HEARTFAID

D20 – Clinical standards and first middleware prototype

Submission date: 16/11/2007
Due date of document: 31/07/2007

HEARTFAID

A KNOWLEDGE BASED PLATFORM OF SERVICES FOR SUPPORTING MEDICAL-CLINICAL MANAGEMENT OF THE HEART FAILURE WITHIN THE ELDERLY POPULATION

Project summary	
Project acronym:	HEARTFAID
Project identifier:	IST – 2005 – 027107
Duration of the Project:	01/02/2006 – 31/01/2009
Project Co-ordinator:	UNICAL University of Calabria (Italy)
Thematic Priority:	Information Society Technology
Instrument:	Specific Targeted Research or Innovation Project

Consortium

- UNICAL- Università della Calabria (Italy)
- UNICZ- Università degli studi Magna Graecia di Catanzaro (Italy)
- UNIMIB- Università degli studi di Milano Bicocca (Italy)
- JUMC- Jagiellonian University Medical College (Poland)
- VMWS- Virtual Medical World Solutions Ltd (United Kingdom)
- FORTHNET S. A.- Hellenic Telecommunications and Telematic Applications Company S. A. (Greece)
- SYNAP- Synapsis s.r.l. (Italy)
- CNR- Consiglio Nazionale delle Ricerche (Italy)
- FORTH-Foundation for Research and Technology Hellas (Greece)
- RBI- Rudjer Boskovic Institute (Croatia)
- AUXOL- Istituto Auxologico Italiano (Italy)

Document summary	
Document title:	D20 – Clinical standards and first middleware prototype
Document classification:	Derivable D20
Dissemination level:	CO
Submission date:	September 15th, 2007
Due date:	July 31st, 2007
Authors:	Christos Biniaris – VMW Manolis Stratakis, Stelios Louloudakis – FORTHNET Davide Guerri, Sergio Di Bona, Marco Lettere – SYNAP Ovidio Salvetti, Davide Moroni – CNR Franco Chiarugi – FORTH
Work package:	WP3 – Middleware, Interoperability and Integration
Report version:	1.2

Short description

Change record		
Version number	Changes	Release date
0.1	first draft of the document	29/06/2007
0.2	Second draft	16/07/2007
1.0	Advanced version	10/10/2007
1.3	Final version	15/10/2007

Table of Contents

Executive Summary

The scope of this document is to describe the work that has been carried out in the context of “WP3 – Middleware, Interoperability and Integration” of the HEARTFAID project to identify the standards adopted for integrating clinical applications and to deliver a sound architecture, implementation guidelines and some concrete use cases of the Heartfaid Integration Middleware.

the design and the implementation of a middleware for an interoperable platform of services operating in the eHealth sector could not leave out of consideration the adoption of standards which are formally considered as reference standards for health applications at international level, as well as commonly adopted by the actors of this sector.

OpenEHR, IHE, HL7 and MML are the main actors in clinical information integration whilst DICOM is surely the most widespread format for representing, storing and exchanging imaging and media related data.

Common and well-known communication standards like Bluetooth, TCP/IP and HTTP have been adopted for implementing the interconnections between the medical devices and the Heartfaid systems and among the different software agents of the system itself. These have been designed in such a way to guarantee scalability and flexibility because of the large number of heterogeneous actors involved in the overall implementation of the platform.

The architectural and the technological approach is based on state of the art ideas, solutions and patterns of Enterprise Integration such as *Message Oriented Middleware*, *Enterprise Portals*, *Service Oriented Architectures* and *Enterprise Services Buses.*

The soundness of the proposed architecture is proven by presenting realistic use cases that will be supported in the Heartfaid project and that are mainly focused on:

- User authentication and profiling
- Patient enrolment and consistent identification
- Reporting of clinical, observational and demographic information
- Automatic decision support for alarming and alerting
- Handling of imaging data.

1. Glossary of terms

TERM	DEFINITION
XML	eXtensible Markup Language
MoM	Message oriented Middleware
ESB	Enterprise Service Bus
JMS	Java Messaging Service
OM	Observation and Measurements
OMF	Open Messaging Framework
SensorML	Sensor Markup Language
OGC	OpenGeospatial Consosrtium
SOAP	Simple Object Application Protocol
HTTP	HyperText Transport Protocol
TCP/IP	Transport Connection Protocol/Internet Protocol
CBR	Content Based Routing
IBR	Itinerary Based Routing
MPI	Master Patient Index
HIS	Hospital Information System
RFID	Radio Frequency Identification
PIX	Patient Identity Cross-reference
XDS	Cross-Enterprise Document Sharing
DICOM	Digital Imaging and Communications
EHR	Electronic Health Record
IHE	Integrating the Healthcare Enterprise
RID	Retrieve Information for Display
XDS	Cross Enterprise Document Sharing
CDA	Clinical Document Architecture
HL7	Health Care Level 7
MML	Medical Markup Language
ECG	ElectroCardioGram
ACR	American College of Radiology
GPRS	General Packet Radio Service
LOINC	Logical Observation Identifiers Names and Codes
SNOMED	Systematized NOmenclature of MEDicine Clinical Terms

2. Introduction

This document will describe the activity carried out in the context of "WP3 – Middleware, Interoperability and Integration" concerning the identification of both clinical and ICT standards for the interoperability of the clinical applications within the HEARTFAID platform of services, and the design of a sound architecture for the implementation of platform itself. In addition, specific procedural guidelines have been accurately selected with the contribution of the clinical partners and implemented, as well as some realistic use cases, adequately identified by the technical partners, that should be supported by the Heartfaid Integration Middleware.

The first efforts have been devoted to the study and selection of available interoperability standards and approaches. In the section 3 of the document the standards selected as reference solutions for the HEARTFAID platform will be described.

In particular, we'll focus on the following clinical standards:

- **HL7** (Health Level Seven), an ANSI-accredited Standards Development Organization defined to allow two of more systems or components, operating in the healthcare sector, to "exchange information" and to "use the information" that has been exchanged. The name comes from "Health Level 7", which refers to the top layer (Level 7) of the Open Systems Interconnection (OSI) layer protocol for the health environment.
- **CDA** (Clinical Document Architecture) an HL7 standard for the representation of clinical documents; it is an XML-based document markup standard that specifies the structure and semantics of clinical documents for the purpose of exchange.
- **DICOM** (Digital Imaging and COmmunications in Medicine), a global Information-Technology standard used in virtually all hospitals worldwide and designed to ensure the interoperability of systems used to manage (produce, store, display, send, retrieve etc) medical images (with its derived structured documents) and the related workflow.
- **IHE** (Integrating Healthcare Enterprise), an initiative by healthcare professionals and industry to improve the way computer systems in healthcare share information. IHE promotes the coordinates use of established standards such as DICOM and HL7 to address specific clinical need in support of optimal patient care. Systems developed in accordance with IHE communicate with one another better, are easier to implement, and enable care providers to use information more effectively.

In a integrated but, however, very heterogenous architecture such as the one proposed by the Heartfaid project, the communication and the information exchange among all the involved actors is fundamental. Section 3.2 shows a brief description of the standards whose adoption has been planned by the Heartfaid developing team (Bluetooth, TCP/IP and HTTP). In section 3.3 and 3.4 the document stresses the importance of **clinical integration profiles** for homogeneizing not only data but also functional processes and workflows (patient identification, information display...) and **imaging standards (DICOM)** for exchanging image based information enriched with structured and possibly anonymized metada.

Section 4 gives a detailed overview of the design of the first prototype of the integration Middleware. The main concepts in terms of design and technologies are shown and a survey of the adopted solutions, patterns and best practices is elaborated. It is important to note that both design and implementation techniques will be derived from an in-depth study of computer science literature that represents the state of the art in arguments like *Message Oriented Middleware, Service Oriented Application, Enterprise Portals and Enterprise Service Bus* which represent the key concerns in *Enterpise Application Integration.*
Particular interest will be shown in the main features of an integration middleware that are mainly:

- Non intrusiveness
- Pervasiveness
- Standards adherence
- High distribution and event-driven programming model
- Autonomy and Federation
- Security and reliability
- Remote configurability
- Incremental adoptance

In the Heartfaid project the requirement is to integrate the application modules developed by the consortium partners:

1. The **AmI** is an Ambient Intelligence framework that handles the storage and monitoring of observational data acquired by sensors and interacts with environmental actuators. It has been developed by SYNAP in conjunction with FORTHNET and VMWS who delivered the sensing infrastructure.
2. The **eCRF** module is a web based Electronic Patient Record (EPR) that offers native user interfaces for input and edit patient related demographic and clinical information. It has been developed by JUMC.
3. The **groupSMS** is an application for sending short messages on mobile phones with handling of addresses lists and accounting. It is developed by FORTHNET.
4. The **CDSS** is the code of a Clinical Decision Support System developed by CNR in conjunction with RBI who designed the clinical ontology of the system.
5. The **HFP** is an image analysis and archiving toolkit used at CNR for implementing algorithms for the analysis of clinical images.

Other modules will be developed for granting the functionality that is missing: a *Master Patient Record, a Documental Repository, a Meta-data Registry, an Orchestration service and the Heartfaid Enterprise Portal.*
In order to prove the effectiveness of the integration proposals, a subset of the possible use cases has been selected for a early mockup implementation. Section 4 shows the mockup consisting in the integration of sensing facilities installed in the home of a patient with the ambient intelligence core provided by Synapsis and with the alarming and alerting system provided by FORTH. Observations flowing from the sensor network are acquired by the AmI core which stores them in an XML database and generates alarming requests to be sent when the observations are tagged with alarm information.
Implementation techniques included already the concepts of Content Based Routing, Data transformation, configurable orchestration, Message Oriented Middleware and HTTP based cross network communications.

The significant effort devoted to the definition and the design of a sound architecture for the Heartfaid Integration Middleware will be enforced by a comprehensive set of use cases that will be described in Section 5.
The use cases described in this document represent a significant subset of all those that Heartfaid plans to implement during the entire project lifetime. In particular:

- **A user authentication and profiling** use case shows how a user of the Heartfaid system can access a set of services through an Enterprise Portal. The list of services may require the user to be logged and could be tailored to the user's profile.
- A **global patient enrolment and consistent identification** use case states a uniform and homogeneized way of enrolling a new patient into the Heartfaid system through the Enterprise Portal and how to grant a consistent and global accepted identification mechanism for the patient.
- The **AmI enrollment use case** shows how monitoring or sensing resources can be assigned to patients for home and on-the-move moitoring.
- **Reporting of clinical, observational and demographic information** will describe how information coming from the clinical database of the eCRF module, observational information coming from the Ambient Intelligence module and demographic information of the patient can be automatically and flexibly integrate into visual reports by a proper Display service module. Those visual reports can be generated interactively or automatically and represent high value information that can be presented, stored or emailed by the middleware.
- The **automatic decision support for alarming and alerting** use cases show how to install (enroll) alarming and alerting services that are abel to decide through the cDD module when to trigger alarm notifications related to the status of a patient and how to handle this notifications (send emails, SMS, audits).

- **Handling of imaging data** has to describe the Heartfaid system level interaction with the services of storing and retrieving DICOM images produced and analyzed by the Imaging module.

These use cases represent the basis for implementing the reference worflows that HEARTFAID aims to support at the end of the project. As reported in the unsolicited "HEARTFAID Scenarios document" as well as in Deliverable "D23 - User needs analysis and functional specifications of the HEARTFAID platform services", three environments have been identified in which the platform should be able to support the different clinical processes: the medical environment (i.e. Primary and Secondary care contexts), the patient environment (i.e. the patient's home) and the research environment (i.e. the context in which addition data are collected for research purposes. In particular three workflows have been identified:

- Workflow 1 (medical environment): processes of diagnosis, management, prognosis assessment with patients' data collected and medical recommendations provided both in the office of the family physician and in the specialized cardiology setting.
- Workflow 2 (medical environment and patient's home): processes of diagnosis, management, prognosis assessment, with patients' data collected and medical recommendations provided, as above, both in the office of the family physician and in the specialized cardiology setting.
- Workflow 3 (medical environment and research environment): processes of diagnosis, management, prognosis assessment with patients' data collected (and medical recommendations provided) in the office of the family physician, in the specialized cardiology setting, and in the ultra-specialized research setting.

In the section 5 of the document each use case has been described in details in terms of objective, prerequisites, involved modules and schema (using UML *use-cases* and *activity diagrams)*.
This section will demonstrate that the architecture devised for the Interoperability Middleware is adequate to support the use cases identified and, consequently, also the HEARTFAID scenarios to be implemented. Therefore, starting from this first prototype developed in the first 18 month of the project, we can confidently assert that we are now able to proceed with the complete implementation of the HEARTFAID platform following the architectural design described in Deliverable "D11 - Functional specifications of the Middleware" and further refined during the last few months of activity (as reported in Section 4).

