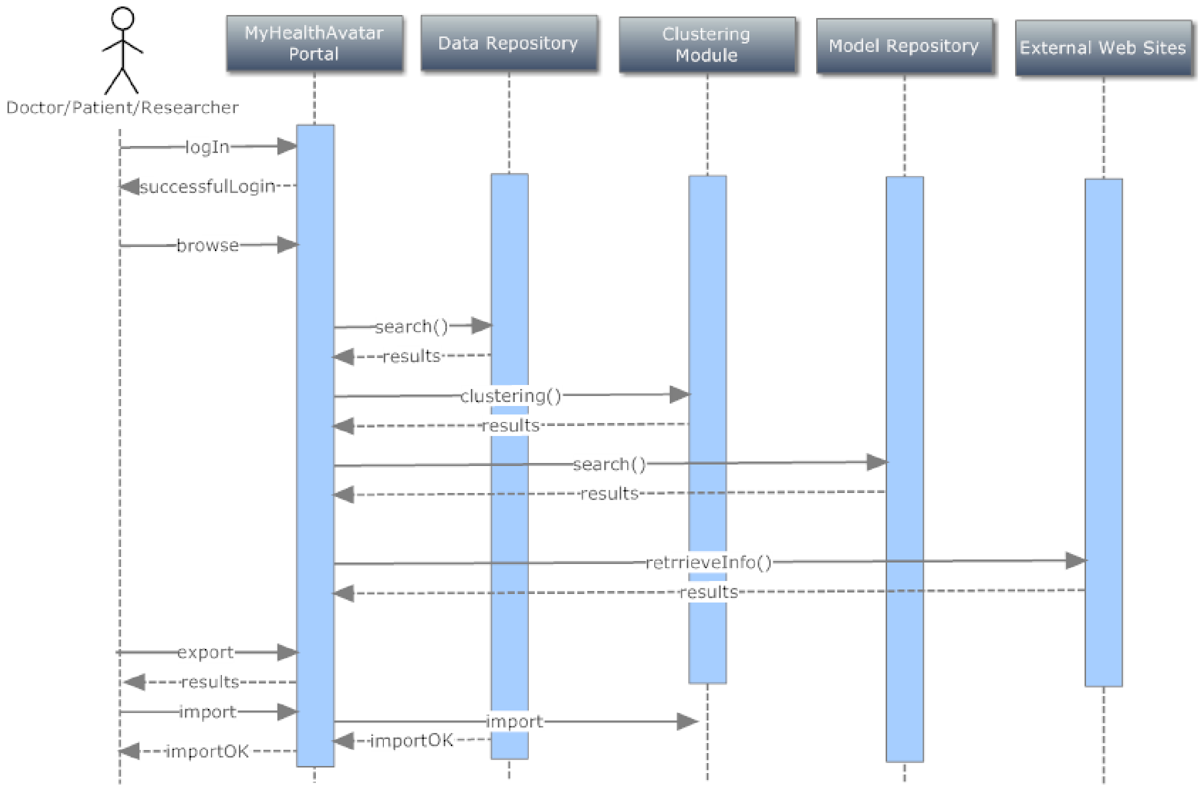

Figure 26. The sequence diagram of UC-11

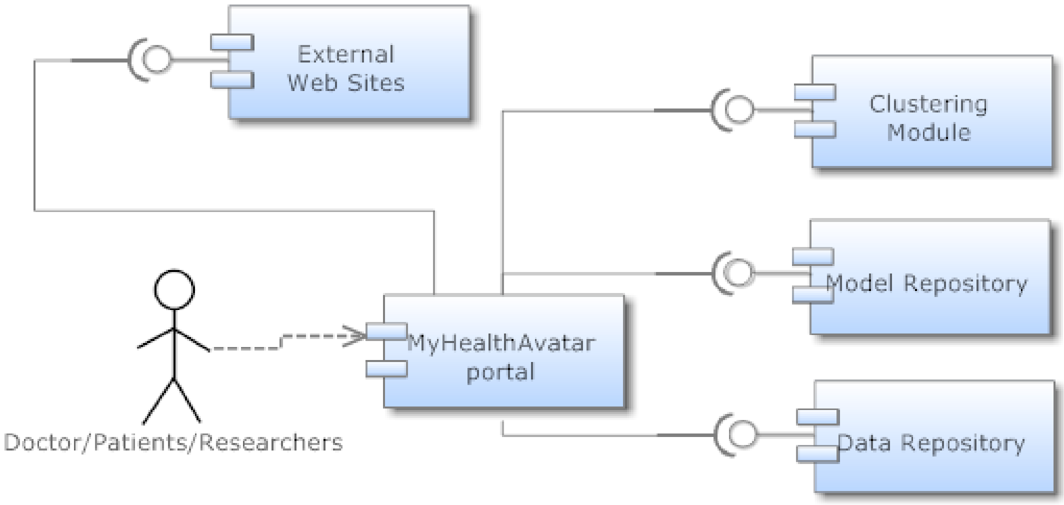

Figure 27. The component diagram of UC-11

Component Name	**MyHealthAvatar Portal**
Responsibilities	To allow citizens and doctors log-in into the system, visualize their results etc. Moreover it is the end-point for data-import and export in this use-case.
Collaborators	Data Repository, Model Repository, Personal Genome, Clustering Module, External Web Sites
Rationale	There should be a component with a user friendly interface to allow interacting with the end-users
Issues and notes	No issues and notes

Component Name	**Data Repository**
Responsibilities	To allow the storage of data in a secure, distributed, highly accessible environment.
Collaborators	MyHealthAvatar Portal
Rationale	There should be a central repository of data that will allow further elaboration on the collected data. We have to note that in this data repository data about medication, clinical conditions and personal genome should be recorded as well.
Issues and notes	Probably an ontology based data access mechanism should be established to allow access to heterogeneous data.

