Deliverable 7.1: Report on overall system design including VPH-Share D2.2 and indicating its impact

Grant Agreement Number:	270089
Deliverable number:	D7.1
Deliverable name:	Report on overall system design including VPH-Share D2.2 and indicating its impact
Contractual submission date:	30/09/2011
Actual submission date:	30/09/2011

Dissemination Level

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

Cover and control page of document

Project Acronym:	p-medicine
Project Full Name:	From data sharing and integration via VPH models to personalized medicine
Deliverable No.:	D3.1
Document name:	Report on overall system design including VPH-Share D2.2 and indicating its impact
Nature:	R
Dissemination Level:	PU
Version:	0.4
Actual Submission Date:	30/09/2011
Editor:	Manolis Tsiknakis
Institution:	FORTH
E-Mail:	tsiknaki@ics.forth.gr

R=Report, P=Prototype, D=Demonstrator, O=Other

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

Abstract

This document describes preliminary requirements and a top-level design for a repository for securely storing and maintaining data from diverse sources integrated semantically, for input to diverse modelling and data mining tools which aim to enable personalised medical treatment.

Keyword list

data warehouse, semantic integration, ontology, requirements, system design, VPH-Share

Modification control

Version	Date	Status	Author
0.1	14/09/2011	Draft	Benjamin Jefferys
0.2	23/09/2011	Draft	Benjamin Jefferys
0.3	29/09/2011	Draft	Benjamin Jefferys
0.4	30/09/2011	Final	Benjamin Jefferys

List of contributors

- Benjamin Jefferys, UCL
- David Chang, UCL
- Miguel García-Remesal, UPM
- Alberto Anguita, UPM
- Juliusz Pukacki, PSNC
- Elias Neri, Custodix

Contents

Glossary

BBC *British Broadcasting Corporation* - British public service broadcaster, and the largest broadcasting organisation in the world

CDISC *Clinical Data Interchange Standards Consortium* - a global, open, multidisciplinary, non-profit organization that has established standards to support the acquisition, exchange, submission and archive of clinical research data and metadata

CDMI *Cloud Data Management Interface* - defines a functional interface that applications can use to create, retrieve, update and delete data elements from the Cloud

CSS *Cascading Style Sheets* - a simple mechanism for adding style (for example fonts, colours, spacing) to Web documents

CSV *Comma Separated Value*

DICOM *Digital Imaging and Communications in Medicine* - a standard for handling, storing, printing, and transmitting information in medical imaging

GridFTP *Grid File Transfer Protocol* - an extension of the standard File Transfer Protocol for use with Grid computing

GUI *Graphical User Inteface*

HL7 *Health Level Seven (International)* - the global authority on standards for interoperability of health information technology with members in over 55 countries

HPC *High Performance Computing*

HTTP(S) *HyperText Transfer Protocol (Secure)* - the standard networking protocol for distributed, collaborative, hypermedia information systems, including the world wide web

IHE *Integrating the Healthcare Enterprise* - an initiative by healthcare professionals and industry to improve the way computer systems in healthcare share information

JEE *Java Platform Enterprise Edition* - a platform for server programming in the Java programming language

JMX *Java Management Extensions* - a Java technology that supplies tools for managing and monitoring applications, system objects, devices such as printers, and service-oriented networks

LOBCDER *Large Object Cloud Data storage federation* - in VPH-Share, will provide reliable, managed access to large binary objects that can be stored across various storage frameworks and providers, exposing these different storage mechanisms as unified resource under a single namespace

PSNC *Poznan Supercomputing and Networking Center*

RDF *Resource Description Framework* - a general method for conceptual description or modeling of information that is implemented in web resources

REST(ful) *Representational State Transfer(ful)* - a style of software architecture for distributed hypermedia systems, the largest implementation of which is the World Wide Web (*ful* suffix makes an adjective describing systems conforming to this style)

SCP *Secure Copy* - a means of securely transferring computer files between a local and a remote host or between two remote hosts

SPARQL *SPARQL Protocol and RDF Query Language* - query language for RDF

URI *Uniform Resource Identifier* - a compact sequence of characters that identifies an abstract or physical resource

UCDMC *University of California Davis Medical Center*

VM *Virtual Machine* - a completely isolated guest operating system installation within your normal host operating system

WebDAV *Web-based Distributed Authoring and Versioning* - a set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web servers

XML *Extensible Markup Language* - a format for encoding documents in machine-readable form, similar in syntax to HTML

1 Requirements

Formal requirements gathering for p-medicine is not yet complete, therefore the design of the warehouse is somewhat speculative. We define here some preliminary requirements which our design is based upon.

A data warehouse is a repository for securely storing and maintaining data from diverse sources integrated semantically to enable reporting and analysis. It is not typically intended to be used as a live data store. Live data stores often purge old data which may be useful for analysis, and they need to support operational systems which have different quality of service needs.

The p-medicine data warehouse must collect data for input to diverse modelling and data mining tools which aim to enable personalised medical treatment. The Description of Work defines the **objectives of Work Package 7** as follows:

1. To develop **federated data warehouse** infrastructure components to store the **multiple different types** of medical data produced in this project
2. To develop **storage services for large data objects**
3. To develop mechanisms for ensuring **reliability** and **auditability** of the data

4. To develop and deploy suitable **programmatic interfaces**, to allow the different types of data to be uploaded to the warehouse via tools developed in other work packages
5. To develop a **service integrated into the web portal** to allow users to search for and view available data
6. To develop federated **capability-based secure access mechanisms** compliant with the legal and ethical framework of the project
7. To integrate **security- and role-based access**, based on the legal and ethical framework developed in the project
8. To **integrate the disparate types of data** produced and available in the project using appropriate ontological tools

1.1 Nature of data to be stored

Goals 1, 2 and 8 refer to the kinds of data to be stored by the warehouse. Early stages of requirements gathering identified the following characteristics:

1. Sources
   (a) Clinical trials
   (b) Patient information systems
   (c) Experimental work
   (d) p-medicine tools
   (e) Obtima clinical trial management system
   (f) Modelling and data mining tools
   (g) Patient empowerment tools
2. Subjects
   (a) Patients:
      i. Patients in clinical trials
      ii. Patients being treated but not in clinical trials
      iii. Patients treated for Acute Lymphoblastic Leukaemia
      iv. Patients treated for breast cancer
      v. Patients treated for nephroblastoma
      vi. Patients treated for any other disease (long-term goal)
   (b) Doctors involved in clinical trials
   (c) Relatives and others involved in patient care in general
   (d) Individuals involved in data mining, modelling and development of p-medicine

   (e) Auditors and others involved in managing data

3. Types
   (a) Case report forms
   (b) Imagery and other large medical datasets
   (c) Sequencing, microarray and biomarker experiments
   (d) Structured data

4. Formats
   (a) Access databases
   (b) Excel spreadsheets
   (c) CSV files
   (d) Relational databases
   (e) Generic and domain-specific image formats
   (f) Domain-specific binary and text files

5. Standards
   (a) *Ad hoc* standards specific to data source
   (b) Structure standards such as CDISC
   (c) Terminology standards such as HL7
   (d) Ontologies such as Gene Ontology
   (e) Multiple languages
   (f) Multiple interpretations of standards

It is clear from this list that warehousing such data is a significant challenge. Strictly speaking, in the context of p-medicine, the standardisation process is not the responsibility of the data warehouse. However its architecture should support processes which make standardisation easier and more complete.