3. Analysis of the adopted clinical standards

The design and the implementation of a middleware for an interoperable platform of services operating in the eHealth sector cannot leave out of consideration the adoption of standards which are formally considered as reference standards for health applications at international level, as well as commonly adopted by the actors of this sector.
At the same time, in order to enable a middleware being a real support for data and business integration as well as for services interoperability, it is necessary to adopt the most common ICT standards for data encoding, information transmission and provision of services.
Therefore, prior to describe the implementation of the first HEARTFAID Middleware prototype, it is worth it to describe the standards selected as reference solutions for the modules to be developed and integrated.

3.1 Clinical standards overview

The HEARTFAID platform integrates health informatics products and medical devices in an efficient and effective system. Clinical standards mainly devoted to interoperability exist for both health informatics products and medical devices. According to IEEE, ***interoperability*** is the ability of two or more systems or components to exchange information (*functional interoperability*) and to use the information (*semantic interoperability*) that has been exchanged. With respect to *software*, the term interoperability is used to describe the capability of different programs to exchange data via a common set of business procedures, and to read and write the same *file formats* and use the same *protocols.* Same meaning has to be applied with respect to *medical devices.*
In the world of clinical information systems there are several standards in order to assure interoperability. In the Heartfaid Project we'll focus on the following:

- HL7 [28] (Health Level Seven), an ANSI-accredited Standards Development Organization operating in the healthcare; the HL7 organization aims to create standards for the exchange, management and integration of electronic healthcare information. In the HL7 framework, the CDA (Clinical Document Architecture) is an HL7 standard for the representation of clinical documents; it is an XML-based document markup standard that specifies the structure and semantics of clinical documents for the purpose of exchange.
- DICOM (Digital Imaging and COmmunications in Medicine), an international standards for communication of biomedical diagnostic and therapeutic information in disciplines that use digital images and associated data. The definition of this standard aims to achieve compatibility and to improve workflow efficiency between imaging systems and other modules of the healthcare information systems.
- **IHE** (Integrating Healthcare Enterprise), an initiative by healthcare

professionals and industry that aims to stimulate integration of healthcare information resources and to improve clinical care. IHE develops and publishes detailed frameworks for implementing established data standards to meet specific healthcare needs and supports testing, demonstration and educational activities to promote the deployment of these frameworks by vendors and users. IHE promotes the coordinates use of established standards such as DICOM and HL7 to address specific clinical need in support of optimal patient care. Systems developed in accordance with IHE communicate with one another better, are easier to implement, and enable care providers to use information more effectively.

Other interesting international standards have defined and used in the international community of healthcare experts.
For instance, the CEN/TC251 is an European Committee for the standardization in the field of Health Information and Communications Technology (ICT) to achieve compatibility and interoperability between independent systems and to enable modularity. The European Committee for Standardization (CEN) is currently working also on a five part European Standard (prEN 13606) that defines an information architecture for communicating contents of electronic health records (EHR). This standard enables health information systems to exchange EHR segments in a semantically interoperable manner. In this context is interesting the work performed by the OpenEHR foundation (http://www.openehr.org). **OpenEHR** is an effort whose aims are:

- promote and publish the formal specification of requirements for representing and communicating electronic health record information, based on implementation experience, and evolving over time as health care and medical knowledge develop;
- promote and publish EHR information architectures, models and data dictionaries, tested in implementations, which meet these requirements;
- manage the sequential validation of the EHR architectures through comprehensive implementation and clinical evaluation;
- maintain open source "reference" implementations, available under licence, to enhance the pool of available tools to support clinical systems; and
- collaborate with other groups working towards high quality, requirements-based and interoperable health information systems, in related fields of health informatics.

Another key issue regards the interoperability in medical devices communication and data format. Medical devices can be divided into two categories:

- Imaging Medical Devices
- Non-Imaging Medical Devices

For the imaging medical devices, **D**igital **I**maging and **Co**mmunications in

Medicine (DICOM) is a comprehensive set of standards for handling, storing, printing, and transmitting information. It includes a file format definition and a network communications protocol.
In the field of non-imaging medical devices, standards are not largely adopted by manufacturers. On the other side, the integration of medical devices with proprietary protocol/data format is time-consuming and not always feasible (several times manufacturers do not disclose the necessary technical information).
Considering these problems, IEEE decided to approach the medical device plug-and-play interoperability problem with a single communications standard. Software engineers designing medical equipment could use such a standard to implement external interfaces once for all models. This work has been published in the IEEE 1073 / ISO 11073 Series of Standards where the concept of Point-of-Care medical device communication is described.
These standards, intended to bring a wide range of medical devices under their purview, aim to encompass transparent plug-and-play interoperability, ease of reconfiguration, and ease of use.
Unfortunately, healthcare premises are plenty of legacy medical devices that do not implement this standard and nowadays in healthcare premises this standard is still "de jure" and not yet "de facto".
There are other standards for data communication and data format related to specific medical devices. For example, for resting ECG there are SCP-ECG[30] and HL7 Annotated ECG[28], while for Holter ECG there are MIT/BIH [31], ISHNE [32] and the EDF or EDF+ physiological signal recordings formats[33].

In the Heartfaid project only medical devices adopting communication standards have been selected.

In the following sections we'll describe how the clinical standards related to the comminications, the integration profiles and the medical imaging will be used in the Heartfaid project.

3.2 Communication standards

The main issue to be faced when realizing the communication infrastructure for the Hearfaid system relates to the adoption of communication and information exchange standards.
In particular the connection of clinical devices used for home and on-the-move monitoring is very critical because of the multitude of different technologies, standards and custom protocols present in the embedded device scenario.
From an architectural point of view this devices have to be connected via *Bluetooth*[40] *(*BT) to a ***reading station*** that is embedded in smart-phone device. In this case the information is transmitted in the proprietary protocol through the wireless BT channel. The smart-phone is responsible for transforming the information into an XML based protocol defined by the Heartfaid Consortium [41] and send it through a *GPRS* [42] gateway to the Heartfaid platform.
The communication over the wireless GPRS channel may occur utilizing standard

transport and *application level*[43] protocols such as TCP/IP or HTTP [44].

Concerning the communications that involve the modules of the Healthcare Information Systems, the Heartfaid platform adopts the HL7 standard.

HL7 standard defines a lot of messages, that corresponds to healthcare events, that healthcare related applications exchange among some others. The wide diffused version of the message standard is the 2, its last release is the 2.5 but the most used all over the world is the 2.3.1. The version 2 of the standard defines messages that are divided in one or more segments which content information is formatted by using delimitations characters, like circumflex accent (^), ampersand (&) and pipes (|), and uses the basic TCP/IP protocol for the transmission. Since the version 3 of the standard all information are encoded in XML and a lot of XML Schemas that enforce messages' structure.
Some of these messages are the one of the main features at the base of the Master Patient Index (MPI) working. The MPI is one of the modules of the Hearthfaid suite, the module that is in charge of patients' identity track. These message are used by MPI in order to inform and be informed of events that concern the patient's activity inside an healthcare enterprise.
More specifically, the “event” messages in following list are responsible for patient's identity propagation and synchronization through all the applications:

- Admit/visit notification (A01)
- Register a patient (A04)
- Pre-admit a patient (A05)
- Update patient information (A08)
- Merge patient (A40)

Moreover, MPI application, is queryable by other applications for patient's identification based on certain search criteria (name, surname, etc...).
In this case the message involved in this process is the “Find candidates” query (Q22).

The use of the standard inside an application is hard, due mainly to the large number of components and the high number of messages that are been defined by HL7 of the HL7 messages. For the Java language a framework library is available as open source with free licence. This library is named HL7 API (HAPI, pronounced Happy) and is available for all the subversion of the version 2 of the standard. The creation, parsing and exchange of the HL7 messages result very easy to implement by using this library.

Another fundamental issue is represented by the capability to manage structured data for the healthcare information production. In the Heartfaid project we adopt the HL7 Clinical Document Architecture standard (CDA). In this way it is possible to produce documents that can be human readable and, at the same time, both portable and easily processed by automatic tools to discover and import complex data. Actually HL7 with CDA standard uses XML as language for the data representation, and in the level 2 of this standard, data must be structured and coded using healthcare coding systems, such as these provided by **Logical Observation Identifiers Names** and **Codes** (LOINC) and by **Systematized NOmenclature of MEDicine Clinical Terms** (SNOMED-CT).

3.3 Integration profiles standards

In order to guarantee the improvement of the productivity of healthcare systems and to provide continuous and more personalized care solutions, it's necessary moreover to provide a patient centric vision of the healthcare delivery system, by integrating different care environments (primary care, homecare, secondary care) and the applicative modules operanting in each care environment. Key issues of this integration are clinical standards, described before and interoperable mechanisms. In particular, to reach actual integration among healthcare organizations and care environments, it is necessary to adopt standardized integration profiles for the implementation of interoperable healthcare services.

To really define an operative integrated healthcare platform of services it is necessary to identify also a set of structures that agree policies, procedures, protocols and so on.
In our opinion, this aspect is well addressed by the concept of the Affinity Domain, proposed by the IHE consortium. ***An Affinity Domain is a group of healthcare enterprises that have agreed to work together using a common set of policies and share a common infrastructure***. An Affinity Domain is equivalent to a Regional Healthcare Information Organization (RHIO) of the U.S.A. in that it defines a group of enterprises which will have an agreed method of sharing medical data. A RHIO implies that the group is regionally located while an Affinity Domain does not imply any regional assumptions. The Affinity Domain implies the presence of an organization and the concept can be applied recursively. Following th IHE indication, a clinical affinity domain is a group of healthcare enterprises that have agreed to work together using a common set of policies and share a common infrastructure.
Examples of affinity domains include:

- Community of Care supported by a regional health information organizations in order to serve all patients in a given region.
- Nationwide EHR.
- Specialized or Disease-oriented Care:
  - Cardiology Specialists and an Acute Cardiology Center.
  - Oncology network.
  - Diabetes network.
- Federation of enterprises.
  - A regional federation made up of several local hospitals and healthcare providers
- Government sponsored facilities (e.g., VA or Military).
- Insurance Provider Supported Communities.

For instance, pathology networks can be defined at regional level by means of different Affinity Domains and successively a nationwide pathology network can be realized by means a new Affinity domain integrating the regional ones.
In the HEARTFAID project we suppose to define and realize the platform of services in the context of an HEARTFAID affinity domain. In this domain, the clinical concept and the functional module that summarizes and exploits the integrated HEARTFAID platform is the Electronic Health Record.
An Electronic Health record (EHR) gives the comprehensive view on all the health information of the patient. In general this is an arduous problem, because healthcare environments are heterogeneous and geographically distributed.
To manage the heterogeneity and to enable integration between different software systems, since the nineties, a very large amount of standards, work-flows and integration guidelines have been introduced by healthcare related actors: healthcare organizations, healthcare institutes and standardization institutions in general.
The most known and widespread standards are: HL7 from **Healthcare Level 7** organization and the **Digital Imaging and Communications in Medicine** (DICOM) standard from American College of Radiology (ACR) and National Electrical Manufacturers' Association (NEMA); moreover there is the **Integrating Healthcare Environments** (IHE) initiative, led by healthcare professionals and industry to improve the way computer systems in healthcare share information.
IHE uses and adopts standards believed valid and helpful and defines the guidelines in one or more documents that are known as *Technical Frameworks*. One technical framework serves as a consensus document of how to think about, discuss, and successfully overcome medical information system integration problems by using existing standards and tools such as HL7, DICOM and so on. Each year the new version of the technical frameworks defines some new *Integration Profiles* in order to realize key interoperability tasks between systems that are identified as **actors** for the profile itself.

Concerning the realization of an EHR, in the context of the IHE guidelines the key modules to enable integration and interoperability are:

- a centralized patient identity management system that is able to manage and resolve patients identifications between different systems: a **Master Patient Index** (MPI) with **Patient Identity Cross Referencing** (PIX) capabilities (In the following simply called MPI);
- a distributed information **Repository** that contains information, that can be heterogeneous.
- a centralized **Registry** archive where all patient related information *meta-data* are stored, with also a reference to the real information, that can be queried and permits to retrieve these information as needed.

Using an MPI, a Repository and a Registry any healthcare related system can, according to transactions that have been defined in the interoperability standard, produce (if it is acting as **Producer**) and retrieve (if it is acting as **Consumer**) all needed information.

Producer	**Consumer**
All systems that act as information producer: ● Should use the MPI to assure that the produced information belongs to the entity that is *shared* by all the others systems, data and mainly the unique identifier. ● Should store on one or more Repository the produced information with its *metadata* using the entity's unique identifier. ● Should register in the unique Registry the information's *meta-data* and the location of the information in the Repository.	All systems that wants to retrieve information: ● Should use the MPI to assure that know the unique identifier of the entity. ● Should query the Registry by using one or more information's *meta-data* constraints in order to obtain zero or more links, as URLs, to documents that are stored into the repositories. ● Should retrieve the information contacting directly the repository with the URL.

Following the XDS-MS IHE profile, the last operation of the Producer is directly performed by the Registry actor, that registers the information *meta-data* subsequently to a *store operation* initiated by the information's source. Please note that the uniqueness is mandatory in this version of the profile and not represents any limitation.
In the following picture a schema of the interactions among the different actors is shown.