Component Name	**Model Repository**
Responsibilities	To allow the storage of biological models in a secure, distributed, highly accessible environment.
Collaborators	MyHealthAvatar Portal
Rationale	There should be a central repository of models, that will allow further elaboration on the collected data
Issues and notes	There should be in place the proper facilities for these models to be executed.

Component Name	**Clustering Module**
Responsibilities	To allow patient clustering into multiple cohorts
Collaborators	MyHealthAvatar Portal
Rationale	Clustering patients in appropriate cohorts and creation of “virtual population” based on demographic, physiology and genomic data The creation of “virtual population” is based in the distribution of the several parameters and the relation with each other (i.e. weight modeled against height) according to specific algorithms followed by a platform for in silico clinical trials Distribution of pharmacogenomics data in the population of patients
Issues and notes	There should be defined the proper mechanisms to load patient data.

Component Name	**External Web Sites**
Responsibilities	To allow actively engaging citizens in monitoring their own health through self-collection of life-logging data
Collaborators	MyHealthAvatar Portal
Rationale	We aim to create a symbiotic relationship of available technology today and MyHealthAvatar platform. Mobile applications, sensors etc. will monitor user’s “health-status”, “lifestyle” and “wellness” and upload data

	to the MyHealthAvatar system for close monitoring of health conditions and prevention of many diseases.
Issues and notes	Probably there should be defined a proper API in order to allow apps & sensors to interact with the MHA platform.

4.3.2.1.2.4 UC-26 Brain Trauma

This use case describes how the pre-injury clinical profile of a patient can help the clinicians to improve the clinical outcome. The sequence diagram of this use-case is shown on Figure 28 whereas the component diagram is shown in Figure 29.