1.2 Core data stores

The p-medicine description of work describes three different core types of data store which will store data for analysis:

1. DICOM infrastructure for storing medical imagery
   (a) Required for integration with DICOM infrastructure in hospitals
   (b) Should make use of the full range of DICOM server features, including metadata store, to ensure useful integration possibilities

   (c) Standards are defined by DICOM

2. Generic file store for storing other files, possibly very large

   (a) Relationships between files and other information about files is expressed in the index in the structure data store, to avoid duplication of information between that and any structure (most commonly hierarchical) functionality available in the file store

3. Structured data store for storing data from three sources:

   (a) Extracted from files and images submitted to the warehouse

   (b) As a distributable index for those files, since the warehouse is to be federated (see below)

   (c) As a means for standardising the data

      i. Programmatically input into the warehouse

      ii. Stored as a result of interactions with the warehouse for the purpose of auditing

         A. Log of transactions

         B. Provenance information

         C. Annotations and rating

         D. Historical data

4. All three data stores must ensure a very high quality of service, particularly regarding speed of response to queries, and download of data

1.3 Federation

A key concept in the description of work is federation. Federating data storage across many systems addresses the following requirements:

1. **Scalability** of storage capacity, important for **storing large amounts of data**

2. Distribution of server load, important for high **quality of service**

3. Distribution of responsibility for storing data, which adds a small amount of **additional security** for very sensitive data

4. Optional use of **cloud technologies** to support these requirements

5. **Unifying multiple data stores** under a single view by maintaining lists of servers to which requests can be delegated

6. Maintaining **local copies of remotely-held data** to further ensure quality of service

1.4 Security

The sensitive nature of the data to be stored makes security a high priority. Work Package objectives 6 and 7 are reliant upon the legal and ethical framework of p-medicine, which has yet to be defined. Therefore security considerations are an open question. However, there are some obvious requirements:

1. Data must only be made available to those with permission to access it
2. The only people with the ability to grant permission to access data are the people who provided it: data providers retain control of the data, if they wish to
3. Data providers will give permission to final users for access to data, rather than to the p-medicine project or tool authors
4. The negotiation of access to data sources is outside the scope of the technology within p-medicine, since it is likely to involve real-world legal processes - however, we must provide interfaces and workflows to support these processes
5. Certain p-medicine participants (particularly engineers involved with writing software and maintaining systems for the warehouse) and external third parties (legal and ethical auditors) will have and/or need access to all data
6. Development of analysis tools may require blanket access to large quantities of data
7. These requirements somewhat dictate the granularity of the security policies we put in place

External to the data warehouse, prior to data submission, a data pseudonymisation and patient identity unification service will optionally "clean" the data. This, in theory, somewhat reduces the security issues with the data warehouse. However, the EU considers pseudonymous data to be as sensitive as personal data, therefore the security system **must be robust enough to deal with personal data** anyway.

1.5 Integrating disparate types of data

We will take data from many diverse sources, discussed above. In fact, the task here is much harder than that for most business data analysis, since we will take data from many organisations, in many languages, and for many complex diseases with a diverse range of tests, measurements and treatments associated. Integrating this data is the task of Work Package 4, however we must support the requirements of that work package.

1. Use ontologies to describe and integrate data from structured sources (for example, Access databases, relational databases, Excel spreadsheets, and so on)

2. Use ontologies to describe, integrate and index the non-structured files stored in the warehouse (for example, text files, images)
3. Allow continuous revision of ontologies and mapping from input binary and structured data to integrated data, to ensure high quality integration
4. Provide data integrated at schema-level that can be used with data mining and modelling tools without concern for data integration issues: data will not necessarily be integrated at data-level, for example, measurement units will not be standardised
5. Allow authorized users to create ontology-based annotations of the data sources to be stored in the warehouse

1.6 Warehouse integration and interfaces

Task 7.5 is dedicated to establishing the internal and external interfaces for the warehouse.

1. Integration between components
   (a) Secure
   (b) Enable distributed ontology-based search and access to data at other warehouse nodes
2. Programmatic interface
   (a) Ontology-based search and access to data
   (b) Advanced querying to specify desired subset of data with simple result formatting
   (c) Additional query result formats to be specified by data mining and modelling work packages
   (d) Add and edit data
   (e) Support requirements of tools developed in other work packages (patient empowerment, clinical decision-making, and so on)
   (f) Via HTTPS to circumvent firewalls
3. User interface
   (a) Browse data in warehouse
   (b) Search data
   (c) Integrated into web portal in WP8
   (d) Adhere to web standards: HTML, CSS, usability
   (e) Key users:
      i. Tool and model developers
      ii. Auditors

4. Direct interface to large object stores
    (a) Efficient access to very large files on federated servers
    (b) Integration with existing infrastructure in clinics
    (c) DICOM standards for access to medical images
    (d) One or more standard file access protocols for access to other binary objects

1.7 Reliability, auditability and so on

Objective 3 expresses the desire for the data warehouse to be a secure of source high-quality, standardised data which can be audited to ensure these attributes are maintained. In general we have interpreted this as the need for a "human factor" in the warehouse - complex data integration and security cannot be accomplished purely programmatically. The following requirements will help to ensure a high quality, reliable resource:

1. Logging
    (a) Log transactions (searches, access, edits)
    (b) Allow security auditing
    (c) Allow optimisation of data distribution and organisation
2. Annotation
    (a) Rating and commenting
    (b) Allow feedback and querying on specific data items by:
        i. Auditors
        ii. Patients
        iii. Data providers
        iv. Data consumers
3. Provenance
    (a) Track where data has come from
    (b) Provide an audit trail outside of the warehouse
4. History
    (a) Maintain historical versions of binary and structured data
    (b) Allow rollback and selective undo/redo of changes
    (c) Closely related to logging
5. Backups

(a) Snapshot backups of data on each warehouse node in case of catastrophic data loss
(b) Follow standard backup protocols (on-site fast restore, off-site historic archives)

6. File and image integrity checks
   (a) Store a small checksum as part of metadata for each file
   (b) User can check data against checksum before using a downloaded file
   (c) Regular checks of all files

1.8 Integration with VPH-Share

Where possible, services (web interfaces, structured, binary file and image stores) must be deployable on cloud infrastructure. Interfaces between components must take into account this requirement. Storage services managed by PSNC are not cloud-based, but the option of using cloud-based storage must exist. p-medicine's sister project, VPH-Share, is developing cloud infrastructure which, where possible, should be exploited such that p-medicine becomes a key external user of the cloud infrastructure developed by VPH-Share. The VPH-Share cloud infrastructure is summarised in Section 2.