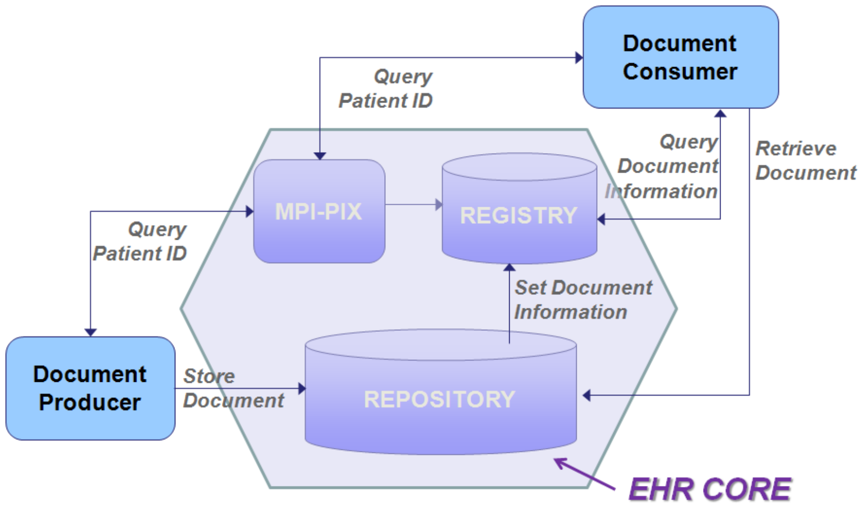


Figure 1: Interactions among the modules of the EHR

This is the approach the we'll follow in the HEARTFAID project. The XDS profile (Cross-Enterprise Document Sharing and the newest XDS for Medical Summaries, XDS-MS and XDS.b) enables a number of healthcare delivery organizations, belonging to a clinical Affinity Domain, to cooperate in the care of patient by sharing clinical records in the form of a document as they proceed with their patients' care delivery activities.
The XDS profile uses therefore the concept of Affinity Domain. An XDS Affinity Domain does imply the exact method for sharing medical data. That method is the XDS profile which is document based and includes a single, generally accessible Document Registry which contains a document entry for every registered document. The document entry contains a defined set of metadata elements. Part of the task of setting up an XDS Affinity Domain is identifying the location of the single Document Registry (there can only be one for the XDS Affinity Domain). Another part is defining a set of policy decisions which must be agreed to and abided by all participating enterprises. The policies include document format, acceptable values for some of the metadata elements of the document entries, coding systems and terminology. These subjects will be described in more detail later in the document.

The actors Repository, Registry and MPI and the related integration profiles allow to define the interoperable infrastructure of an EHR in the context of an Affinity Domain. IHE defines also an actor to access the patient-centric clinical information that is located outside the user's current application: the ***Display***. The Display is a module of the Healthcare Information System (HIS), that can become a component of the EHR front end. In particular, IHE defines the ***Retrieve Information for Display Integration Profile*** (RID), that can be exploited to access, in a simple and rapid read-only way, the documents of the EHR of each patient. The RID profile is used to access the information in a format ready to be presented to the requesting user and it supports access to existing persistent documents in well-known presentation formats such as CDA, PDF, JPEG, etc.

It also supports access to specific key patient-centric information such as allergies, current medications, summary of reports, etc. for presentation to a clinician.

The Heartfaid platform includes the Display module and it will be in charge of presenting the different and heterogeneous information related to a patient. In particular, the Heartfaid platform will extend the concept of the Display actor, including the presentation of information coming from modules not included in an HIS (for instance, the information related to the home monitoring of the patient, the results of a Clinical Decision Support System). Concerning the transactions that involve the modules of the HIS (Repository, Registry, MPI), the Display of the Heartfaid platform will adhere to the RID profile.

3.4 Imaging standards

3.4.1 DICOM

Since the 1970s, the emerging idea of a digital image archive (PACS) and electronic image distribution in a hospital created the need to exchange digital images between medical devices of different manufacturers. The American College of Radiology (ACR) and the National Electrical Manufacturers Association (NEMA) formed a working group in order to develop an image exchange standard. After some first failure, DICOM [34] was released in 1993 to offer an open vendor independent platform for the communication of medical images and related data, supporting PACS networks and guaranteeing interoperability. DICOM standard, which is composed of several parts, is continuously being extended and updated. The last part “WADO” was released in 2003.

It is important to point out that DICOM is not just an exchange format for medical image data. Actually DICOM goes far beyond, defining among several others:

- Data structures (formats) for medical images and related data
- Network oriented services, e. g.:
    - Image transmission
    - Query of an image archive (PACS)
    - Web-access to images
- Print (hardcopy)
- RIS - PACS - modality integration
- Formats for storage media exchange
- Requirements for conforming devices and programs.

For these reasons, any implementation of a DICOM standard should not just consider images in DICOM format, but should also involve the way images are stored, queried, retrieved and accessed through the web.

DICOM data structures

A DICOM image consists of a list of data elements (so-called attributes) which contain a multitude of image related information:

- patient information (name, gender, identification number)
- modality and imaging procedure information (device parameters, calibration, radiation dose, contrast media)
- image information (resolution, windowing)

Generally, the images belonging to one patient are organized in a *study*, that is the result of a complete diagnostic test. Such test may include several *series*, in their turn consisting in organized sets of images.

Each images is provided with a so-called DICOM header, gathering various image related information. The image Information Object Definition (IOD) is shown pictorially in Figure 2.

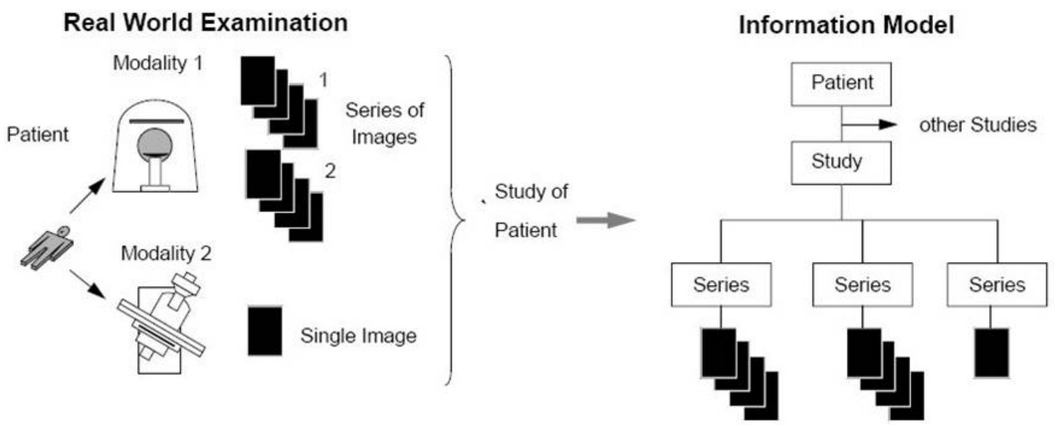


Figure 2: Information model for DICOM data structures

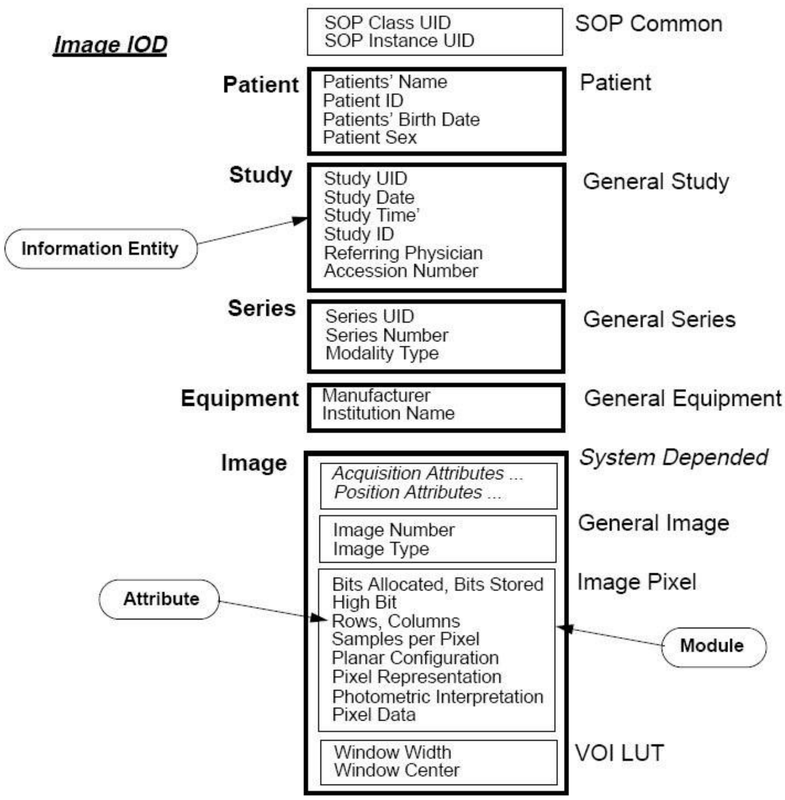


Figure 3: Image IOD

Network Services
DICOM network services are based on the client/server concept (see for example [37]). In the case of two DICOM applications exchanging information, they must establish a connection and agree on the following parameters:

- who acts as the client and who acts as the server
- which DICOM services are to be used
- data format of the transmission (e. g. JPEG compressed or uncompressed)

Only if both applications agree on a common set of parameters, the connection can and will be established. In addition to the most basic DICOM service image transmission (the so-called *Storage Service Class* such as C-STORE), several advanced services are available. The following two, for example, may reveal useful in HFP:

1. The DICOM *image archive service* (“Query/ Retrieve Service Class”) allows to search images in a PACS archive according to certain user-defined criteria (patient, time of creation of the images, modality etc.) and to selectively download images from this archive.
2. The DICOM modality worklist service allows to automatically download up-to-date worklists including a patient’s demographic data from an information system (HIS/RIS) to the modality.

Media Exchange
In addition to the exchange of medical images over a network, media exchange has become another focus which has been integrated into the DICOM standard in 1996.
Fields of application are for example the storage of cardiac angiography films in cardiology or the storage of ultrasound images.
In order to make sure that DICOM storage media are really interchangeable, the standard defines so-called application profiles which explicitly define images from which modalities may be present on the medium (e. g. ’only X-Ray Angiography images’), which encoding formats and compression schemes may be used (e. g. only uncompressed or loss-less JPEG), which storage medium is to be used (e. g. ’CD-R with ISO file system’).
Further, each DICOM medium contains a so-called DICOM directory This directory contains the most important information (patient name, modality, unique identifiers etc.). It is possible to quickly browse or search through all images on the medium without having to read the complete image files (which can take a very long time).

Web Services
Users of medical information systems may benefit from rapid and reliable access to reports and images. For example it could be important during an examination to retrieve on the fly reports or even images of the previous examination. Within HFP it is likely that such access will be based on web technologies also for image retrieval and visualization. The access to relevant DICOM *persistent objects* (i. e.

images and reports, not logs of workflows) should be either in native DICOM format for advanced use or rendered into a generic format (e. g. JPEG, PDF) that can be presented with off-the-shelf applications. DICOM standard offers the so-called "WADO" (Web Access to DICOM persistent Objects) to answer these needs. It is a web-based service for accessing and presenting DICOM persistent objects, consisting in a simple mechanism for accessing a DICOM persistent object from HTML pages or XML documents, through HTTP/HTTPs protocol, using the DICOM UIDs (study, series, and instances). Actually, DICOM specifies how to compose an URL based on the involved UIDs pointing to a particular image (See Figure 4). We refer to [36] for a discussion of WADO and the presentation of several scenarios.

Security Concerns and Current limitation of WADO

Clearly the information contained within DICOM objects may be considered as Protected Healthcare Information (PHI). Thus the protocol used, that is HTTP, can be replaced by HTTPs for that purpose. Further, DICOM standard defines two optional parameters, *anonymize* and *annotation*, which control respectively the absence of patient identification in a retrieved DICOM object and the presence of patient identification burned into the pixel data of images. It is likely, however, that for patient enrolled into HFP personal information will be "erased" and replaced with an ID (the same used in eCRF) before storing images in the HEARTFAID image archive.

When dealing with echocardiography (the key image modality in HFP), one usually deals with multiframe images (i.e. image sequences). It would be interesting to be able to retrieve and visualize multiframe images via a web-based service. However, considering the size of such kind of images, some sort of *streaming* is necessary. Currently, up to the best of our knowledge, no kind of streaming is suggested or prescribed by DICOM standard. In particular, WADO consents the access only to the first frame of a multiframe image.