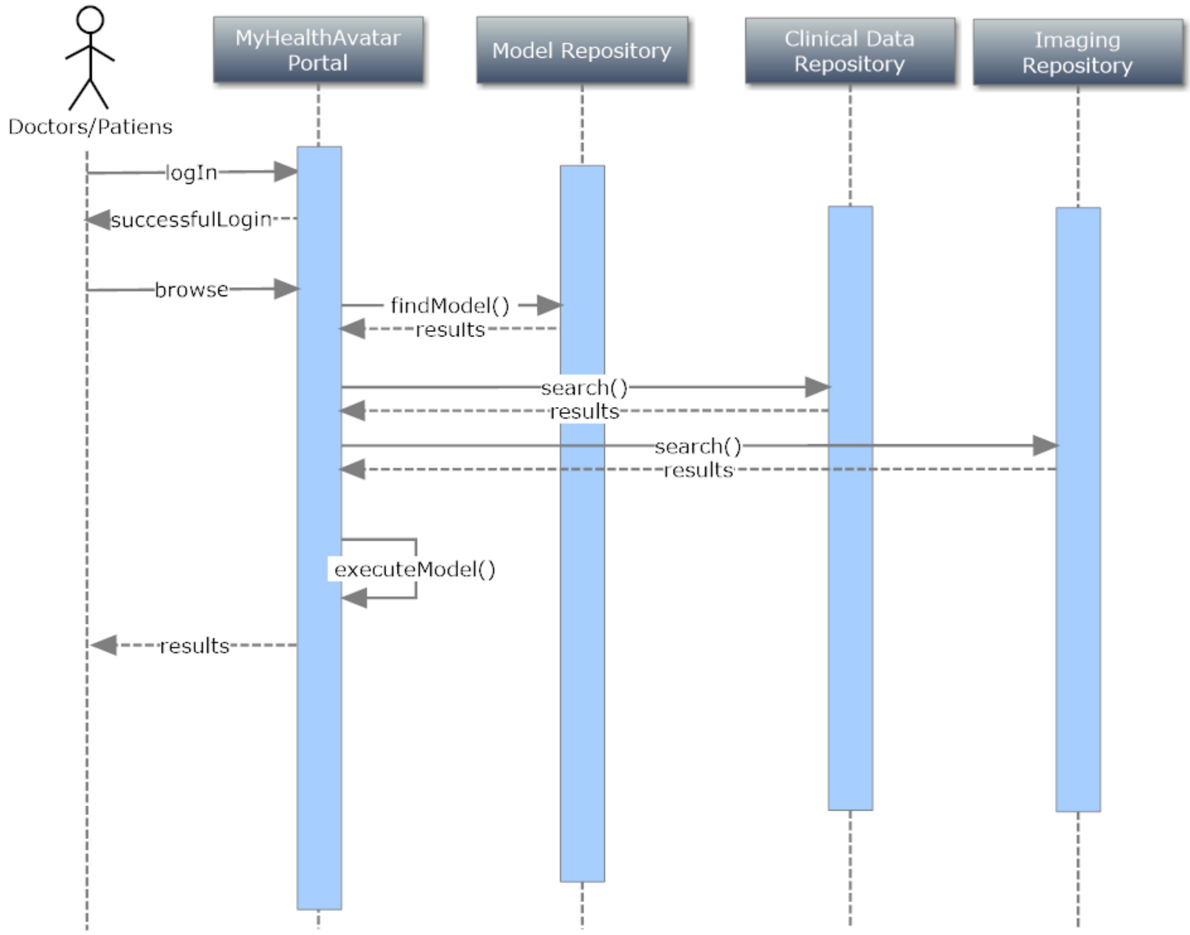

Figure 28. The sequence diagram of UC-26

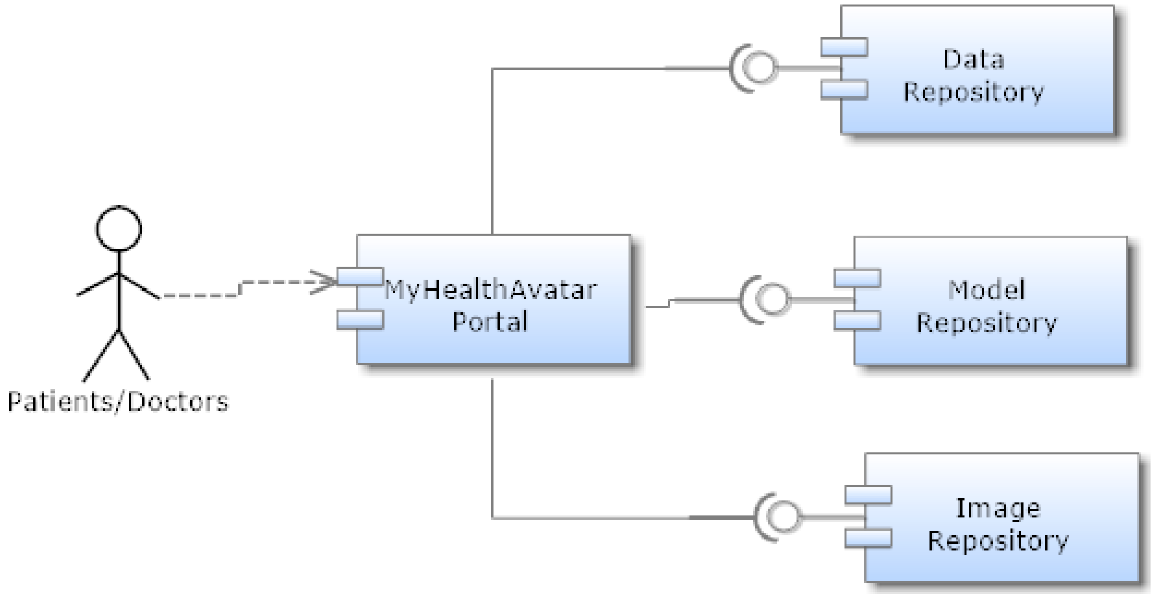

Figure 29. The component diagram of UC-26

The different components of this use case are described below:

Component Name	**MyHealthAvatar Portal**
Responsibilities	To allow citizens and doctors log-in into the system, visualize their results etc.
Collaborators	Data Repository, Model Repository, Image Repository
Rationale	There should be a component with a user friendly interface to allow interacting with the end-users
Issues and notes	No issues and notes

Component Name	**Data Repository**
Responsibilities	To allow the storage of data in a secure, distributed, highly accessible environment.
Collaborators	MyHealthAvatar Portal
Rationale	There should be a central repository of data, that will allow further elaboration on the collected data

Issues and notes	Probably an ontology based data access mechanism should be established to allow access to heterogeneous data.

Component Name	**Model Repository**
Responsibilities	To allow the storage of biological models in a secure, distributed, highly accessible environment.
Collaborators	MyHealthAvatar Platform
Rationale	There should be a central repository of models, that will allow further elaboration on the collected data
Issues and notes	There should be in place the proper facilities for these models to be executed.

Component Name	**Image Repository**
Responsibilities	To allow the storage of images in a secure, distributed, highly accessible environment.
Collaborators	MyHealthAvatar Portal
Rationale	There should be a central repository of images, that will allow further elaboration on the collected images
Issues and notes	Probably an imaging access mechanism should be established.