2 VPH-Share Deliverable 2.2

2.1 Overview

The aim of VPH-Share work package 2 (WP2) is to deliver a service-based cloud computing platform and access to high performance computing infrastructure that is integrated into the VPH-Share workflow, or any applications which may require the usage of VPH-Share resource.

The aim of deliverable 2.2 (D2.2) from VPH-Share is to set out a detailed and in-depth description of the users, workflows, methodologies and tools that constitute the architecture of VPH-Share WP2. This section will outline the general architecture, as well as design and purpose of individual components and tools, and comment on the possible impact on the design of the p-medicine data warehouse.

The general architecture and workflow of WP2 can be described in Figure 1. A central concept of the WP2 architecture is the *atomic service*. Components and applications are to be treated as a service if they are to be managed by the WP2 component and deployed in a cloud. This requirement places several constraints on how an application is to be deployed. Ideally, applications should expose a remote interface based on web services technology. However for legacy applications, it is unrealistic to expect that developers will re-implement tools to suit VPH-Share requirements. As such, these legacy applications will be wrapped by an external client which will provide the web service interface and translate those commands into the internal legacy application calls. A third scenario of application deployment is using remote GUI

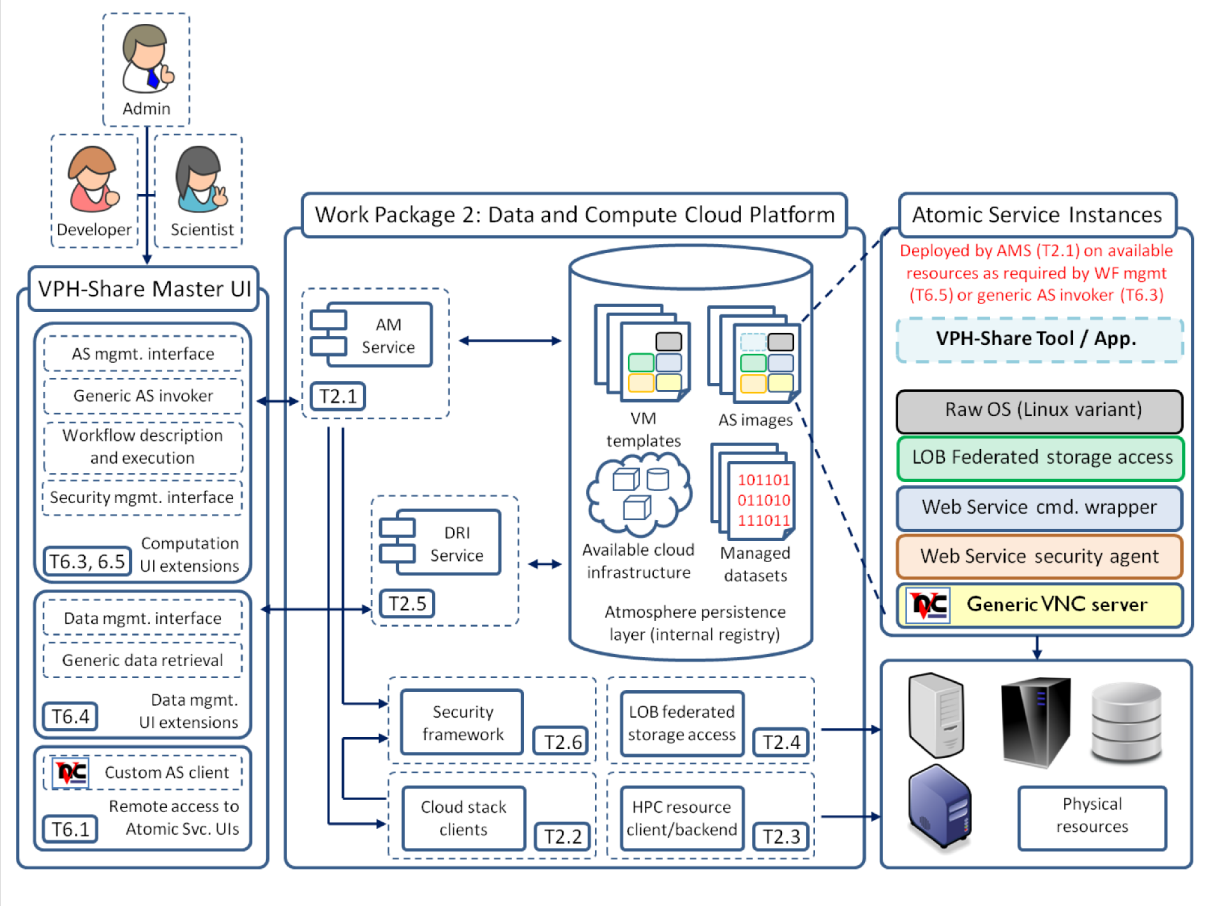

Figure 1: Overall architecture of the VPH-Share Data and Compute Cloud Platform (VPH-Share Work Package 2) and its relation to external project components

using a remote desktop mechanism for applications that do not normally expose command line interfaces.

With a web services-enabled application, a developer can then choose a specific operating system template to install the application which will come with preinstalled VPH-Share cloud component. Once the application has been installed, the atomic service can be registered and stored in an internal registry called *atmosphere*. This atomic service is then available to use through the VPH-Share Master User Interface where end users can dynamically instantiate a copy of that VM.

The VPH-Share WP2 architecture consists of several components. These include:

- Cloud resource allocation management
- Cloud application deployment and execution component
- Access to high performance computing environment
- Access to large binary data in the cloud
- Data reliability and integrity component
- Cloud security component

The cloud technology to be used includes the OpenStack cloud framework with a preference of Amazon as the preferred commercial cloud provider, due to its status as the *de facto* industry standard in cloud technologies. OpenStack is an open source cloud computing system developed by NASA and hosting company RackSpace, and many other large IT companies are involved. OpenStack consisted of three components:

Compute provides large networks of virtual machines to ensure reliability through redundancy, and performance through rapid scalability. Software and user interfaces are provided for managing the VM cloud, and multiple virtual machine managers are supported.

Swift provides an interface to redundant and scalable object storage of petabytes of accessible data. It is intended as a long-term storage system for essentially static data, rather than a live file system for real-time data storage.