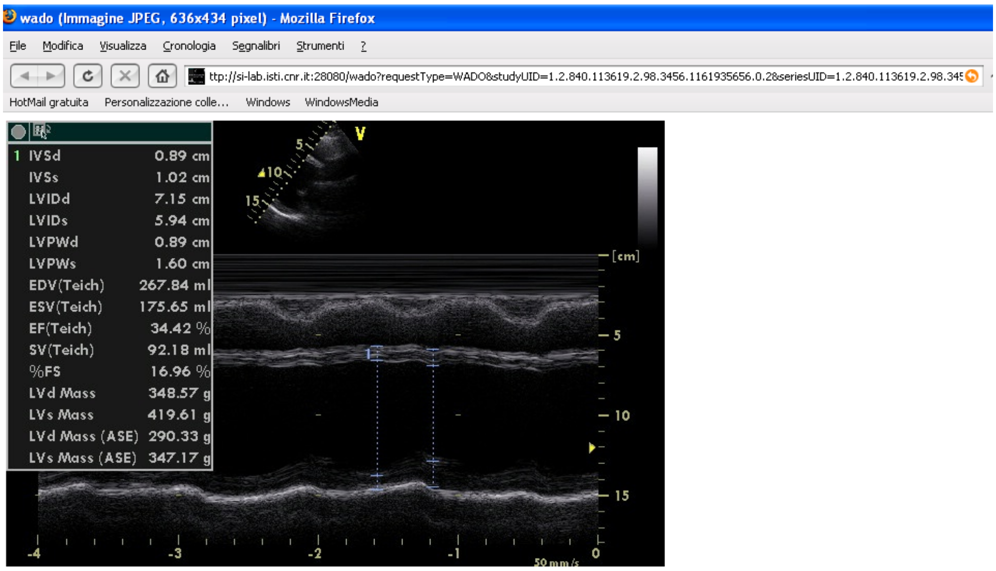


Figure 4: A basic WADO request

3.4.2 IHE Cardiology

As we saw, DICOM defines a complex standard to guarantee interoperability among different DICOM compliant devices ranging from DICOM server to radiology workstation, from imaging devices (US, CT, X-ray...) to HIS. Any DICOM compliant device comes with a DICOM conformance statement in which the portion of implemented DICOM features is described. In practice, however, conformance statements are only comprehensible by experts and they are frequently inadequate since often only a minimum set of features is documented. In some cases interoperability problems tend to occur because some inconspicuous details do not go together.

To answer these needs, IHE (an initiative by healthcare professionals and industry) [27] tries to improve the sharing of healthcare information by promoting the coordinated use of established standards such as DICOM and HL7 to address specific clinical needs in support of optimal patient care.

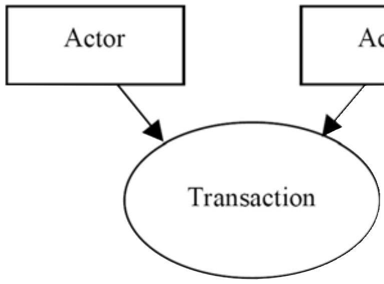
IHE Technical Framework defines specific implementations of the afore-mentioned established standards to achieve seamless transmission of information among physicians, medical specialists, nurses, administrators and other stakeholders.

IHE specifies several integration profiles, which are representations of a real-world capability that is supported by a set of actors that interact through transactions. In IHE terminology, actors are information systems that produce, manage, or act on categories of information required by operational activities in the enterprise/platform.

Transactions, in turn, are interactions between actors that transfer the required information through standards-based messages (e.g. HL7).

In particular IHE provided a Cardiology Technical Framework [38,39] (written by the American College of Cardiology (ACC), the Radiology Society of North

America (RSNA) and Healthcare Information and Management Systems Society (HIMSS)).



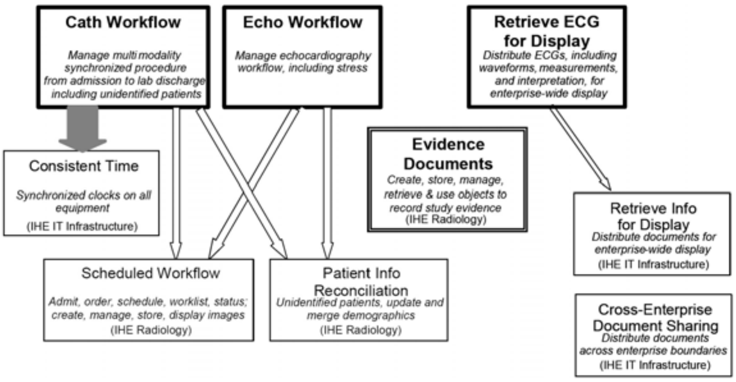


Figure 5: IHE cardiology normative area

The normative area of IHE cardiology consider a bunch of workflows relative to diagnostic procedures such as echocardiography,catheterization and ECG. Services for XDS, consistent visualization and integration with the overall patient and hospital workflow are also considered (See Figure 5).

In particular, echocardiography workflow is included with the aim of (see also [37]):

- Providing echo measurements interoperability
- Ensuring images and measurements are securely stored
- Uploading echo reports to a repository
- Reconciling patient demographics

According to previusly introduced IHE terminology, this is obtained implementing several *actors* (ADT/HIS, Scheduler, Acquisition modality, Image Manager, Evidence Creator).

For example, the process of image acquisition, storing, visualization and annotation proposed by IHE is depicted in Figure 6. Since there is an obvious relationship between IHE actors and various modules in the HFP related to the image archive, it is at least reasonable to explore IHE Cardiology Technical Framework in the close future. For the moment, it is timely to adopt a DICOM image archive implementation which integrates IHE actors, that is, according to IHE terminology, *Image Manager/Archive* and optionally *Report Manager* and

Report Repository.

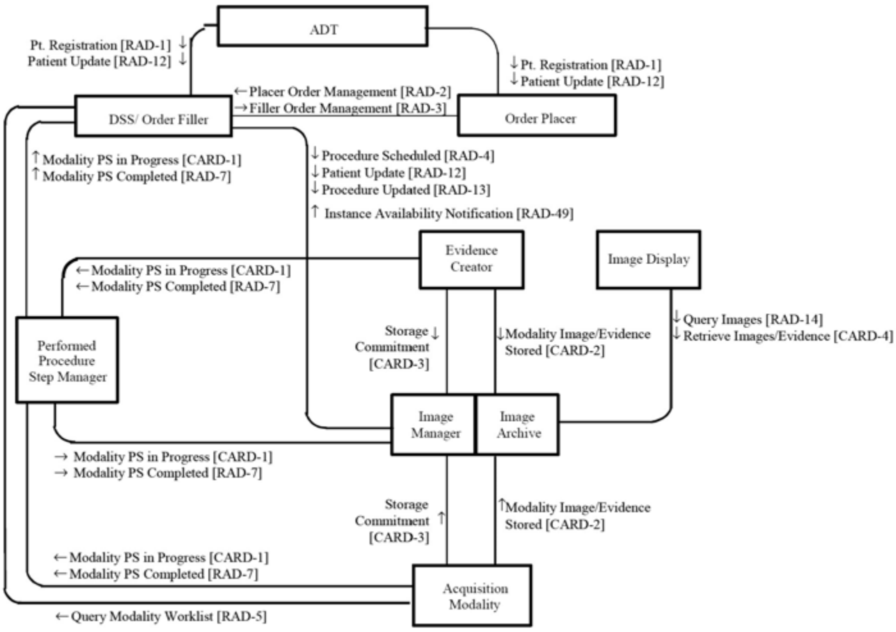


Figure 6: Diagram of IHE actors involved in echocardiography workflow

4. The first middleware prototipe

In a large modern enterprise, it is almost inevitable that different parts of the organization will use different systems to produce, store, and search their critical data. Yet, it is only by combining the information from these various systems that the enterprise can realize the full value of the data they contain [1].
In a scenario where the data sources are several, distributed, pre-existing and heterogeneous it is not possible to think of a solution that creates a new data source by physically merging the others.
[3] reports this about integration:

> *Of what might be called "market driver" trends, integration has replaced security as the highest priority in IT planning for 2004 in North America among IT and executives interviewed in June.*

The key issues is to choose the correct ways of implementing the integration of the systems taking into account flexibility of the system to be integrated, its legal access policy and the technologies it is implemented with and that it can support.
There are many mechanisms for integrating data and functionality.
Application-specific solutions: data integration is via special-purpose applications that access sources of interest directly and combine the data retrieved from those sources with the application itself. This approach always works, but it is expensive (in terms of both time and skills), fragile (changes to the underlying sources may all too easily break the application), and hard to extend (a new data source requires new code to be written).
Application-integration frameworks: typically employ a standard data or programming model, such as CORBA or J2EE. Data and application access is performed through adapters that use well-defined interfaces implemented by the framework. These frameworks protect the application somewhat from changes in the data sources (if the source changes, the adapter may have to change). The application programmer is not required to have detailed systems knowledge, so applications will typically be easier to write. However, such systems do not necessarily address data integration issues; if combination, analysis, or comparison of the data received from the various sources is needed, the application developer must provide that code.
Digital libraries and Portals: collect results from multiple different data sources in response to a user's request. Usually the sources are similar in function (e. g. all store text documents, bibliographies, URLs [uniform resource locators], images, etc.). For most of these "meta search engines," no combination of results is done, except perhaps for normalization of relevance scores or formats—all are returned as part of one list, possibly sorted by estimated relevance. Similarly, *portals* provide a means of collecting information—perhaps from quite different data sources—and putting the results together for the user to see in conjunction.

Data warehouses and database federation: provide users with a powerful, high-level query language that can be used to combine, contrast, analyse and otherwise manipulate their data. Technology for optimizing queries ensures that they are answered efficiently, even though the queries are posed non procedurally, greatly easing application development. A data warehouse is built by loading data from one or more data sources into a newly defined schema in a relational database. The data are often cleansed and transformed in the load process. Changes in the underlying sources may cause changes to the load process, but the part of the application that deals with data analysis is protected. New data sources may introduce changes to the schema, requiring that a new load process for the new data be defined. SQL views can further protect the application from such evolutions. However, any functions of the data source that are not a standard part of a relational database management system must be re-implemented in the warehouse or as part of the application.
Enterprise Service Bus (ESB): is a new approach to integration that can be used as the foundation for a loosely coupled highly distributed and thus highly scalable integration network. *An ESB is a standards based integration platform that combines messaging, web services, data transformation and intelligent routing to reliably connect and coordinate the interaction of a significant numbers of diverse applications across extended enterprises with transactional* integrity [2].

In a scenario that closely matches the Heartfaid project one can imagine that there will be different data sources for storing patient related clinical and demographic data that is produced and edited from a traditional web-based application, observational data that is produced by sensors, imaging tools, analysis algorithms, alarm systems, reporting results kept in a document oriented repository.
In most cases each system (the web application, the AmI platform, the sensor network, the image analysis algorithms) is possibly not aware of being part of a bigger picture.
Each ***system administration*** must somehow clarify the data it wants to share, the functionalities it wants to promote to the overall infrastructure for being a component functionality of a higher level service, the user, role and access rights it wants to be granted.
All those systems (data bases and functionality) have to be put together by an **integration middleware** which has the following characteristics:

- **Non intrusive**: the integration should be as transparent and seamless as possible to the integrated applications and application modules.
- **Pervasive:** the integration middleware should be able to span an extended domain that crosses technologies and organizational boundaries such as (Firewalls, Internet, Departmental access policies, etc.)
- **Standards based**
- **Highly distributed and event-driven**
- **Autonomous but federated:** cooperation as a means for granting performances, robustness and fault tolerance
- **Secure and reliable**

- **Remotely configurable: natively** distributed architectures have to be controllable by remote management consoles to reduce maintaining costs.
- **Incrementally adoptable:** every module might have its own pace of adhering to common policies and integration requirements. The middleware should allow the more flexible organizations to integrate quickly without impacting on all other that could be slower.

Our approach to the design and implementation of the Heartfaid integration middleware will be to consider patterns and best practices in ***Enterprise Application Integration*** together with the edge technologies in terms of integration namely the Enterprise Service Bus (ESB) enriched by the delivery of proper modules seamlessly accessible and integrated supporting the development of a Heartfaid Portal that offers a highly integrated user experience to the Heartfaid end users.

4.1 The architectural design

In a first analysis step the different modules that contribute to the Heartfaid applications have been identified. The main application modules are developed by the Heartfaid partners and range from full blown applications with their own native interfaces and storages, through head- and state-less web services to code-on-demand libraries for very specific computations.

1. The **AmI** is an Ambient Intelligence framework that handles the storage and monitoring of observational data acquired by sensors and interacts with environmental actuators. It has been developed by SYNAP in conjunction with FORTHNET and VMWS who delivered the sensing infrastructure.
2. The **eCRF** module is a web based Electronic Patient Record (EPR) that offers native user interfaces for input and edit patient related demographic and clinical information. It has been developed by JUMC.
3. The **groupSMS** is an application for sending short messages on mobile phones with handling of address lists and accounting. It is developed by FORTHNET.
4. The **CDSS** is the code of a Clinical Decision Support System developed by CNR in conjunction with RBI who designed the clinical ontology of the system.
5. The **HFP** is an image analysis and archiving toolkit used at CNR for implementing algorithms for the analysis of clinical images.

The scenario is depicted in the following picture showing the application with their datamodel and each one with its applicative core and native access interface.