4.3.3 Deployment View

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

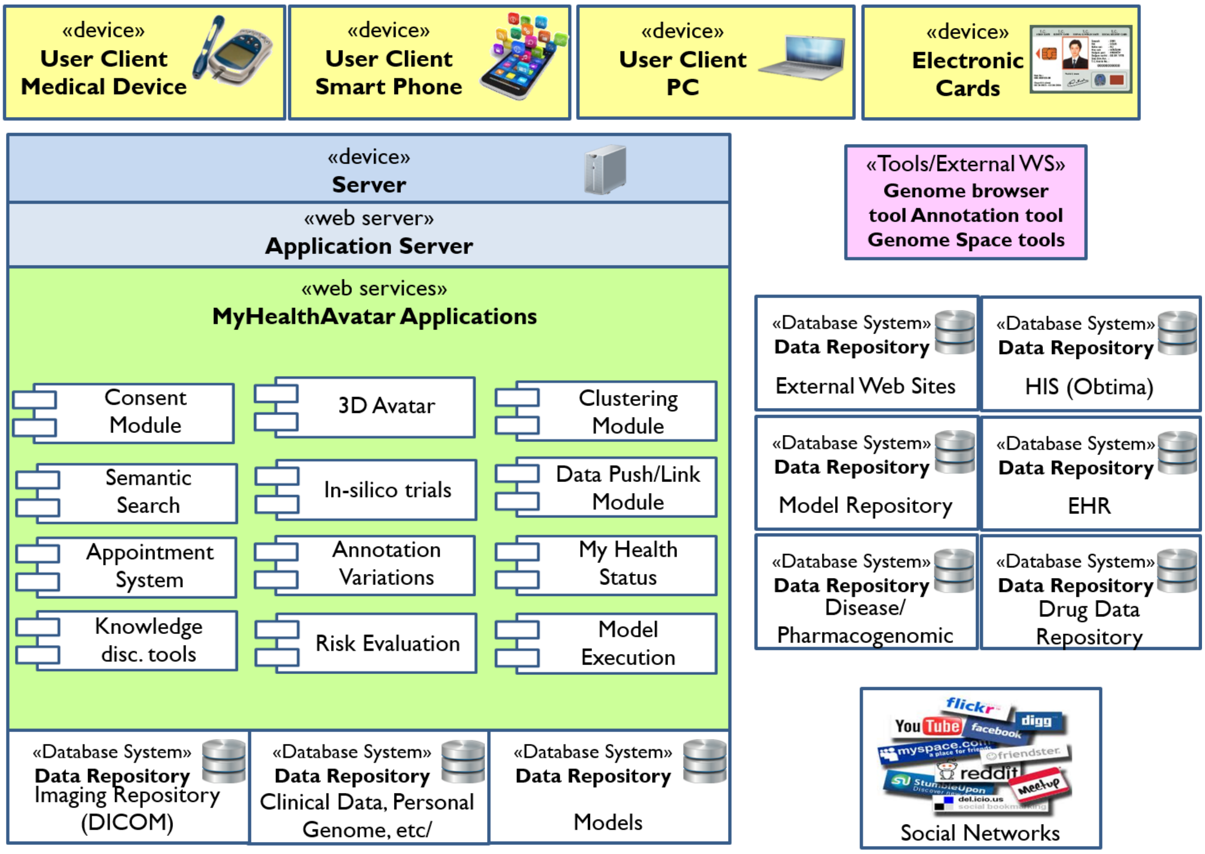

Figure 30 An initial deployment diagram for the system

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

- Medical/Clinical Data
- DICOM Images
- Genome data
- Pharmaceutical data
- Model data
- Generic files, i.e. “unstructured” (binary) data from the data warehouse point of view

The deployment aspects of this cloud infrastructure are described next.

4.3.4 Cloud infrastructure

MyHealthAvatar Cloud Storage System is the lowest level component in the data management architecture. 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.

The main approach is to work on a locally-deployed cloud infrastructure which can utilize local computing power as well as maintain the ability to outsource the infrastructure to commercial cloud computing facilities (e.g. Amazon EC2). This could be potentially used by the organizations that host the data as the architecture to perform data storage and analysis services to provide information for remote users without transferring the data out.

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.

4.3.5 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 organization, common processing that all modules must implement, any required standardization of design, coding and testing and the organization of the system’s code base. Our opinion is that in this stage of the 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.

4.3.5.1 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 and the adoption of the principles of the REST architectural style.
- The usage of LAMP/LAPP (Linux, Apache, MySQL/PostgreSQL, Perl/PHP/Python) solutions instead of proprietary server solutions.

4.3.6 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. This will be defined later in the project and will be reported in updated version of this deliverable

4.4 System Qualities

System qualities capture in essence the non-functional requirements of the entire platform. Usually these qualities are orthogonal to functionality and are observable properties of the system. They are usually "systemic" in the sense that there's not 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.

4.4.1 Performance and scalability

The requirements regarding performance and scalability have not yet specified. The specific requirement will be available together with the availability of the data in the data repository and with the definition of the queries that the users (clinicians and citizens) will do.

The computational requirement will also depend on where the simulation will take place. It is envisaged that the MyHealthAvatar platform will have access to simulation models in various model repositories by clinicians. A possible scenario is to have access to the computing service provided (or linked to) by the model repository in order to run these models, in which case no additional computing resources are required for the simulation.

In subsequent versions of the architecture we expect to be clarified in detail. Nevertheless, the system should provide a good quality of service and responsiveness

4.4.2 Security

Security is connected to all architectural layers. Ideally the MHA security solution will consist of re-usable modularized components that respectively deal with authentication, authorization, audit and privacy. Users interacting with MHA platform need to be authenticated (and authorized) before they can access the services and resources that the platform offers. More specifically the authentication requirement will be met by single sign on and single sign out mechanism. Moreover, one of the major requirements of the MHA platform is that access, to a particular service or resource, should only be provided to users that are allowed to see them. In the current iteration we only specify that the proper security layers and groups will be established and will be available by the MHA portal.