Glance provides discovery, registration, and delivery services for virtual disk images

2.2 Cloud Resource Allocation Management

The *cloud resource allocation management* is responsible for ensuring that all atomic service instances are allocated sufficient cloud or HPC computational resources. An atomic service instance can be defined as a computer system with a VPH-Share application installed with web service interface available.

2.3 Cloud application deployment and execution

The *Allocation management service (AMS)* provides system administrators with the right tools to perform their duties such as shutting down, initializing, configuring and monitoring. Such control of the VM instances will minimise the cost of hosting the instance. The AMS will provide a mechanism for turning applications into an atomic service instance, host atomic services in private and public cloud infrastructure, as well as API to manage these services.

2.4 Access to high performance computing environment

Through the use of grid infrastructure, High Performance Computing (HPC) will be provided for applications which require computational resources greater than that provided by the cloud infrastructure. To simplify access to the grid infrastructure, a light weight middleware will be introduced. The Application Hosting Environment (AHE) and Audited Credential Delegation (ACD) will expose the grid infrastructure as web services and simplify the security mechanism. These will be deployed as an atomic service instance and integrated into the overall WP2 architecture.

2.5 Access to large binary data in the cloud

Accessing large binary data in a cloud environment is a major issue. To tackle this problem an efficient and transparent storage mechanism is required. VPH-Share WP2 introduces the Large Object Cloud Data storage federation (LOBCDER) to provide reliable, managed access to large binary objects that can be stored across various storage frameworks and providers, exposing these different storage mechanisms as unified resource under a single namespace. This allows dynamic addition and removal of storage resources, improves collaboration through data sharing, and simplifies development of scientific applications. LOBCDER will provide interface in the form of a WebDAV server and a common interface to simplify the design and maintenance overhead.

2.6 Data reliability and integrity

There is a need in the cloud data infrastructure to perform auditing tasks in a certified and trustworthy manner to ensure the reliability and integrity of the biomedical datasets used in the VPH-Share project. To achieve this, D2.2 describes components to perform the following tasks:

- Periodic integrity checks on data with use of hash algorithms
- Storage of multiple copies of data on various cloud platforms
- Tracking history and origin of datasets.

The data reliability and integrity component will introduce the concept of a *managed dataset*, which is known to the VPH-Share service and is referenced in the metadata registry. This will allow the VPH-Share service to track and manage the dataset in an automatic manner.

2.7 Security for Cloud applications

The VPH-Share security platform consists of three components:

Security management framework which contains the security management tools for system administrators and application developers. There will be one instance of security management framework installed in the overall VPH-Share deployment.

Security proxy which intercepts and protects the incoming and outgoing traffic to and from the virtual appliance

Security Agent module which provides the security proxy with the necessary tools to enforce the authentication and access control rules of users.

Each agent and proxy will be installed per virtual appliance.

3 System design

See Figure 2 for an overview of the major components and relationships between them.

3.1 Generic file store

The generic file store will be able to store any kind of binary data. It will not have any concerns with data integration or other semantic matters. Files in the store will be referred to by URI, since many federated file stores may exist and they may be referred to by the structured data in other data warehouses. There is no specific need to keep any information about the file beyond its name (which may be a simple reference code) and the content of the file. File metadata and relationships between files (resembling a hierarchy) will be stored in the structured data store. The file store may be deployed on cloud infrastructure, and must support this as well as deployment on standard servers.

The file store should offer direct, secure access to files through a set of standard file access protocols, for example CDMI or WebDAV. The need for data to only be accessible to those with permission means that the security mechanism for the file access protocols must be integrated with that of p-medicine. It must refer to the Custodix identity management system for identity information, and to information stored in the structured data store on who is permitted to access what data. Therefore, a file storage solution must be chosen which can be adapted in this way.

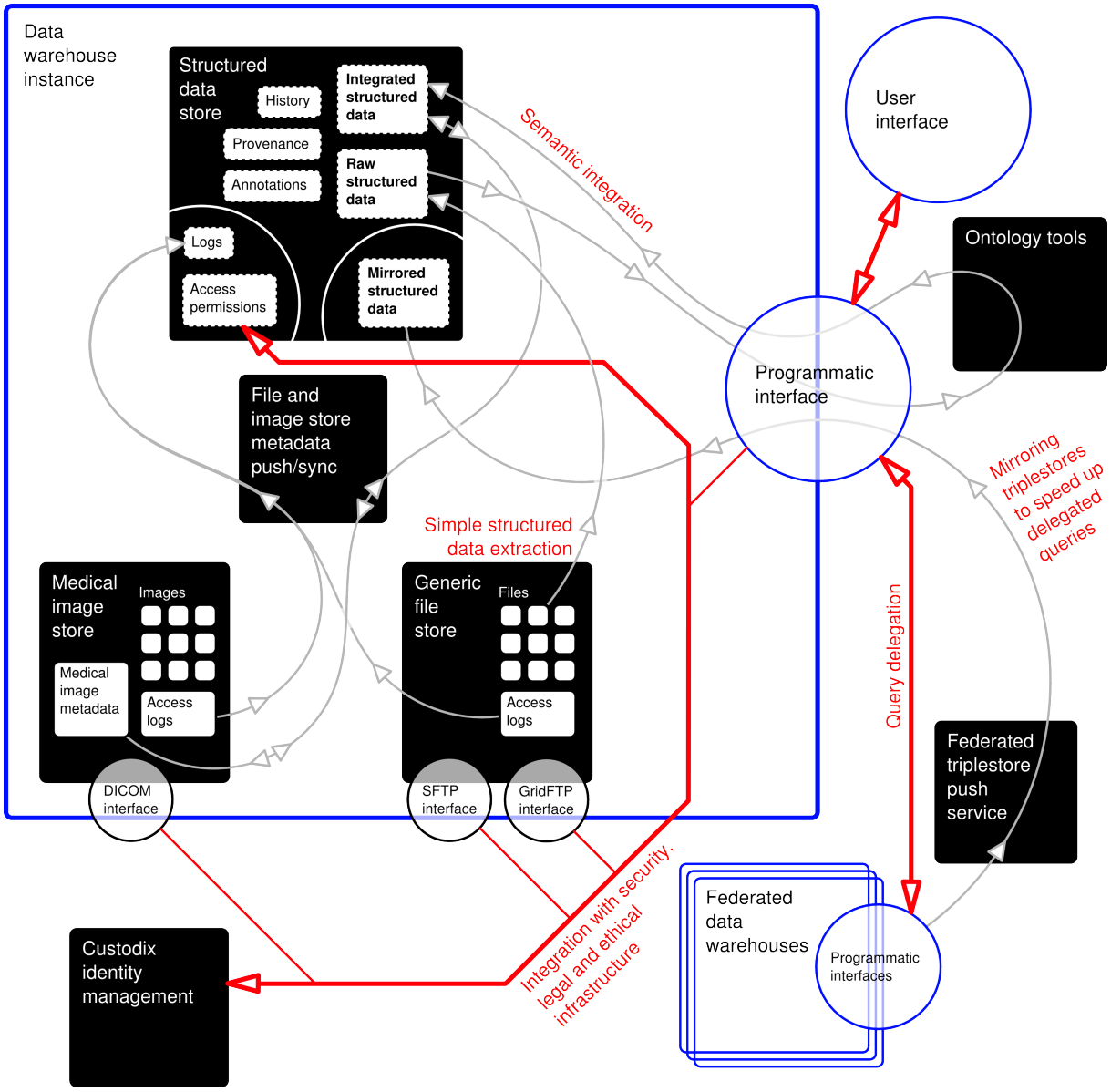

Figure 2: Annotated diagram showing the major components and services of a data warehouse, its relationship with other warehouses, and the semantic and identity services. Grey lines show major data flows. Red lines show the most important links between components.