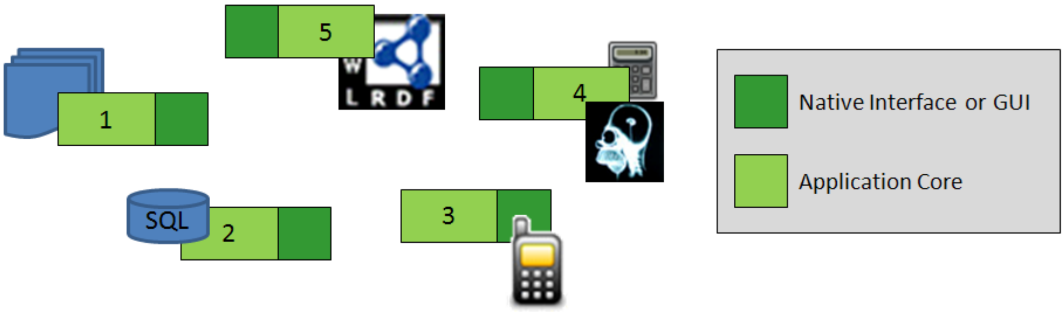


Figure 7: Datamodel of the involved applications

This short list already presents several issues in terms of integration. Every module is written in its own language (Java, VB, HTML, C/C++, ontology modeling languages ...), according to its own application model (web application, distributed application, legacy stand alone rich application, code-on-demand library ...), requiring its own data type (name-value strings, XML, relational data) and exposing native interfaces in its own presentation formalism (HTML view, Rich Internet Application, Java Swing, OS native GUIs).
Moreover every module has its own way of implementing interconnection and networking (from nothing at all through bare TCP/IP to web services and JMS based messaging) and for handling life-cycle management.
In such a complex scenario it becomes very obvious that an integration strategy can not be based on point-to-point agreements among each pair of application modules that have to be integrated. This approach would lead to an explosive situation from a maintenance point of view and to a worst case approach from a scalability point of view.
Finally there are also difference and mismatches in how the different modules handle common or correlated information such as for example patient identification.Nonetheless they all contribute with important functionality and even more important data.
Thus it has been decided to adopt a state of the art approach based on the installation of an ESB.

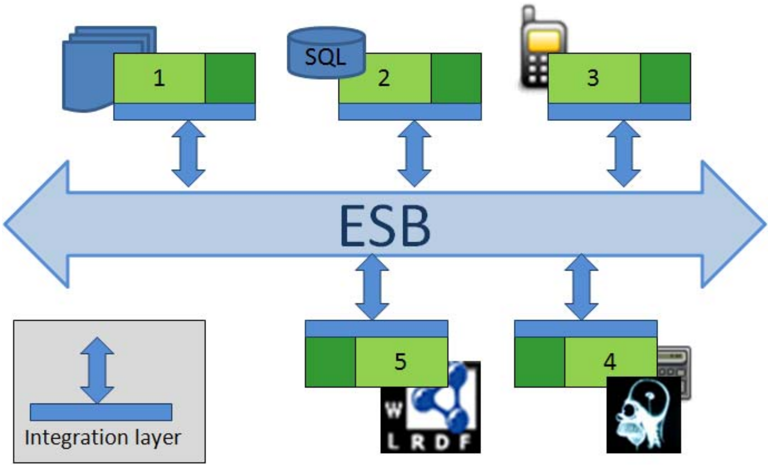


Figure 8: Architectural viewpoint

In Figure 8 a possible scenario is depicted from an architectural viewpoint.
An adapter has to be written for plugging the application modules to the ESB. The wrapped application will be able to loosely interconnect and communicate with the other modules through a message passing infrastructure that is the foundation of the ESB.
At this point the integration architect takes care of designing and configuring the interactions among the application modules, the data transformations, quality of service, synchronization, workflow specification, life-cycle management and so on for granting the implementation of the services that want to be exposed at global Heartfaid level. This activities are performed leveraging the technologies and tool support contributed by the underlying ESB thus dramatically reducing the development effort.

So far the integration has been analyzed only from a technical viewpoint. There are also requirements on functionality that has to be added for completing the integration middleware as a whole.
In particular there four main functionalities that have to be taken into account.
Uniform access to shared resources: besides the data and information locally generated by each application module, a system like the integrated Heartfaid platform requires a lot of information to be shared. This information has to put into the middleware in a way that is as transparent and easy as possible for the application itself without however impacting to heavily on the retrieval costs.
A best practice is to use a combination of two pattern of application integration [4]: a ***documental repository*** and a ***registry service.*** The documental repository one stores data in a weakly structured way that allows the developer to access it with very simple primitives such as *put* and *get* calls thus removing from the software engineer the burden of writing complex queries in native database idioms. The registries allows any module of a system to retrieve a *reference to* a piece of information stored in the system (possibly a repository) through querying by a subset of the meta-data identifying it.
Unique identification of patients: it is very important that the possibly different

identifications of the patient that are kept by the different application modules are uniformed and synchronized across the whole Heartfaid information system. This cannot be done by enforcing a common identifier to all the participant application modules because of the high cost it could have to change such a typically hard coded and structural information. Thus a proper entity called ***Master Patient Index*** has to be integrated with the task of synchronizing different patient identification schemes and making this information regulate the access to patient related information across the whole system in a way that is as transparent as possible.

Generation of visual information: a proper *Display* application module can be designed for handling all the activities related to the presentation of information in a scalable though homogeneous way. This application could be actually seen as a report production service that is able to generate displayable documents in different formats (HTML, PDF, ...) according to specific requests, time scheduling or other events.

User access control and interaction: most of the services exposed at the overall Hearfaid level have to be interactively started or guided by human operators. This implies that a proper strategy of integration has to be thought of also at the presentation level. According to best practices and patterns in enterprise application integration the most suitable solution is to setup an ***Enterprise Web Portal*** (shortly Portal). The Portal will be in charge of unifying the access to the Heartfaid system for all kind of purposes (configuration, service interaction, dissemination).

Unifying the access means also controlling the access policies and access rights of the different classes of users that will be able to selectively access the Heartfaid system according to profiling information. Moreover the Portal is also a mean of realizing an early prototype of a ***Single Sign On*** (SSO) system that grants the user access to all the available components, data or functionality logging in only once.

The following picture shows the above mentioned architecture enriched by the described application modules.

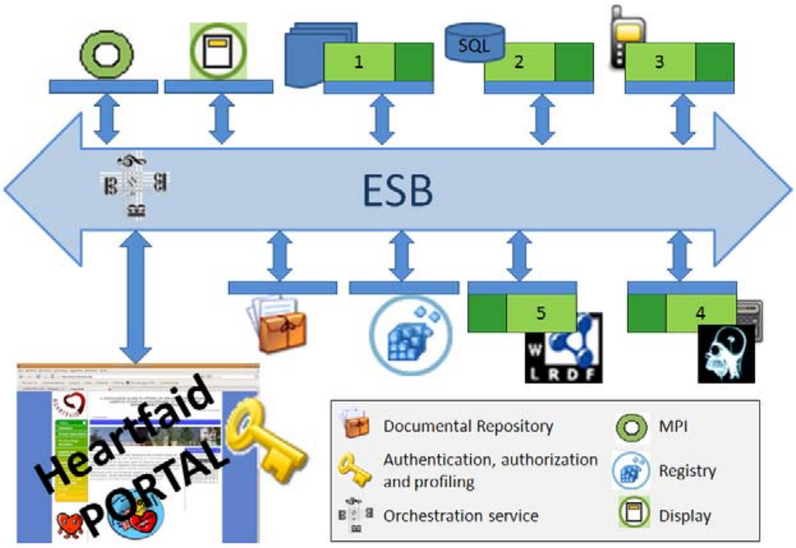


Figure 9: The Heartfaid architecture

As can be seen the middleware also makes use of one of the most important feature of ESBs: the **Orchestration Service (OS)**. The OS is highly configurable, scriptable actor of the ESB which is involved in the workflow management regulating the interactions and data exchanges of the different application modules participating in the integrated Heartfaid platform.

4.2 Implementation details

4.2.1 General considerations

As a general approach for the implementation of the Heartfaid integration middleware, the development team will choose open source technologies whenever possible and leverage on agile software design methodologies for achieving rapid and experience prototyping.
The development of software will be object oriented
A significant shift from coding and programming based solution towards configuration based, extensible development patterns such as Inversion of control and dependency injection will be

4.2.2 Heartfaid state of the art

The problem of integrating different heterogenous systems has already been faced in the early Heartfaid prototypes. The solutions adopted in those cases adhered already to the main patterns of enterprise application integration.

- XML for information exchange
- standard protocols (TCP/IP, HTTP) for interconnecting to legacy systems
- an enterprise MoM based on JMS[11]

- business logic orchestration language based on an Synapsis proprietary XML specification
- use of standard information models such as OM, SensorML, OMF [13]
- use of native XML persistence based on XML:DB[]
- use of W3C[20] technologies (XSLT, Xquery, Xpath) for transformation, querying and CBR.

The adopted solutions allowed the development and integration team to face different scenarios with a big percentage of code and experience reuse thus reducing dramatically the implementation time. The following picture describes an example of the use of the above mentioned technical solutions to implement the **home monitoring of a patient** through automatic sensing devices, the notification of SMS and email alarms, the storage of the acquired information in a **Heartfaid observational database.**
The information has to be merged with a patient presentation tool that runs on a palmtop PC and recognizes the patient through an **RFID** device which has to display primary care information. The integration with an **MPI** and a **Hospital Information System** allows the application to identify the patient globally and to retrieve demographic and clinical data that are displayed on the palmtop together with a report on the data monitored at home through the Heartfaid devices.

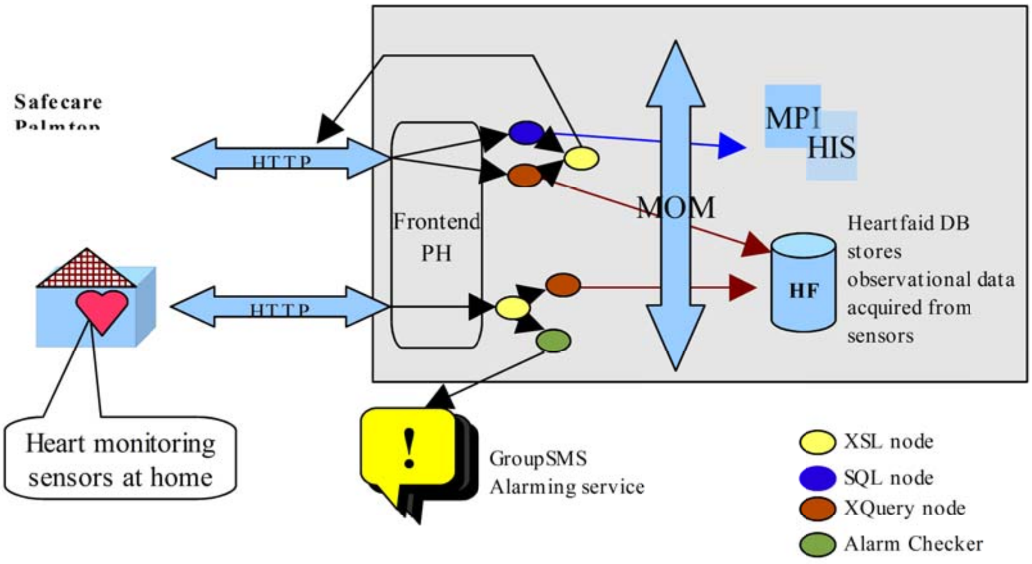


Figure 10: State of the Art of the Heartfaid Architecture

4.2.3 Technological guidelines

The implementation of the Heartfaid integration middleware consists in:

- Portable programming environment
- A reliable, asynchronous message passing infrastructure.
- XML based information exchange, transformation and filtering
- The implementation of the Web Portal.
- The design of the integration interface that homogenize the communication among the different application modules converting from and to the **common**

language (the integration interface).

- The implementation of a management framework for the (remote) management of the integration interfaces' life-cycle to be possibly included in the Portal.
- The implementation of an **Orchestration Service** with its workflow description language.

There are several technologies that address the above mentioned implementation aspects or subsets of them.

Portable programming environments are used to programming and deploy applications, modules, libraries without having to worry too much about the target platform. There are several solutions based on interpreted scripting languages (Python[6], Perl[7], Ruby[8]) or on virtual machines (Flash[9], Java[10]). The choice of implementation will be Java because of the huge amount of software solutions that already exist for nearly every available platform.

There exist many possible solutions for implementing **reliable and asynchronous message passing systems**. The natural choice for a Java based integration middleware is to select one messaging framework that adheres to the Java Messaging Service (**JMS)** specification. The *Java Message Service (JMS) API is a messaging standard that allows application components based on the Java 2 Platform, Enterprise Edition (J2EE) to create, send, receive, and read messages. It enables distributed communication that is loosely coupled, reliable, and asynchronous[11].*

There exist several implementations of the JMS specification many of them are in the open-source domain (OpenJMS, JBossMQ, ActiveMQ, Websphere)

XML messaging will be based on information content based on the standards mentioned in previous section 3. The content will be wrapped with a message description XML language such as SOAP[12] or OMF[13].