4.4.3 Location

There are no specific requirements for the geographical location for the platform to be installed.

4.4.4 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 MyHealthAvatar 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.

5 Conclusion

We have made a selection for the design process and proceeded to the identification of the major views following a strict approach in compliance with the terminology of the IEEE 1471 standard and current best practice. We have not delved deep in the details and the meticulous specification of all the possible aspects and characteristics of MyHealthAvatar architecture. The reason is that we have just initiated the process and we expect that to revisit and enhance this document in the course of the project. We plan to follow a more agile software development process, that 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 [Nuseibeh 2001] shown in the following figure. 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".

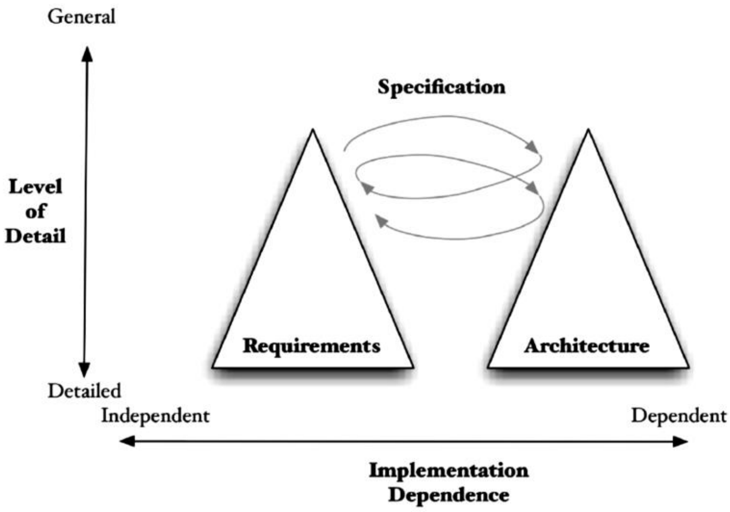

Figure 31 Architecture definition context

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 trade-offs 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 MyHealthAvatar. 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.

6 References

[Bass 1997] Bass L., Clements P. and Kazman R., Software Architecture in Practice, Addison-Wesley, 1997.

[IEEE 2000] 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

[Rozanski 2011] Rozanski, Nick, and Woods, Eoin. Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives, 2nd Edition. Addison Wesley, 2011.

[Fairbanks 2010] George H. Fairbanks, Just Enough Software Architecture: A Risk-Driven Approach Marshall & Brainerd. 2010

[Nuseibeh 2001] Nuseibeh, Bashar. “Weaving Together Requirements and Architectures.” IEEE Computer, 34(3):115–117, March 2001.

Appendix 1 – MyHealthAvatar Use Cases

A. Avatar system use cases

	Use Case	Responsible Partners	Clinical collaborator	End Users	Data
UC1	MyHealthAvatar User accounts	BED,FORTH, ICCS	USAAR	Patients/citizens, doctors/medical researchers	Data from internal Warehouse
UC2	3D avatar visualization	BED, FORTH	USAAR	Patients/citizens, doctors/medical researchers	Data from internal Warehouse (lab results, images etc.)
UC3	MyHealthAvatar data browse	BED, LIN, FORTH, ICCS	USAAR	Patients/citizens, doctors/medical researchers	Data from internal Warehouse (lab results, images etc.)
UC4	MyHealthAvatar virtual community	BED, FORTH	USAAR	Patients/citizens, doctors/medical researchers	Data from internal Warehouse (lab results, images etc.)
UC5	Self data collection for citizens	BED, FORTH	USAAR	Patients/citizens	
UC6	MyHealthAvatar Toolbox	BED, FORTH, ICCS	USAAR	Remote consultation: doctors (GPs) and patients Knowledge discovery: medical researchers, doctors	
UC7	Link MyHealthAvatar to hospitals	FORTH, ICCS	USAAR	Patients/citizens will see their own health records from the hospitals Doctors will be able to see patient data in their avatars	
UC8	Compile and perform a simulation using a biological model	ICCS	USAAR	Doctors, Medical Researchers	Clinical data (already preprocessed)
UC9	Utilization of personal genomic information for the individualization of MHA platform	FORTH	USAAR	Doctors, Medical Researchers, Citizens	Personal Genome, Health Records, Pharmacogenomics Profile

B. Clinical use cases

	Use Case	Responsible Partners	Clinical collaborator	End Users	Data
UC10	Building personalized CHF related risk profiles and “real-time monitoring” services	FORTH	USAAR	Doctors, Patients	Models, Genome, Clinical Data, Environmental Factors, Imaging Data, DrugBank, EHR data, Medscape, PharmKGB, PubMed
UC11	Decision making tools regarding emergency situations in clinical practice. The example of anti-platelet & anticoagulation therapy in the pre-operative patients	FORTH	USAAR	Doctors, Patients, Research Staff for in silico trials	Drug data,Population data, Clinical trials protocols and parameters
UC12	Brain Trauma	LIN	BED	Doctors and patients	Patients' data from MyHealthAvatar, Clinical and neurological examination data at the time of admission to hospital, Related studies on patient cohorts, such as IMPACT, CRASH