Most file servers will have some logging capability built in. This must be integrated and augmented with the p-medicine-specific logger. File annotations, provenance and history will be maintained outside of the file store.

PSNC will provide storage from the National Data Storage (NDS) project[1]. This is intended to build a storage system distributed across Poland, that provides high efficiency of data access and I/O operations as well as high level of security, reliability and data durability. The main application of the system is the backup/archival data storage service for academic, educational, local government and other public institutions.

The NDS Backup/Archive service offers a backup/archival functionality to end-users. The service supports on-the-fly replication, encryption of the data before they leave the user system, integrity control as well as typical backup/archive scenarios such as full and incremental backup and data archival.

The NDS Virtual filesystem service provides an easy interface to store and retrieve data. The service implements the data replication, that is transparent to end-users. Users have illusion of interacting with the remote SCP, HTTP or GridFTP site, containing the virtual filesystem. This logical view hides the complexity of the physical data repository and details of data processing happening in the system such as physical data replication.

Users' data may be stored in multiple replicas located in geographically-distant NDS centers. In case of failure of default storage node, the data can be retrieved from any existing replicas. This feature is supported by both services provided by NDS.

OpenStack Swift Cloud interfaces will also be available for access. Security technologies will be adopted and used inside Swift to conform to the p-medicine's security scheme.

3.2 Medical image store

DICOM (Digital Imaging and Communications in Medicine) is the *de facto* standard for handling, storing, printing, and transmitting information in medical imaging. A typical DICOM file server fulfils most of the requirements for storing medical imagery.

As for the generic file store, images must be referred to by URI, since many federated image stores may exist. The image store should offer direct, secure access to images through the standard DICOM image access protocols. The need for data to only be accessible to those with permission means that the security mechanism for the DICOM server must be integrated with that of p-medicine. It must refer to the Custodix identity management system for identity information, and to information stored in a central data warehouse system on who is permitted to access what data. Therefore, an image storage solution must be chosen which can be adapted in this way. Most DICOM servers will have some logging capability built in. This must be integrated and augmented with the p-medicine-specific logger. Unlike the generic file store, the metadata infrastructure of DICOM will be used and maintained in parallel with the data in the structured

[1] http://nds.psnc.pl

data store. Close integration between these elements is essential, and a syncing service will be necessary to ensure data is consistent. Image annotations, provenance and history information added to the warehouse directly (rather than imported with images) will be primarily maintained outside of the file store, but synced with the DICOM server metadata where appropriate.

Currently the most promising candidate for the DICOM server is dcm4chee[2]. This is a JEE and JMX system which is deployed within the JBoss Application Server. It includes a DICOM archive and image manager and a PACS, when coupled with a viewer such as OsiriX, K-PACS, ClearCanvas. It is implemented in Java, making it fairly easy to integrate it with any Cloud technology and provide data archiving in the Cloud instead of local storage.

Several modules are available for dcm4chee:

Web-based User Interface dcm4chee contains a robust user interface for administrators which runs entirely in a Web browser

DICOM Interfaces Acting as an archive, dcm4chee is able to store, query, and retrieve any type of DICOM object. In addition, support is included for MPPS, GPWL, MWL, Storage Committment, Instance Availability Notification, Study Content Notification, Output Content to CD Media, Hanging Protocols, and more.

HL7 Interfaces The archive includes an integrated HL7 server which can act on ADT, ORM, and ORU message types.

WADO and RID Interfaces WADO (DICOM Part 18, Web Access to DICOM Objects) and RID (IHE Retrieve Information for Display) interfaces enable access to the archived content from the Web.

Audit Record Repository (ARR) IHE ATNA audit logging

Media Creation (CDW) CD Export Manager

XDS/XDS-I XDS is an IHE profile for Cross Enterprise Document Sharing. dcm4chee participates in XDS as a Document Repository, and XDS-I as an Imaging Document Source.

Oviyam Oviyam is a lightweight web viewer for clinical (that is, non-diagnostic) access to studies and images. This component is intended for users such as Nurses, Doctors and perhaps Patients who can't easily install a full Radiology client station and/or would benefit from a quick representation of the images.

Other DICOM solutions exist, which will be explored:

PacsOne Server Not fully free software, source code is not available

[2] http://www.dcm4che.org/confluence/display/ee2/Home

Conquest DICOM [3] Extends public domain UCDMC DICOM code. Open source but much less functionality than dcm4chee.

PGCTN [4] Not updated for two years, perhaps not worth investigating further. Implemented in PHP and javascript.

3.3 Structured data store

The structured data store is at the core of the warehouse, and will be used to store data fulfilling several requirements:

1. Structured data added directly to the warehouse through domain-specific web services and user interfaces
2. Structured data extracted from and relating to files added to the generic and medical image file stores, for the purpose of indexing and standardisation
3. Provenance information
4. User data annotations (comments and rating)
5. Historical data
6. Logging information
7. Security information on who can access what data

Due to the large part ontologies play in the p-medicine project, a data store which has ontologies at its heart is a sensible choice. Therefore an RDF triplestore will be used as the primary structured data store. This decision means that all structured data stored must have a terminology associated with it, either taken from existing definitions or developed for the project. Candidate ontologies have been identified in Deliverable 3.1.

The structured data stores will be federated, with data being kept primarily on the store which data was originally submitted to, and queries to a particular store being optionally passed to other stores. Caches on each store will speed up certain queries.

External services developed in other work packages will be responsible for ensuring data is fully semantically integrated with the data already in the warehouse.

Security and logging information may also be stored. However, due to the importance of such information, it must be stored in a separate store, or a substore of the main store inaccessible to normal queries. Access permissions relating to the medical image and generic file stores will be maintained here, as well as that relating to the other structured data. Security policies in the

[3] http://ingenium.home.xs4all.nl/dicom.html
[4] http://sourceforge.net/projects/pgctn/

triplestore will be enforced by the separate programmatic interface. Typically triplestores provide a web services interface for querying and changing data. However, no direct access to this interface will be available outside of the data warehousing system.