Web Portal technologies are widely used in designing of web sites that serve as user interface level integrators of information and services.

There are also standard specification being developed for defining web portal behavior [14][15][16][17].

Some common known and used web portals are Websphere Portal from IBM, Zope which is open source and written in Python and PHP portal which is also open source but written in PHP.

Jboss [18] is a Java J2EE and JSR168[16] based portal product which is open source. This could be a viable solution since it is also a J2EE application server implementing all J2EE specification including a built-in JMS implementation (JBossMQ).

The design of the integration interfaces is a very application module specific (probably the only one) activity. It consists in the realization of adapters that make the application specific communication, data format and life-cycle adherent to the common model of the ESB. The adapters will be written in Java and will exploit typical interconnection technologies for connecting to the applications module (Web services, TCP/IP communications, Java Native Interface JNI for loading and execution of legacy code ...).

The implementation of the adapter management infrastructure will leverage the Java Management Extension (JMX) specification that *provides the tools for building distributed, Web-based, modular and dynamic solutions for managing and monitoring devices, applications, and service-driven networks. By design, this standard is suitable for adapting legacy systems, implementing new management and monitoring solutions, and plugging into those of the future.* [19] Some implementations of JMX offer also web interfaces that could be inserted in the web Portal and made available to the Heartfaid user.

As for the Orchestration service the possibility is to utilize a centralized work-flow management tool or distributed routing strategies based on IBR or CBR that allows to develop orchestrated workflow without having ti rely on a centralized routing agent [2]. In the latter cases the routing rules are kept as meta-data inside the messages or injected as configuration scripts (written in Xpath, Xquery or other scripting languages i. .e. Javascript) into the distributed integration interfaces.

As a final implementation hint it's worth noticing that there exist several ESB systems that implement all or a significant subset of the above mentioned technologies thus delivering commercial [23][24][25] or open-source [21][22] off the shelf ESB implementations.

For the definition of the interactions among the different modules of the Heartfaid system a consistent approach has been selected. It is based on encapsulating the different modules into IHE Actors and establishing the information exchange as an implementation of the IHE integration profiles transactions. In particular the MPI module will implement the PIX integration profile, the Repository and the Registry will implement the XDS profile whilst the Display module will implement the RID profile. The information exchanged in the communications will be encoded according to the HL7 protocol as described in Section 3.2.

5. Use cases Description

5.1 Use cases design approach

As a design approach for understanding the requirements of a possible implementation of the Heartfaid integrated platform, the partners agreed upon examining a set of possible use cases and services that could be implemented as demonstrator.

These use cases are considered necessary in order to implement the workflows reported in both the unsolicited document on the "HEARTFAID Scenarios" and in Deliverable "D23 - User needs analysis and functional specifications of the HEARTFAID platform services". These workflows, which involve three different environments, that is the medical environment (primary and secondary care), the patient environment (patient's home) and the research environment, have the main goal to involve all the services that are expected from the platform and, therefore, the functionalities that need to be implemented.
In other words, these workflows are very important since they are considered as the reference scenarios to implement all the modules of the platform of services. For this reason, the use cases have been identified and defined according to these reference forkflows.

The use cases have been modeled according to the *flow of information* that has to be realized in order to deliver the required services. In particular, we focus on:

- **the user authentication and profiling** use case: it shows how a user of the Heartfaid system can access a set of services through an Enterprise Portal. The list of services may require the user to be logged and could be tailored to the user's profile.
- The **global patient enrolment and consistent identification** use case: it states a uniform and homogeneized way of enrolling a new patient into the Heartfaid system through the Enterprise Portal and how to grant a consistent and global accepted identification mechanism for the patient.
- The **AmI enrollment** use case: it shows how monitoring or sensing resources can be assigned to patients for home and on-the-move moitoring.
- **Reporting of clinical, observational and demographic** information: this use case will describe how information coming from the clinical database of the eCRF module, observational information coming from the Ambient Intelligence module and demographic information of the patient can be automatically and flexibly integrate into visual reports by a proper Display service module. Those visual reports can be generated interactively or automatically and represent high value information that

can be presented, stored or emailed by the middleware.

- The **automatic decision support for alarming and alerting** use cases: they show how to install (enroll) alarming and alerting services that are abel to decide through the cDD module when to trigger alarm notifications related to the status of a patient and how to handle this notifications (send emails, SMS, audits).
- **Handling of imaging** data: this use case has to describe the Heartfaid system level interaction with the services of storing and retrieving DICOM images produced and analysed by the Imaging module.

In the following sections each use case will be described accurately in terms of descriptions, prerequisites and involved modules, reporting also the drawing of UML use cases and activity diagrams.

5.2 Login

Description: in the login use case a user is asked to identify himself through a proper form in the Portal. User name and password are then matched against the profiling database that is maintained directly by the Portal and on authorization the right presentation page is shown.
Prerequisite: the user accesses the public section of the Portal from where he/she can login. The Portal user profiling database contains a record describing the user's profile.
Result: the user is prompted with a page in the Portal which states the success or failure of the login procedure.
Involved modules: the Login is performed as a whole by the Portal thus no other application module is involved.

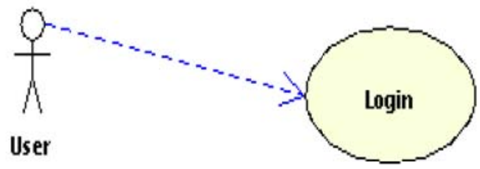


Figure 12: Login use case

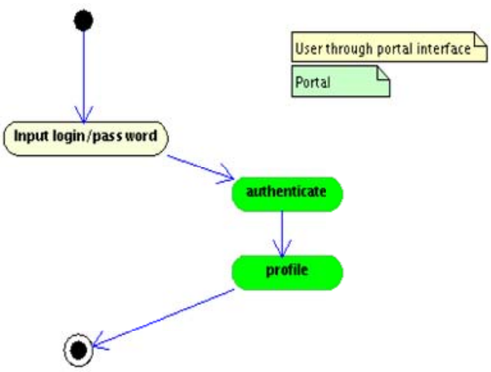


Figure 11: Login activity diagram

5.3 Set Patient Context (SPC)

Description: *SPC* is a basic use case that sets the user in a proper patient context. This means that the user handles information related to a specific patient until he/she closes the application or switches to another patient. The patient has to be identified through a proper patient search interface in the Portal. The identification of the patient occurs through the input of patient demographic information and after the search process completes, one patient has to be selected from the list of possibly matching patients.
Prerequisite: a *Login* has to be executed successfully.
Result: the user is prompted with a page containing a list of patients matching the query and he/she has to choose one among the matching patients. At this point the patient context is set.
Involved modules: besides the Portal, the Master Patient Index is involved for executing the demographic query and the Display for generating a report of all matching patients.

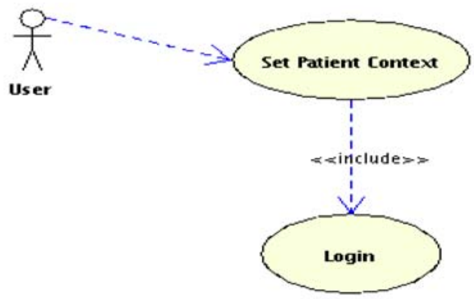


Figure 13: SPC use case

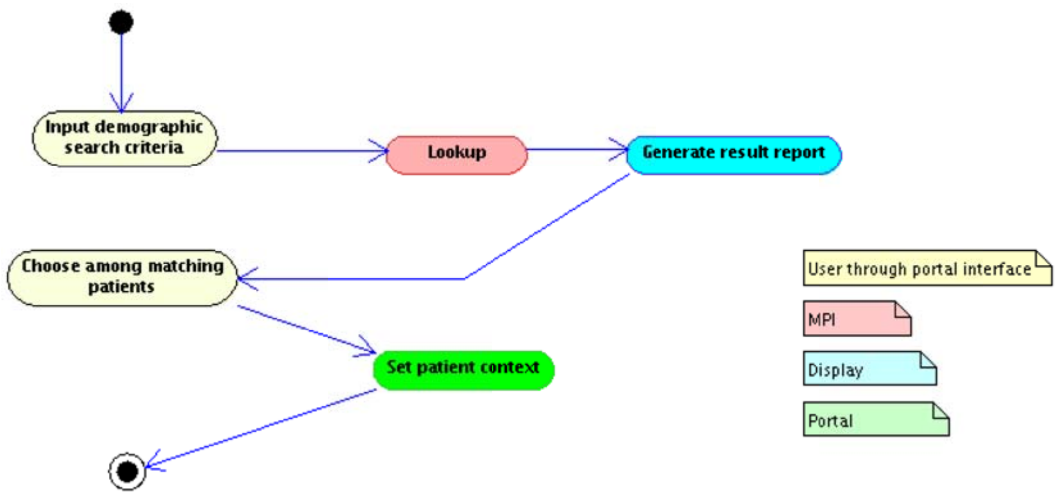


Figure 14: SPC activity diagram

5.4 Global Patient Enrollment (GPE)

Description: the *Global Patient Enrollment* describes the act of *inserting* a new patient into the global Heartfaid platform. This operation is different from *local enrollment procedures* that actually register patient related data into the different application modules possibly by using native interface functionality. With the *GPE* the platform and in particular the MPI and the eCRF become aware of the presence of:

- a new Heartfaid Patient identity
- the patient's demographic data
- the patient's inclusion/exclusion criteria

Prerequisite: the user requesting a *GPE* has to be logged and authorized by the Portal.
Result: after a *GPE* the MPI and eCRF are synchronized with the demographic information of the new patient. The Portal context is set to the enrolled patient.
Involved modules: the GPE is guided by a proper interface in the Portal. The new inserted data (identification and demographic data of a patient) are sent to the MPI and the eCRF.

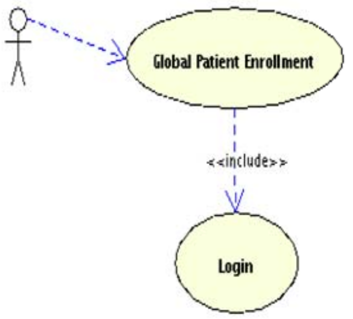


Figure 16: GPE use case

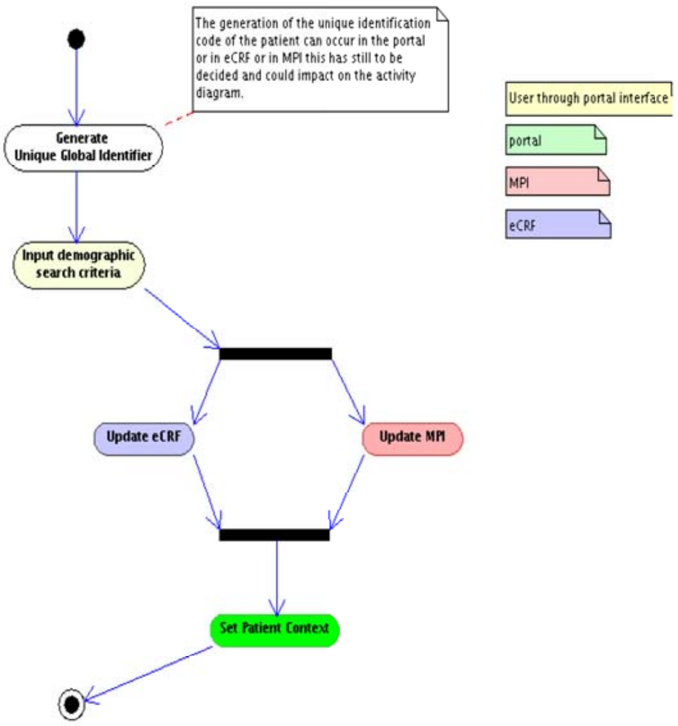


Figure 15: GPE activity

5.5 Real Time Reporting of patient data(RTR)

Description: the *RTR* use case is meant to be the operation through which data

that is stored locally in each application scenario is retrieved for the assembly of global reports that are then rendered on a specific displaying device by the Display module. The *RTR* matches the *Federated Query* integration pattern. The list of available reports is visible in a proper view of the Portal. The user could select the report and the format. At that point the Portal triggers the orchestration service which is responsible for steering the federated query, the gathering of the result and the production of the visual report.
Prerequisite: *Login* and *SPC* have to be executed. The Orchestration Service has to be made aware of the workflow of Federated Query.
Result: a report is sent back to the Portal. Optionally the report could also be sent to an email address or stored in the documental repository (in a possibly different format).
Involved modules: The Portal allows the user to set the patient context and choose the report and then it should be able to display it. The *Orchestration Service* steers the federated query and redirects the resulting report. The Display produces the visual report. The Registry and repository could be involved in the storage of the report data.