Appendix 2 – Abbreviations and acronyms

ADaM	Analysis Data Model CDISC standard supporting efficient generation, replication, and review of analysis results
AES	Advanced Encryption Standard a specification for the encryption of electronic data based on a design principle known as a Substitution permutation network
AGPL	Affero General Public License refers to two free software licenses. Affero General Public License, Version 1 and GNU Affero General Public License, version 3.
API	application programming interface is a particular set of rules ('code') and specifications that software programs can follow to communicate with each other
ASCII	American Standard Code for Information Interchange a character-encoding scheme based on the ordering of the English alphabet
BMP	Bitmap a raster graphics image file format used to store bitmap digital images
BSD	Berkeley Software Distribution is a Unix operating system derivative developed and distributed by the Computer Systems Research Group (CSRG) of the University of California, Berkeley
CAS	Central Authentication Service a single sign-on web protocol
CA	Certification Authority an entity that issues digital certificates
CBC	Cipher-Block Chaining a cryptographic mode of operation in which each block of plaintext is XORed with the previous ciphertext block before being encrypted
CCM	Counter with CBC-MAC Mode a mode of operation for cryptographic block ciphers
CCZero	Creative Commons licenses are several copyright licenses that allow the distribution of copyrighted works
CDASH	Clinical Data Acquisition Standards Harmonization CDISC standard describing the basic recommended (minimal) data collection fields for 18 domains, including common header fields, and demographic, adverse events, and other safety domains that are common to all therapeutic areas and phases of clinical research
CDA	Clinical Document Architecture is an XML-based markup standard intended to specify the encoding, structure and semantics of clinical documents for exchange
CDE	Clinical Document Architecture an XML-based markup standard defined by HL7 intended to specify the encoding, structure and semantics of clinical documents for exchange
CDISC	Clinical Data Interchange Standards Consortium - a global, open, multidisciplinary, non-profit organization that has established standards to support the acquisition, exchange, submission and archive of clinical research data and metadata
CDMI	Cloud Data Management Interface - defines a functional interface that applications can use to create, retrieve, update and delete data elements from the Cloud
CDS	Clinical Decision Support decision support software designed to assist physicians and other health professionals with decision making tasks, as determining diagnosis of patient data
CMWG	Cloud Management Work Group focused on standardizing interactions between cloud environments by developing specifications that deliver architectural semantics

	and implementation details to achieve interoperable cloud management between service providers and their consumers and developers
CRISP-DM	Cross Industry Standard Process for Data Mining a data mining process model that describes commonly used approaches that expert data miners use to tackle problems
CRL	Certificate Revocation List a list of certificates that have been revoked, and therefore should not be relied upon
CSS	Cascading Style Sheets is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a markup language
CSV	Comma-Separated Values a set of file formats used to store tabular data in which numbers and text are stored in plain-text form that can be easily written and read in a text editor
CWM	Common Warehouse Metamodel a specification for modeling metadata for relational, non-relational, multi-dimensional, and most other objects found in a data warehousing environment
CeCILL	CEA CNRS INRIA Logiciel Libre is a free software license adapted to both international and French legal matters, in the spirit of and retaining compatibility with the GNU General Public License
CellML	Cell Markup Language is an XML based markup language for describing mathematical models
DICOM	Digital Imaging and Communications in Medicine - a standard for handling, storing, printing, and transmitting information in medical imaging
DTMF	Distributed Management Task Force brings the IT industry together to collaborate on the development, validation and promotion of systems management standards
EHR	Electronic health record is an evolving concept defined as a systematic collection of electronic health information about individual patients or populations
EULA	End-user licensing agreements An EULA is a legal contract between the manufacturer and/or the author and the end user of an application
EUPL	European Union Public License the first European Free/Open Source Software (F/OSS) license
FMA F	Foundational Model of Anatomy it is concerned with the representation of classes or types and relationships necessary for the symbolic representation of the phenotypic structure of the human body in a form that is understandable to humans and is also navigable, parseable and interpretable by machine-based systems
FieldML	Field Markup Language is an XML based markup language for describing field models
GAS	Gridge Authorization Service provides functionality that would be able to fulfill most authorization requirements of grid computing environments
GCM	Galois/Counter Mode a mode of operation for symmetric key cryptographic block ciphers that has been widely adopted because of its efficiency and performance
GEM	Guideline Elements Model an XML-based guideline document model that can store and organize the heterogeneous information contained in practice guidelines