Currently the most promising candidate for the triplestore is OWLIM[5], which has been used in at least one large-scale mission critical industrial project during the BBC's coverage of the 2010 World Cup[6] and also already deployed by PSNC in other projects.

3.4 Federation and integration

Since all other services can be built upon the programmatic interface, it will be the main focus for integrating all the components of the data warehouse, including enforcement of security policies and logging. The next section deals with the programmatic interface. This section deals with components which operate in the background, independently of calls to the programmatic interface.

3.4.1 Federated triplestore push service

SPARQL queries to the programmatic interface can be delegated to other warehouses. However, this presents a possible query bottleneck, relying on further network communication and server responses before results can be given to the caller. A service will maintain a copy of all or some subset of a remote triplestore on a separate local triplestore. The operation will be one-way - no changes to a local copy of a remote triplestore will be allowed. This will allow the remote query to take place locally, avoiding additional network communication. Which remote nodes are copied, and what subset of data is copied, will be entirely configurable.

3.4.2 File and image store metadata push/sync service

The image and file stores will maintain their own access logs, and the image store will have editable metadata. It may be that the software chosen to maintain these resources can be adapted to update the central structured data store when changes are made to them. However, if not, there will be a need for a service to push changes made to these stores to the structured data store. Additionally, changes to the structured data store must be pushed to the image store, effectively keeping the two in sync.

3.5 Programmatic interface

All other services can be built upon the programmatic interface. A RESTful web services interface over HTTP Secure (HTTPS) will provide access to a set of data warehouse functions. A set of

[5] http://www.ontotext.com/owlim/editions
[6] http://www.bbc.co.uk/blogs/bbcinternet/2010/07/bbc_world_cup_2010_dynamic_sem.html

functions to be provided by the interface is described below. Security and logging information will be at a granularity that is yet to be decided, depending upon a review of the kind of data we are intending to store. Therefore, the description of these functions refer to a subgraph of the entire triplestore graph, which may be as small as a single node or triple.

All functions will be called with a verified originator identity, which will be checked with the central Custodix identity management system.

1. SPARQL query
   (a) with specified return format from RDF XML, RDF Notation 3 and CSV
   (b) must only return or summarise data which originator has permission to access
   (c) query and summary of results must be logged
   (d) A call to a given data warehouse node may be delegated to other data warehouses, and the results compiled into a single result set
      i. To speed this process up, a local copy of structured data held on a remote warehouse may be queried instead
      ii. A default set of "sister" data warehouses will be consulted, which can be added to or reduced by the caller
2. Create node(s)
   (a) generates a new unique node, or creates a node based on given URI
   (b) returns URI for node
   (c) must be logged
   (d) created node must be associated with provenance information
   (e) created node only accessible to originator
3. Add relation(s)
   (a) must only refer to nodes to which the originator has access, reporting that a node does not exist otherwise
   (b) successful or otherwise, must be logged
   (c) created relation must be associated with provenance information
   (d) created relation only accessible to originator
4. Remove relation(s)
   (a) must only refer to relation to which the originator has access, reporting that the relation does not exist otherwise
   (b) associated nodes are not removed, just the relation between them
   (c) successful or otherwise, must be logged

(d) deleted relation must be recorded in history

(e) associated provenance information must be removed with additional calls

5. Remove node(s)

   (a) must only refer to nodes to which the originator has access, reporting that the node does not exist otherwise

   (b) must only refer to nodes with associated edges to which the originator has access, failing otherwise

   (c) associated edges are removed with additional calls

   (d) successful or otherwise, must be logged

   (e) deleted node must be recorded in history

   (f) associated provenance information must be removed with additional calls

6. Upload RDF file

   (a) XML or Notation 3

   (b) parses file and generates appropriate calls to other functions

   (c) a single instance of provenance information is associated with all created triples

7. Upload file request

   (a) Allocates a URI to which the file should be uploaded

   (b) Creates minimal set of metadata (perhaps based upon passed parameters)

8. Upload image request

   (a) Allocates a URI to which the image should be uploaded

   (b) actual file upload occurs directly to the file store after this function is called

9. File access request

   (a) returns true if originator has permission to access file contents

   (b) actual file access occurs directly to the file store, and this function must only be used by the file store

10. File delete request

    (a) returns true if originator has permission to delete a file

    (b) actual file deletion occurs directly on the file store, and this function must only be used by the file store

11. Functions relating to security permissions (these specific functions will not be provided if security information is fully integrated with the triplestore, in which case the generic triple editing functions can be used in conjunction with an appropriate terminology)

   (a) Grant/revoke access to a particular subgraph by a particular originators or group of originators
   (b) Grant/revoke access to a particular file by a particular originators or group of originators
   (c) Grant/revoke access to a particular image by a particular originators or group of originators

12. Functions relating to logging (these specific functions will not be provided if logging information is fully integrated with the triplestore, in which case the generic triple editing functions can be used in conjunction with an appropriate terminology)
   (a) Fetch log relating to a particular subgraph for some time period
   (b) Fetch log relating to a particular file for some time period
   (c) Fetch log relating to a particular image for some time period

3.6 User interface

The user interface to the data warehouse will be quite basic, with application-specific interfaces (for example, Obtima and the patient empowerment tool) the preferred method for accessing the data. The structured data store provides the key index to all the data stored in a particular node of the warehouse, so this will be the primary way in to the data. The user interface will call the programmatic interface at the back end. The user interface will be web-based, and use HTTP Secure (HTTPS) to ensure data in transit is not readable.

3.6.1 Login

The front page will provide two user identification mechanisms, basic username and password login, and an option to make use of previously authenticated identity using whatever such services are developed by Custodix. These identify the "originator" of calls to the programmatic interface described above, herein described as the user.

3.6.2 Front page

The front page will contain the following:

1. A single free-text search box
2. A link to the advanced search page
3. A small set of important categories for faceted browsing, for example:
   (a) Data added by the user

   (b) Disease types
   (c) Research institutions
   (d) Others, to be decided

4. A link to a more complete faceted browsing facility
5. A list of the latest changes to the structured data store to which the user has access, highlighting data added by that user
6. A list of the latest comments and ratings of data to which the user has access, highlighting data added by that user
7. A list of news items relating to the running of the data warehouse
8. A link to help pages
9. A link to contact the administrator
10. A link to Terms and Conditions for using the data warehouse

3.6.3 Triplestore browser

Several front page elements will consist of, or lead to, a presentation of data from the triplestore.

- The free-text search box will search all the subjects, predicates and objects in the triplestore and return matching elements.
- A faceted browser which presents several lists under particular categories determined by commonly-used terms for edges in the graph (for example, data added by a particular user, patients with particular diseases, measurements within particular ranges)

However, a single element in a triplestore is fairly meaningless: the interesting information is in the relationships between elements. Therefore, in presenting such results, care must be taken to select appropriate contextual information.