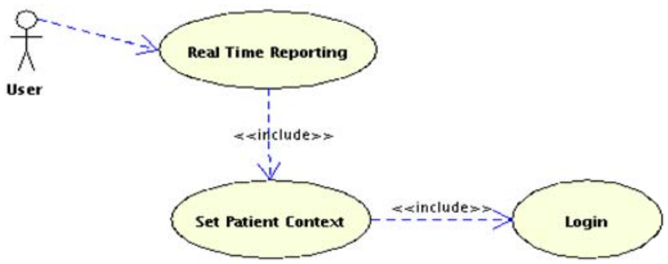


Figure 17: RTR use case

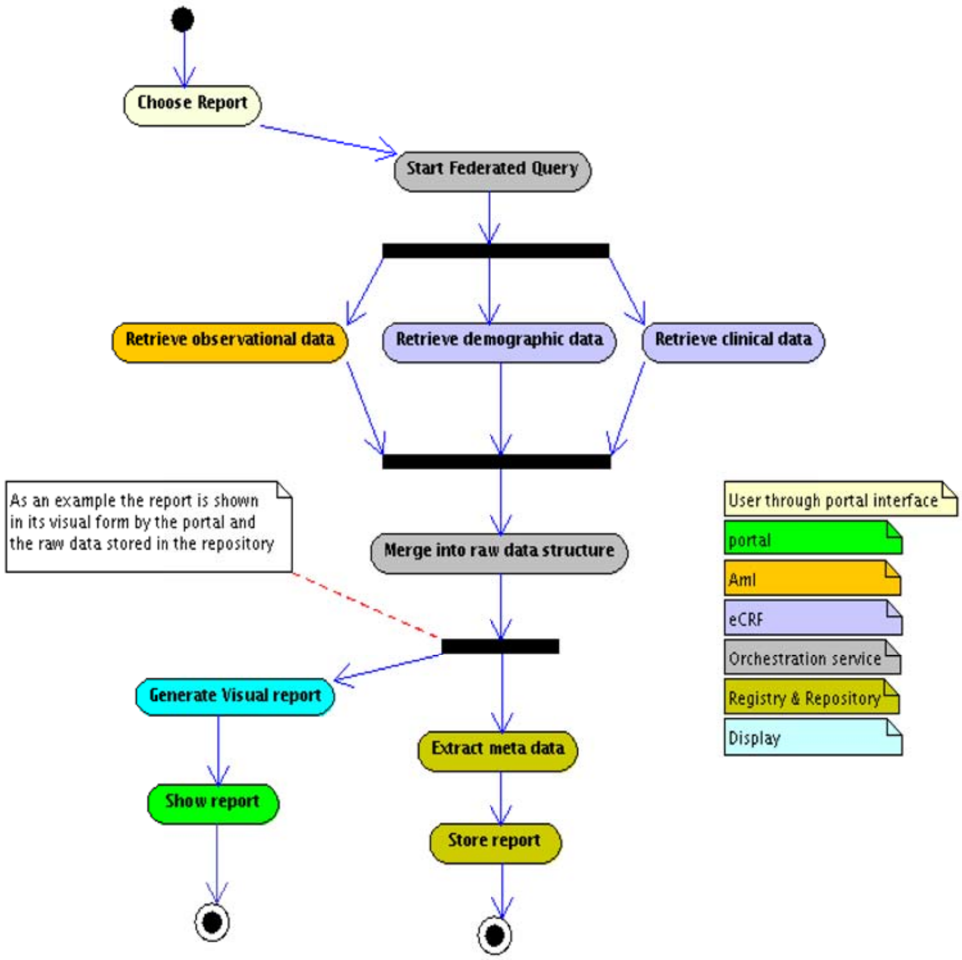


Figure 18: RTR acitivity diagram

5.6 Forward Reporting of patient data (FWR)

Description: the *FWR* is different from the RTR because it matches the *Forward Cache* integration pattern. With this approach data is pushed towards the final product (a report in this case) autonomously by the Orchestration Service (on a specific event or at a specific time schedule). The data gathered is transformed into a visual product by the Display and pushed to its final destination that could be an email address or the Repository from where it can be retrieved later.
Prerequisite: the system runs autonomously the Orchestration Service has to be made aware of the workflow of Federated Query and the events that trigger it.
Result: a report that is sent to a specific destination or saved in the Repository.
Involved modules: the same modules as in RTR are involved with exception of the Portal.

Figure 19: FWR use case

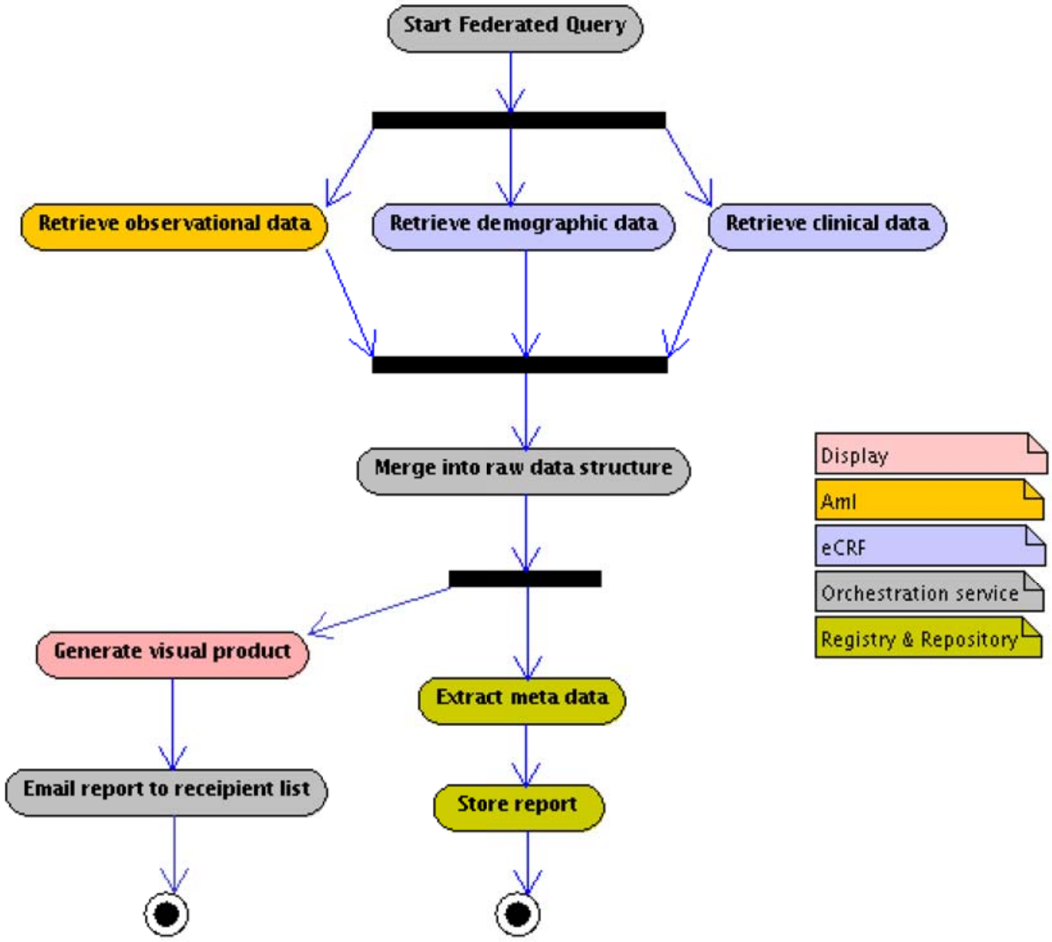


Figure 20: FWR activity

5.7 AmI Enrollment (AmIE)

Description: The Portal is able to show a list of resources (sensors, actuators, processes) that can be assigned for home monitoring. The user can select a subset of resources to assign to the patient in the context. This assignment list is then sent to the AmI application module that will update the list of associations. It should also be possible to query the current assignments of the patient for for reassignment or unrollment.
Prerequisite: Login and SPC have to be executed.
Result: a list of assignment of resources are associated with the patient inside the AmI platform.
Involved modules: Portal, AmI.

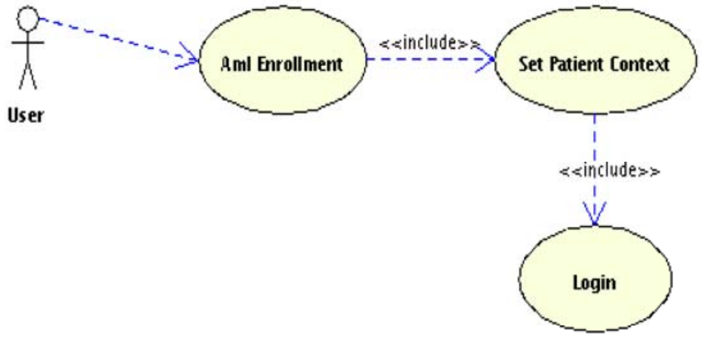


Figure 21: AmI Enrollment use case

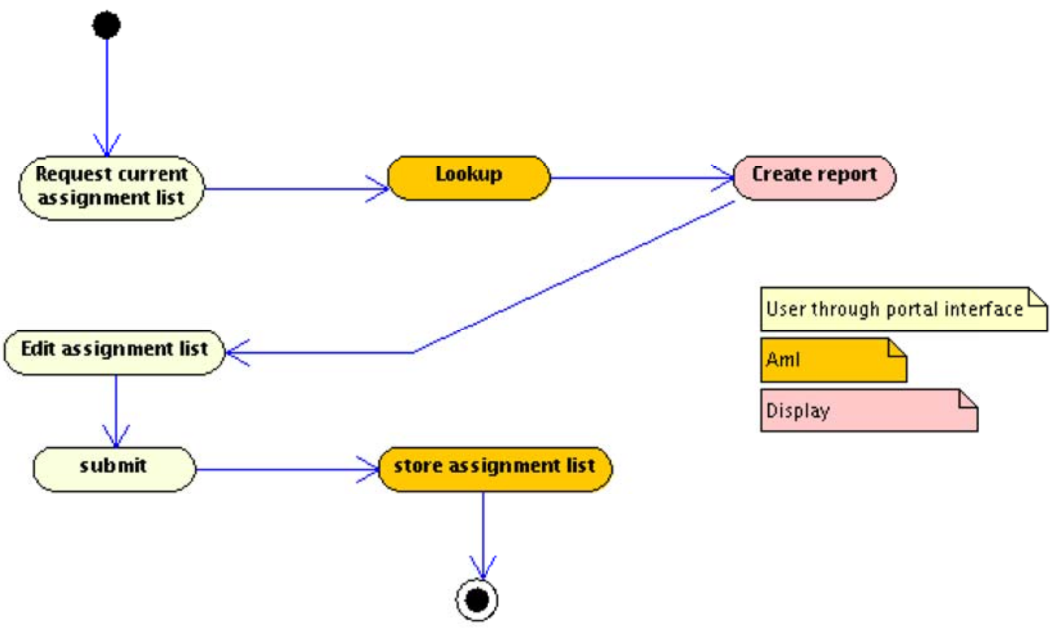


Figure 22: AmI Enrollment activity diagram

5.8 Alarm System Enrollment (ASE)

Description: The Portal is able to show a list of alarm resources that can be configured and activated. The user can select a subset of alarms to be activated for the patient in the context. Alarms typically involve different modules that play the role of value producers, decision support, reporting, storage and notification. Thus the configuration of an alarm mainly consists in the definition of workflow for the *Orchestration Service* and the triggering conditions that trigger it.
Prerequisite: Login and SPC have to be executed.
Result: a new workflow description for the *Orchestration Service*.
Involved modules: Portal, Orchestration Service.

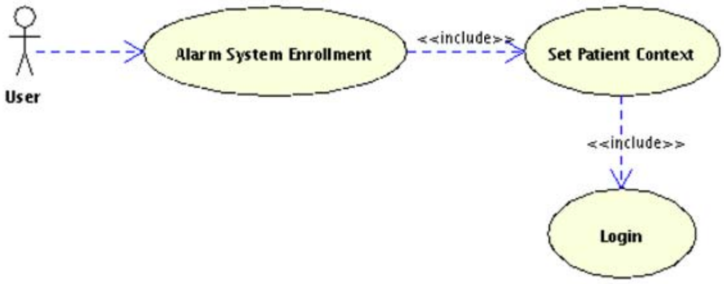


Figure 23: ASE use case

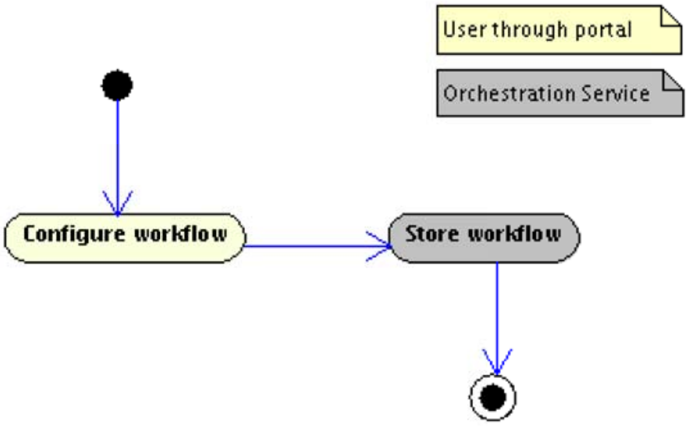


Figure 24: ASE activity diagram

5.9 Alarm handling (AH)

Description: the *AH* use case consists in the execution of a workflow specification that has to be configured during an *ASE* use case upon the verification of the trigger event. The *AH* is a completely autonomous use case. After gathering all the needed information the cDSS module is asked to verify the input condition against the configured ontology for checking whether an alarm has actually to be produced. If an alarming condition could be verified then the workflow provides a description of what modules have to be alerted.
Prerequisite: an alarm has to be configured by an *ASE* use case.
Result: the result is described by the configuration of the orchestration service it could be the notification of emails or SMS to a configured participant list, a report stored in the Repository, a triggering message for an AmI actuator, a flag set in the clinical database of the eCRF ...
Involved modules: Orchestration Service, all application modules.