GNU	Gnu's Not Unix is a Unix-like computer operating system developed by the GNU project, ultimately aiming to be a "complete Unix-compatible software system" composed wholly of free software.
GO	Gene Ontology is a major bioinformatics initiative with the aim of standardizing the representation of gene and gene product attributes across species and databases
GPL	General Public License is the most widely used free software license, originally written by Richard Stallman for the GNU Project
GridFTP	GridFTP is an extension of the standard File Transfer Protocol (FTP) for use with Grid computing
HL7	Health Level Seven is an all-volunteer, non-profit organization involved in development of international healthcare informatics interoperability standards
HMAC	Hash-based Message Authentication Code a mechanism for message authentication using cryptographic hash functions
HTML	Hypertext Markup Language is the predominant markup language for web pages. HTML elements are the basic building-blocks of webpages.
HTTPS	Hypertext Transfer Protocol Secure is a combination of the Hypertext Transfer Protocol (HTTP) with SSL/TLS protocol to provide encrypted communication and secure identification of a network web server
IBM	International Business Machines
ID-FF	Liberty Identity Federation Framework an approach for implementing a single sign-on with federated identities based on commonly deployed technologies
ID-WSF	Liberty Identity Web Services Framework a framework for identity-based web services in a federated network identity environment
IEC	International Electrotechnical Commission is the world's leading organization that prepares and publishes International Standards for all electrical, electronic and related technologies
IEEE	Institute of Electrical and Electronics Engineers is a non-profit professional association headquartered in the United States that is dedicated to advancing technological innovation and excellence
IETF	Internet Engineering Task Force a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet
IHE	Integrating the Healthcare Enterprise - an initiative by healthcare professionals and industry to improve the way computer systems in healthcare share information
IPSec	Internet Protocol Security a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a communication session
ISBN	International Standard Book Number is a unique numeric commercial book identifier based upon the 9-digit Standard Book Numbering (SBN) code created by Gordon Foster
ISO	International Organization for Standardization is an international standard-setting body composed of representatives from various national standards organizations

InSilicoML	InSilico Markup Language is a markup language that can explicitly describe the multi-level hierarchical structures of the physiological functions in mathematical models
JDMP	Java Data Mining Package an open source Java library for data analysis and machine learning
JPEG	Joint Photographic Experts Group is a commonly used method of lossy compression for digital photography
JSDL	Job Submission Description Language is an extensible XML specification from the Global Grid Forum for the description of simple tasks to non-interactive computer execution systems
KNIME	Konstanz Information Miner a user-friendly and comprehensive open source data integration, process, analysis and exploration platform
LGPL	Lesser General Public License is a free software license published by the Free Software Foundation
LOINC	Logical Observation Identifiers Names and Codes is a database and universal standard for identifying medical laboratory observations
MAGE-ML	Microarray and Gene Expression - Markup Language markup language format for the representation of gene expression data from microarrays to facilitate the exchange of information between different data systems
MAGE-OM	Microarray and Gene Expression - Object Model data exchange model for the representation of gene expression data from microarrays to facilitate the exchange of information between different data systems
MAGE-TAB	Microarray and Gene Expression - Tabular tabular format for the representation of gene expression data from microarrays to facilitate the exchange of information between different data systems
MIAME	Minimum Information About a Microarray Experiment needed to enable the interpretation of the results of the experiment unambiguously and potentially to reproduce the experiment
MIASE	Minimal Information About a Simulation Experiment common set of information a modeller needs to provide in order to enable the execution and reproduction of a numerical simulation experiment, derived from a given set of quantitative models
MIASE	Minimum Information About a Simulation Experiment is an effort to list the common set of information a modeller needs to provide in order to enable the execution and reproduction of a numerical simulation experiment, derived from a given set of quantitative models.
MIBBI	Minimum Information for Biological and Biomedical Investigations maintains a web-based, freely accessible resource for "Minimum Information" checklist projects, providing straightforward access to extant checklists (and to complementary data formats, controlled vocabularies, tools and databases), thereby enhancing both transparency and accessibility
MIT	MIT License is a free software license originating at the Massachusetts Institute of Technology
ML	Markup Language is a modern system for annotating a text in a way that is syntactically distinguishable from that text

MOF	MetaObject Facility the foundation of OMG's industry-standard environment where models can be exported from one application, imported into another, transported across a network, stored in a repository and then retrieved, rendered into different formats
MPL	Mozilla Public License is a free and open source software license
MS	Microsoft is an American public multinational corporation headquartered in Redmond, Washington
MTOM	Message Transmission Optimization Mechanism is the W3C Message Transmission Optimization Mechanism, a method of efficiently sending binary data to and from Web services
MedLEE	Medical Language Extraction and Encoding system System to extract, structure, and encode clinical information in textual patient reports so that the data can be used by subsequent automated processes
NeuroML	Neuro Markup Language is an XML (Extensible Markup Language) based model description language that aims to provide a common data format for defining and exchanging models in computational neuroscience
OASIS	Organization for the Advancement of Structured Information Standards a not-for-profit consortium that drives the development, convergence and adoption of open standards for the global information society
OBO	Open Biomedical Ontologies is an effort to create controlled vocabularies for shared use across different biological and medical domains
OGSA-BES	Open Grid Services Architecture - Basic Execution Services defines Web Services interfaces for creating, monitoring, and controlling computational entities such as UNIX or Windows processes, Web Services, or parallel programswhat we call activities within a defined environment
OGSA-DAI	Open Grid Service Architecture-Data Access and Integration allows data resources (e.g. relational or XML databases, files or web services) to be federated and accessed via web services on the web or within grids or clouds. Via these web services, data can be queried, updated, transformed and combined in various ways.
OSI	Open Source Initiative is an organization dedicated to promoting open source software
OS	Operating System is a set of programs that manages computer hardware resources, and provides common services for application software
OWL-S	Ontology Web Language for web Services an ontology of services to discover, invoke, compose, and monitor Web resources offering particular services and having particular properties
OWL Web	Ontology Language is a family of knowledge representation languages for authoring ontologies.
OpenID	Open Identity provider of web-based SSO services
PAOS	Reverse HTTP Binding for SOAP a binding that enables HTTP clients to expose services using the SOAP protocol, where a SOAP request is bound to a HTTP response and vice versa