A typical data browsing interface is accessing a table-based data store, which immediately suggests that a sensible means for viewing such data is a tabular format, and the data to be presented is suggested by the columns in the table of the matching row. Although this is not necessarily the best method for viewing the data, it is a reasonable basic method which many users are familiar with.

The structured data store is based around a triplestore. There are no tables (or rather, there is a single table with three columns), and the data in the store may be fairly abstract. The structure is best described as a directed graph rather than a table. There are no established norms for browsing such a graph, although there are many experimental methods which have not gained

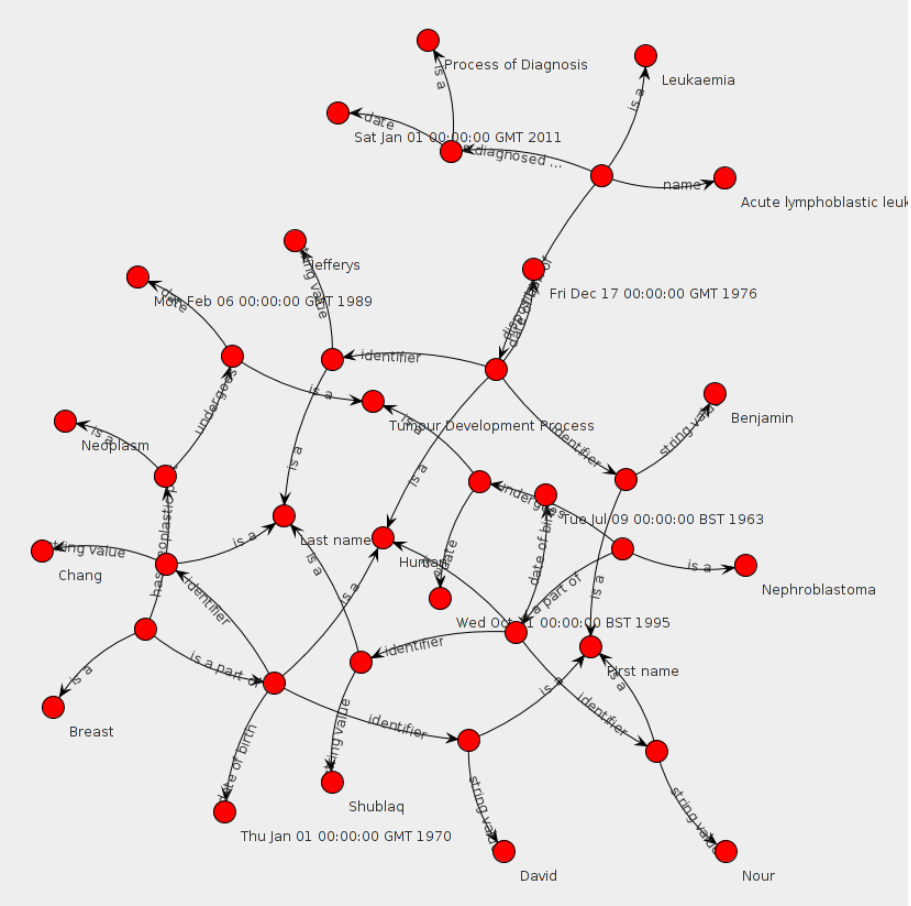

Figure 3: A complete triplestore graph for three individuals with different diseases, including date of birth and date of diagnosis, using the ACGT ontology

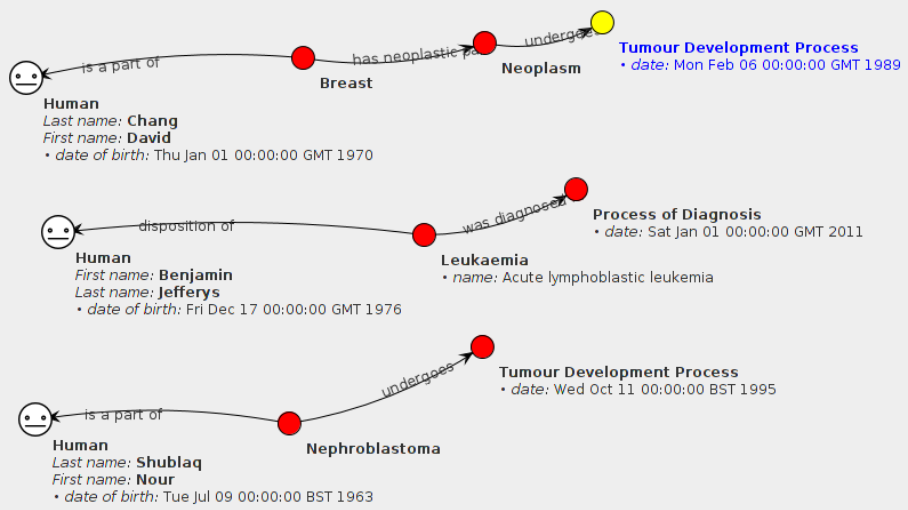

Figure 4: The triplestore graph from Figure 3 with nodes and edges reduced to present information more sensibly, either as text or as icons

widespread user acceptance. Here onwards, the terminology of graphs (nodes and edges) will be used in preference to the terminology of RDF triples (subject, predicate and object)

It is important not to overburden the user with details of the internal data model, but it also important to represent the data that is present accurately. Methods for summarising and reducing data to something more comprehensible must be developed. It is envisaged that several complementary approaches will be necessary, catering for different levels of user. However, none are designed to cater for patients, clinicians or other non-technical users. Such user access will be provided by dedicated domain-specific tools. Minimally, two presentations of triples will be available:

1. A very low level browser which exposes all edges and nodes radiating within some degree of separation from a given node, with links allowing refocusing of the view on other nodes (see Figure 3)
   (a) An option will be provided to collaps common edges (such as the class of a particular node and constant values associated with attributes of a node) to give a clearer presentation (see Figure 4)
2. A more advanced presentation which uses specific templates (perhaps using Fresnel[7] representation) to present certain important nodes - such as patients, their diseases, treatments, and so on - very clearly

In the first two cases, it is proposed that both a free-form 2D visualisation of the graph (see Figures 3 and 4), and simpler text-based presentation is given. The template-based presentation

[7] http://www.w3.org/2005/04/fresnel-info/

method should be primarily text-based, but may present imagery if it is appropriate for the domain (for example, it may show thumbnails of image files which a particular node refers to).

Note that nodes and edges will generally refer to terms in an ontology by URL. Therefore, the default presentation will be quite ugly. A very simple table or triples stored in the triplestore will translate these terms into human-readable form, either textual or iconic.