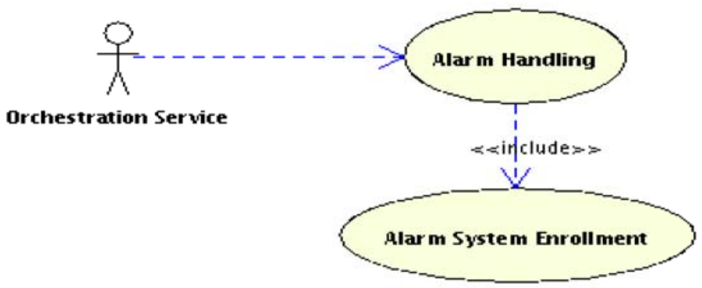


Figure 25: AH use case

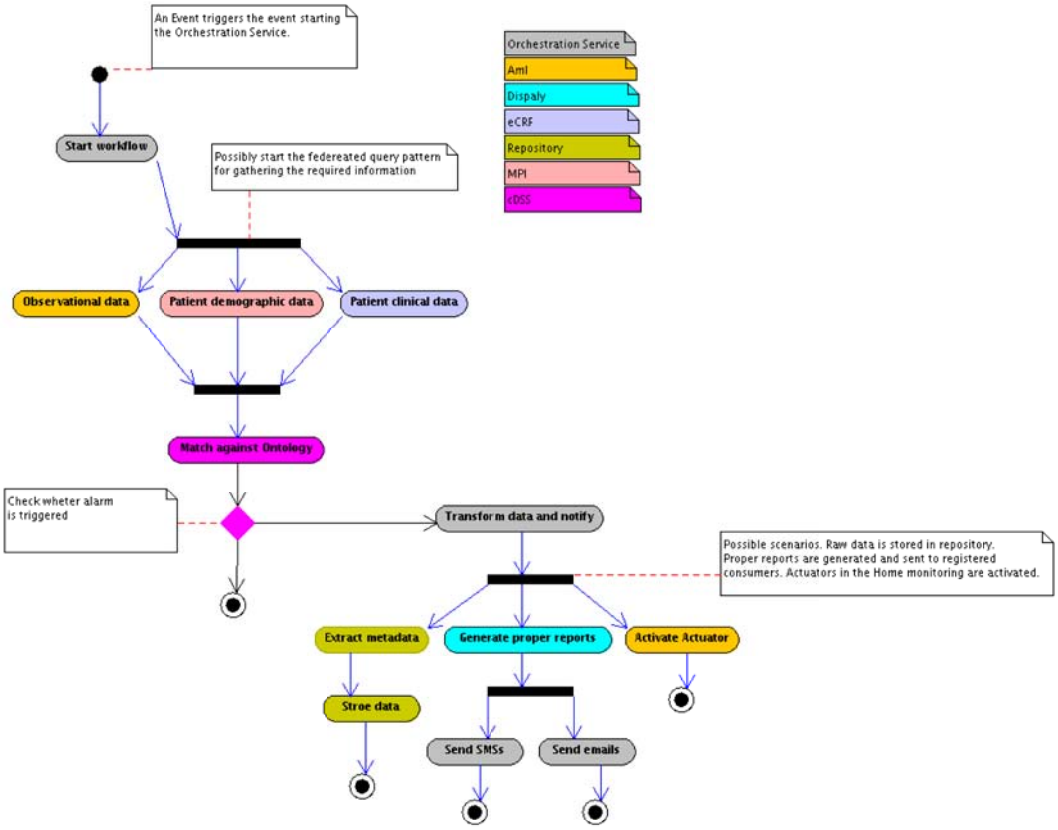


Figure 26: AH Activity

5.10 Image Upload

Description: the IU use case consists in allowing an authorized and logged user to upload a DICOM image to the image repository that is maintained in the middleware.
Prerequisite: the user has to be logged and authorized.
Result: a DICOM image is stored in the Repository and its metadata are kept in the registry for retrieval.
Involved modules: Portal, Repository, Registry.

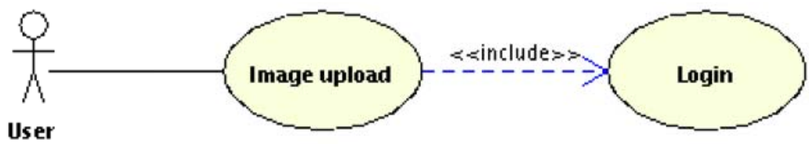


Figure 27: Image Upload usecase

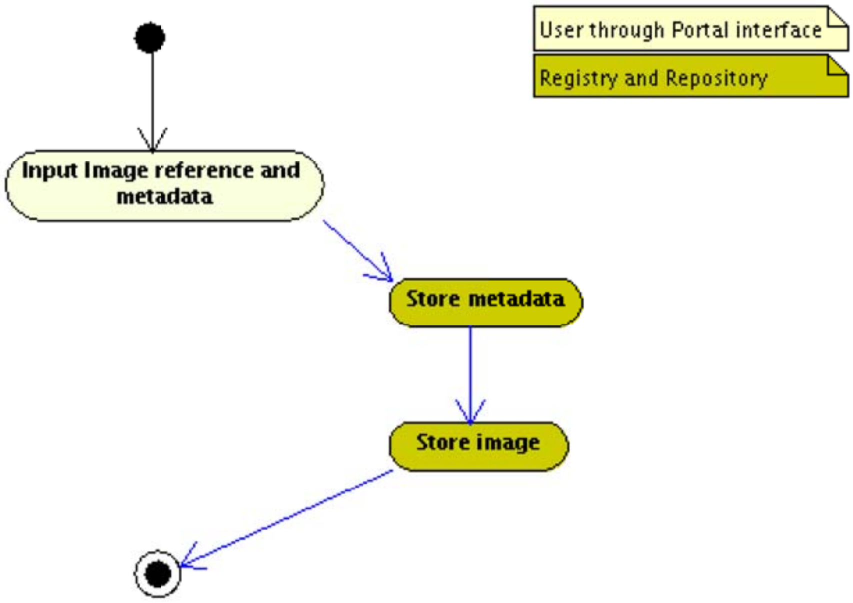


Figure 28: Image upload activity diagram

6. Conclusions

This document outlined the work that has been done in identifying standards for integrating clinical applications and to deliver a sound architecture, implementation guidelines and some concrete use cases of the Heartfaid Integration Middleware.

Standards for clinical and imaging information (OpenEHR, HL7, MML, DICOM), integration and nomenclature (IHE, Snomed-CT, LOINC) have been identified that will be useful for realizing a common language for the integration of all the different modules of the Heartfaid project.

Each module implements a specific functionality of the overal Heartfaid system. The **AmI** is an Ambient Intelligence framework that handles the storage and monitoring of observational data acquired by sensors and interacts with environmental actuators. The **eCRF** module is a web based Electronic Patient Record (EPR) that offers native user interfaces for input and edit patient related demographic and clinical information. The **groupSMS** is an application for sending short messages on mobile phones with handling of address lists and accounting. The **CDSS** is the code of a Clinical Decision Support System based on a clinical ontology of the system. The **HFP** is an image analysis and archiving toolkit used for implementing algorithms for the analysis of clinical images.

More modules will be added for implementing functionalities required for proper integration: a patient identification module (MPI), an orchestration service, an enterprise portal, a documental repository and a metadata based registry.

To integrate this big set of functionalities, services and application modules an in-depth study of state-of-the art computer science literature has been carried out. Many patterns, best practices and solutions such as *Message Oriented Middleware*, *Service Oriented Architecture*, *Enterprise Portals and Enterprise Service Bus* have been isolated and will be adopted for the implementation of the Heartfaid system.

Once decided upon the technological aspects the document shows the intensive study of all the architectural aspects that led to the design of the expected Heartfaid middleware architecture which is based on a bus-centric communication topology.

To prove the effectiveness of the integration proposals, a subset of the possible use cases has been selected for a early mockup implementation. The mockup consisted in the integration of sensing facilities installed in the home of a patient with the ambient intelligence core provided by Synapsis and with the alarming and alerting system provided by FORTH and it already showed a consistent application of concepts such as CBR, data transformation, configurable orchestration, MoM and HTTP.

To prove the soundness of the architecture several explicative and exhaustive use cases have been selected from the overall set of use cases that the Heartfaid project plans to address. Each use case has been described accurately in terms of descriptions, prerequisites, involved modules and with the drawing of UML *use cases* and *activity diagrams*.

The selected use cases have been identified as the basic functionalities that the platform should guarantee in order to support the workflows described in the HEARTFAID Scenarios document. The first mock-up presented in section 4 and the study described in Section 5, show that the proposed architecture is adherent to the functional specifications of the HEARTFAID platform.
This allows us to go on with the successive implementation phase and to move from the first mock-up to a first prototype of the entire middleware and, after that, to a prototype of the whole HARTFAID platform with a preliminary set of services and applications for the end users.

7. Bibliography and References

[1] **Data integration through database federation** by L. M. Haas, E. T. Lin and M. A. Roth
[2] **Enterprise Service Bus** by David A. Chappel. O'Reilly Theory in practice Series, June 2004 first edition
[3] IDC, *Integration Standard Trends in Program Development: It All Depends on What the Meaning of "Open" Is*, November 2003 (document #30365), *http://www.idc.com*
[4] **Enterprise Integration Patterns**, Designing, Building, And Deploying Messaging Solutions by G. Hohpe & B. Woolf. The Addison Wesley Signature Series
[5] http://mule.mulesource.org/display/MULE/Home
[6] http://www.python.org
[7] http://www.perl.org
[8] http://www.ruby-lang.org
[9] http://www.adobe.com/products/flash/
[10] http://java.sun.com
[11] http://java.sun.com/products/jms/
[12] http://www.w3.org/TR/**soap**/
[13] http://www.opengeospatial.org
[14]http://www.oasis-open.org/committees/download.php/3343/oasis-200304-wsrp-specification-1.0.pdf
[15]http://www.oasis-open.org/committees/download.php/18617/wsrp-2.0-spec-pr-01.html
[16]http://www.jcp.org/en/jsr/detail?id=168
[17]http://www.jcp.org/en/jsr/detail?id=286
[18] http://www.jboss.org
[19] http://java.sun.com/javase/technologies/core/mntr-mgmt/javamanagement/
[20] http://www.w3.org
[21] http://celtix.objectweb.org/
[22] http://mule.mulesource.org/display/MULE/Home
[23] http://www.iona.com/products/artix/
[24] http://www.sonicsoftware.com/index.ssp
[25] http://www.capeclear.com/
[26] http://www.openehr.org/
[27] http://www.ihe.net/
[28] http://www.hl7.org/index.cfm
[29] http://www.medxml.net/worldwide/index.htm
[30] www.openecg.net
[31] http://**ecg**.mit.edu
[32] Wojciech Zareba M.D., Ph.D. (2001)
New Era for Digital ECG: *FDA Requires Digital ECG Submission for Tested Drugs*
Annals of Noninvasive Electrocardiology 7 (1), 1–3.
[33] http://reference.wolfram.com/mathematica/ref/format/EDF.html

[34] DICOM: http://medical.nema.org/

[35] Bas Revet. *DICOM Cook book.* Philips Medical Systems, ftp://ftp.philips.com/pub/pms-3003/DICOM_Information/CookBook.pdf, 1997

[36] E. Cordonnier (ETIAM, France) *WADO presentation*: medical.nema.org/dicom/workshop-03/pres/cordonnier.ppt, 2003

[37] IHE Echocardiography *Workflow and Evidence Documents Integration Profiles*, http://www.ihe.net/Cardiology/committees/upload/IHE\%20Cardiology\%20Echo\%20white\%20paper\%20March\%202006\%20v.5.pdf, 2006

[38] *IHE Cardiology Technical Framework* Vol.1, http://www.ihe.net/Technical_Framework/upload/ihe_CARD_tf_vol1_2.pdf, 2006

[39] *IHE Cardiology Technical Framework* Vol.2, http://www.ihe.net/Technical_Framework/upload/ihe_CARD_tf_vol2_2.pdf, 2006

[40] http://www.bluetooth.com/bluetooth/

[41] Hearfaid Deliverable D11

[42] http://www.3gpp.org

[43] http://www.iso.org

[44] http://www.ietf.org