PATO	PATO an ontology of phenotypic qualities, intended for use in a number of applications, primarily defining composite phenotypes and phenotype annotation.
PHP	PHP: Hypertext Preprocessor is a general-purpose server-side scripting language originally designed for web development to produce dynamic web pages
PKIX	Public-Key Infrastructure Working Group was established in the fall of 1995 with the goal of developing Internet standards to support X.509-based Public Key Infrastructures
PMML	Predictive Model Markup Language an XML-based language which provides a way for applications to define statistical and data mining models and to share models between PMML compliant applications
PNG	Portable Network Graphics is a bitmapped image format that employs lossless data compression
POST	POST is one of many request methods supported by the HTTP protocol used by the World Wide Web
RAD	Rapid application development is a software development methodology that uses minimal planning in favor of rapid prototyping
RDF	Resource Description Framework is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata data model
REST	Representational state transfer is a style of software architecture for distributed hypermedia systems such as the World Wide Web
RFC	Request for Comments is a memorandum published by the Internet Engineering Task Force (IETF) describing methods, behaviors, research, or innovations applicable to the working of the Internet and Internet-connected systems
RICORDO	RICORDO is focused on the study and design of a multiscale ontological framework in support of the Virtual Physiological Human community to improve the interoperability amongst its Data and Modelling resources
RIM	Reference Information Model is the cornerstone of the HL7 Version 3 development process and an essential part of the HL7 V3 development methodology
SAML	Security Assertion Markup Language a standard, XML-based framework for creating and exchanging security information between online partners
SAS	Business analytics software and service developer, and independent vendor in the business intelligence market
SAWSDL	Semantic Annotations for WSDL defines mechanisms using which semantic annotations can be added to WSDL components
SBML	System Biology Markup Language is a representation format, based on XML, for communicating and storing computational models of biological processes
SDTM	Study Data Tabulation Model CDISC defining a standard structure for human clinical trial (study) data tabulations that are to be submitted as part of a product application to a regulatory authority
SED-ML	Simulation Experiment Description Markup Language an XML-based format for encoding simulation experiments, following the requirements defined in the MIASE guidelines

SHA	Secure Hash Algorithm a number of cryptographic hash functions published by the National Institute of Standards and Technology as a U.S. Federal Information Processing Standard
SLO	Single Log-Out termination of a SSO action
SNIA	Storage Networking Industry Association not-for-profit trade organization for companies and individuals in various sectors of the storage industry
SNOMED-CT	Systematized Nomenclature of Medicine - Clinical Term is a systematically organised computer processable collection of medical terminology covering most areas of clinical information such as diseases, findings, procedures, microorganisms, pharmaceuticals etc
SOAP	Simple Object Access Protocol is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks
SOAP	Simple Object Access Protocol a lightweight XML-based protocol for exchange of structured information in a decentralized, distributed environment
SOA	Service-Oriented Architecture s a set of principles and methodologies for designing and developing software in the form of interoperable services
SPARQL	SPARQL Protocol and RDF Query Language - query language for RDF
SQL	Structured Query Language a standard language for accessing and manipulating databases
SSH	Secure Shell is a network protocol for secure data communication, remote shell services or command execution and other secure network services between two networked computers that it connects via a secure channel over an insecure network
SSL	Secure Sockets Layer a cryptographic protocol that provides communication security over the Internet, predecessor of TLS
SSO	Single Sign-On a mechanism whereby a single action of user authentication and authorization can permit a user to access all computers and systems where he has access permission, without the need to enter multiple passwords
TCP/IP	Transmission Control Protocol/Internet Protocol the first two networking protocols defined in the Internet Protocol Suite standard
TDD	Test-driven development is a software development process that relies on the repetition of a very short development cycle.
TLS	Transport Layer Security a cryptographic protocol that provides communication security over the Internet, successor of SSL
UML	Unified Modelling Language a specification defining a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems
VDM	Vienna Development Method is one of the longest-established Formal Methods for the development of computer-based systems
VPH-NoE	Virtual Physiological Human - Network of Excellence is a project which aims to help support and progress European research in biomedical modelling and simulation of the human body

VPH	Virtual Physiological Human is a methodological and technological framework that, once established, will enable collaborative investigation of the human body as a single complex system
WAV	Waveform Audio File Format is a Microsoft and IBM audio file format standard for storing an audio bitstream on PCs
WS-*	Web Services-* common prefix for the family of Web Services specifications
WSDL	Web Services Description Language a way to describe the abstract functionalities of a service and concretely how and where to invoke it
WSMO	Web Service Modelling Ontology ontology for describing Semantic Web Services
XFree86	A freely redistributable open-source implementation of the X Window System
XHTML	Extensible HyperText Markup Language is a family of XML markup languages that mirror or extend versions of the widely-used Hypertext Markup Language (HTML), the language in which web pages are written
XML	Extensible Markup Language - a format for encoding documents in machine-readable form, similar in syntax to HTML
XTS	XEX-based Tweaked Codebook a mode of operation for cryptographic block ciphers
caBIG	Cancer Biomedical Informatics Grid a virtual network of interconnected data, individuals, and organizations that work together to redefine how cancer research is conducted