Domain-specific viewers should be used to present files indexed and annotated in the triplestore, where available. In particular, images should be converted to a format conforming to web standards for display (images in a DICOM server are typically in a proprietary format). Additionally, a direct web-based user interface to the DICOM server may be provided, depending upon the choice of server - dcm4chee includes such an interface. A browsing interface specifically designed to allow browsing of files in the generic file store may be provided if the project requirements specify this.

3.6.4 Advanced search page

The advanced search page will allow users to enter or construct SPARQL queries. Tools will be provided to construct such queries using standard HTML form components and commonly-used ontology terms. This will intentionally be quite a technical interface, however.

3.6.5 Additional user interface elements

Additional features for the data warehouse interface will be specified in WP2, WP3 and WP8 as part of requirements capture, system design and portal design. In particular, the needs of auditors to access particular views of the data will need to be taken into account.

3.7 Integration with VPH-Share cloud infrastructure

There are several areas of VPH-Share WP2 that provide functionality which is potentially useful for the p-medicine data warehouse, with a heavy focus on using the cloud environment. These include security, data reliability and integrity and storage of and access to large binary objects.

3.7.1 Generic file store

Accessing large binary objects in VPH-Share shares many similar requirements to the file storage component in p-medicine. However, VPH-Share will not develop any large file storage capability, this responsibility being delegated to the developers of individual workflows which will consume or produce such data. According to the design architect of this VPH-Share task, it is possible to allow the LOBCDER data access middleware to access the underlying data storage components developed in p-medicine. Additionally, if a WebDAV interface is provided by

LOBCDER, this could be used to federate file stores which are made available to VPH-Share into the p-medicine data infrastructure. This could simplify access as well as provide a easier mean to integrate the workflows in the two projects. This will fulfil the requirement that relevant p-medicine technologies are packaged for deployment in VPH-Share.

The open source cloud platform OpenStack is being used by the VPH-Share cloud infrastructure. The p-medicine data warehouse file storage component could use the same platform to facilitate reuse of components developed in each project.

The data reliability and integrity task could potentially develop tools that could be helpful in preserving the integrity of p-medicine data.

3.7.2 Security infrastructure

The security component developed in VPH-Share was tailored with a focus on the general architecture of WP2. The p-medicine security infrastructure has yet to be defined, but it may be informed by or integrated with that developed in p-medicine. There are also areas where the cloud environment will be used that the security component will be useful.

3.7.3 Cloud hosting

VPH-Share WP2 also provides a cloud execution and hosting environment which is accessible through REST web services. The primary intention of this is to deploy applications (for example, a simulation program) on a cloud platform in a transparent manner. Although this is not relevant to the p-medicine data warehouse, this mechanism could be used to host web service enabled research tools from other p-medicine work packages.

4 Summary

This document has set out a list of preliminary requirements, in the absence of fixed formal requirements from other work packages. It is likely that these will be extended over time, so we must remain reactive to changing circumstances. We have also summarised a related work package in VPH-Share, with which we aim to cooperate and share solutions and ideas. Finally, we gave an overview of the components which will make up a data warehouse, with ideas as to how they will be loosely coupled via standard protocols. Figure 2 gives an overview of the components.

A **generic filestore**, accessed through multiple protocols, will store files in a distributed and scalable manner, with redundancy to ensure maximum availability. A filestore will have an associated structured data store. Access to the filestore will be governed by the Custodix security infrastructure and warehouse-specific access permissions stored in a separate system, most probably the associated structured data store. Other information will be stored on the associated

structured data store, including access logs, data extracted from the files in the store, and information that might normally be found in a standard filestore, such as relationships between files (perhaps hierarchical) and timestamps. Software developed by PSNC is likely to be at the core of the file store for p-medicine, but loose coupling and use of standard protocols will allow us to make use of other file storage infrastructures.

A server specifically designed to **store medical images** and associated metadata will be provided, accessed via the DICOM standard protocol. Image metadata and access logs will be kept in sync with data in the associated structured data store. This server is provided to allow direct integration between the data warehouse and hospital systems. A web-based user interface will be provided as part of the server as a domain-specific interface to this important data source. These requirements are currently bes fulfilled by the freely available *dcm2chee* software, however future work may lead us to a different solution.

A primary function of a data warehouse it to bring together and integrate data for further analysis, to inform future courses of action. A **structured data store** will enable this process. The diversity of data sources, the requirement that ontologies be at the core of our integration effort, and the expectations of data mining and modelling groups, have led us to conclude that structured data should be kept in an RDF triplestore. The means we benefit from existing infrastructure and software which supports RDF storage and querying. Currently the *OWLIM* software from *ontotext*[8] appears to be the best choice for the triplestore, although future work may identify superior alternatives.

Data associated with files and images submitted to the stores must be **semantically integrated** (or, in data warehouse terminology, conformed) before being added to the core structured data store. This enables data mining and modelling to ignore many subtle aspects of data management, and focus upon analysis, a key task for a data warehouse.

Integration will primarily be performed by services external to the data warehouse, developed by UPM. We will support this process by performing a **preliminary data extraction** task which extracts data from some standard file formats (for example, Microsoft Access and Excel, CSV, microarray formats) and adds them to a dedicated part of the structured data store, ready for UPM's integration process. A basic schema for such data will either be extracted from the file, or requested from the submitting user.

Maintaining the data warehouse as a secure source of high-quality, integrated data on cancer treatment is absolutely core to its role in p-medicine. This cannot be achieved by computational means alone. The structured data store will also keep provenance information, user annotations, historical data, and logging information, to allow **human curation and auditing**. This information will be integrated with the rest of the structured data, with appropriate ontologies, allowing it to be analysed just like the core medical information. Web-based user interfaces to this information will be provided.

Multiple federated data warehouses will support work under p-medicine. These will be loosely coupled to allow delegated querying of a subset of these warehouses via a particular warehouse.

[8] http://www.ontotext.com

Each warehouse may maintain a local copy of remotely-managed structured data, to speed up queries.

Any of the components of the data warehouse might be deployed on **cloud infrastructure**, and the services and interfaces provided must support this. We will focus in the first instance on services developed in VPH-Share, but will also make use of commercial providers where appropriate, assuming appropriate security mechanisms are supported.

As well as the direct, standard interfaces to the generic file store and image store, we will provide a **RESTful web services interface** to the structured data store. This will allow SPARQL querying and direct input of structured data, as well as provide an interface to data access permissions and logging information which will be accessed by the file and image stores.

Finally, on top of all of this infrastructure, a **web-based user interface** will be provided for browsing and querying the data in the warehouse. The primary users of this will be technical, for example, data miners and modellers, developers of the p-medicine infrastructure, and auditing and data management people. Therefore, the interface will be quite technical (with a few embellishments to make certain tasks easier). Domain-specific interfaces, such as Obtima and the patient empowerment tool, will provide more user-friendly data warehouse access.

References