Electronic Health Records for Clinical Research

Executive Summary for deliverable D3.1: Initial EHR4CR architecture and interoperability framework specifications

Project acronym: EHR4CR
Project full title: Electronic Health Records for Clinical Research
Grant agreement no.: 115189
Budget: 16 million EURO
Start: 01.03.2011 - End: 28.02.2015
Website: www.ehr4cr.eu

Coordinator: AstraZeneca

Managing Entity: EUROREC

Document description

Deliverable no:	D3.1 (executive summary)		
Deliverable title:	Initial EHR4CR architecture and interoperability framework specifications		
Description:	First deliverable of Work Package 3		
Status:	Final		
Version:	0.2	Date:	14/08/2012
Deadline:			
Editors:	David Voets		

Document history

Date	Revision	Author(s)	Changes
03/08/2012	0.1	David Voets	Initial draft
14/08/2012	0.2	David Voets	Final version

Table of Content

1. Introduction

Deliverable D3.1 - *Initial EHR4CR architecture and interoperability framework specifications* is a snapshot of the *architecture description* (AD) for the EHR4CR platform taken after the first year of the project. The EHR4CR AD is a continuously evolving document reflecting the underlying architecture of the EHR4CR platform. The following annual releases of the WP3 (Architecture and Integration) deliverables (D3.2, D3.3 and D3.4) are going to reflect the state of the EHR4CR architecture continuously after each subsequent project year, hereby documenting in particular how the architecture supports each of the four main clinical scenarios (*Protocol Feasibility Scenario* (PFS), *Patient Recruitment Scenario* (PRS), *Clinical Trial Execution Scenario* (CTES) and *Adverse Event Reporting Scenario* (AERS)) .

The purpose of the *AD* is to provide a blue-print for allowing design and implementation of a set of specified components and services and in order to integrate and operate them as a platform. The AD is described based on the perspectives of the different stakeholders of the EHR4CR platform and is structured according to different views. Each view addresses a number of concerns as expressed by the different stakeholders and explains the rationale behind the architecture decisions that have been taken in order to address these concerns. *Correspondence rules* ensure alignment of the different views and consistency between them by describing the relationships between different architecture elements described in these views.

Although the current iteration of this document focuses on the PFS, the general characteristics of the EHR4CR platform are described in a high-level overview relevant for all four clinical scenarios. Where applicable, additional views describe how services in support of the PFS may be reused and extended in order to support the future scenarios.

At the initial stage of the project, the *AD* is limited to the following views: overall view, functional view, information view, data protection view and standards view. In this executive summary, we elaborate on the overall view and provide a summary of the other views in order to illustrate the approach for documenting the architecture. In the second year of the project - following a first demonstration of the technical platform - a development and integration view, deployment view and operational view will be added and all existing views will be incrementally extended.

This AD aims conformity to related requirements specified by ISO/IEC/IEEE Std 1471:2011.

2. Architecture description background

The EHR4CR architecture description (AD) is based on the 'concept of view' as specified in *ISO/IEC/IEEE 42010:2011*. A view is a representation of a system from the perspective of a set of related concerns (expressed by the stakeholders). The set of conventions on how to construct, interpret and use a view is called a viewpoint. A viewpoint specifies the models to be used for describing the concepts that are relevant to that view (e.g. UML static structure diagram used in the information model view). Some views may cover concerns that affect many of the other views (called crosscutting concerns). A typical example is a security view, which is likely to interact with many other views (e.g. functional, operational, development...). It is therefore important that consistency is maintained between the different models and views by defining correspondence rules. Correspondence rules capture the relationships between architecture description elements used in different views and make the relationship between these elements explicit and thus manageable.

3. Overall view on the EHR4CR platform

3.1. General principles

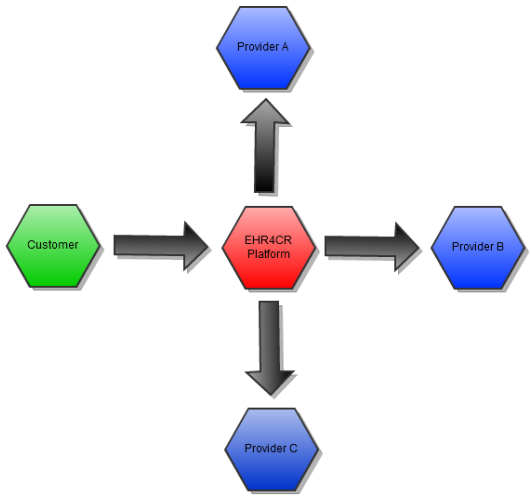

Figure 1 - General architecture approach

The EHR4CR platform provides a set of application and data access services for addressing the four main EHR4CR scenarios as well as most of the necessary infrastructure services (e.g. authentication and authorization, service registry ...) needed to support them.

The EHR4CR platform is based on a Service Oriented Architecture (SOA) in which service providers and consumers can dynamically connect. As such, the primary goal of the EHR4R AD is the specification of clearly defined interfaces and responsibilities while the physical location of service consumers and providers is only of secondary importance[1].

Each of the data provider systems connected to the platform provides a set of services to address one or more of the main EHR4CR scenarios. A data provider may not be able or willing to provide all specified EHR4CR services. Therefore, the EHR4CR compliance of a data provider's services is specified in terms of addressing a certain profile (a profile is a set of services). It is important to remark that the set of platform services exposed to the customers of the platform (in the form of an API or a [web] application) does not necessarily map *one-to-one* to the services provided by the data provider nodes. This allows for a different granularity of reuse (e.g. addressing aspects of protocol feasibility and patient recruitment with the same query interfaces) and allows adding capabilities to the platform's application services while avoiding frequent updates (extensions) to the data access services.

Given the distributed nature of the architecture, service and information model versioning is a major concern. This is based by the fact that service providers will not be able to upgrade to a new service nor a new information model version at the same time (next to operational difficulties, this would render the entire platform unavailable for a certain amount of time) and the fact that not all providers may support the same set of services and level of data integration. A central registry allows querying metadata on the different endpoints, including supported profile and version information. Most issues related to the distributed nature and the heterogeneity of the supported services and data models will be handled centrally by the platform. However, for many aspects the end-user will need to be aware about this heterogeneity (in order to correctly interpret query results and patient recruitment outcomes). This concern is addressed by the audit logging and provenance services provided by the platform.

3.2. Data endpoints

Data endpoints are key services in the EHR4CR platform from which the different scenarios can be built up. Given their importance and complexity, they are a particular focus point of the EHR4CR AD.

[1] Although the physical location of data should be an important aspect from a legal, ethical and regulatory perspective, it appears to be of less relevance from an architecture perspective.

The following figure provides a simplified view of the EHR4CR platform, focusing on the data endpoints:

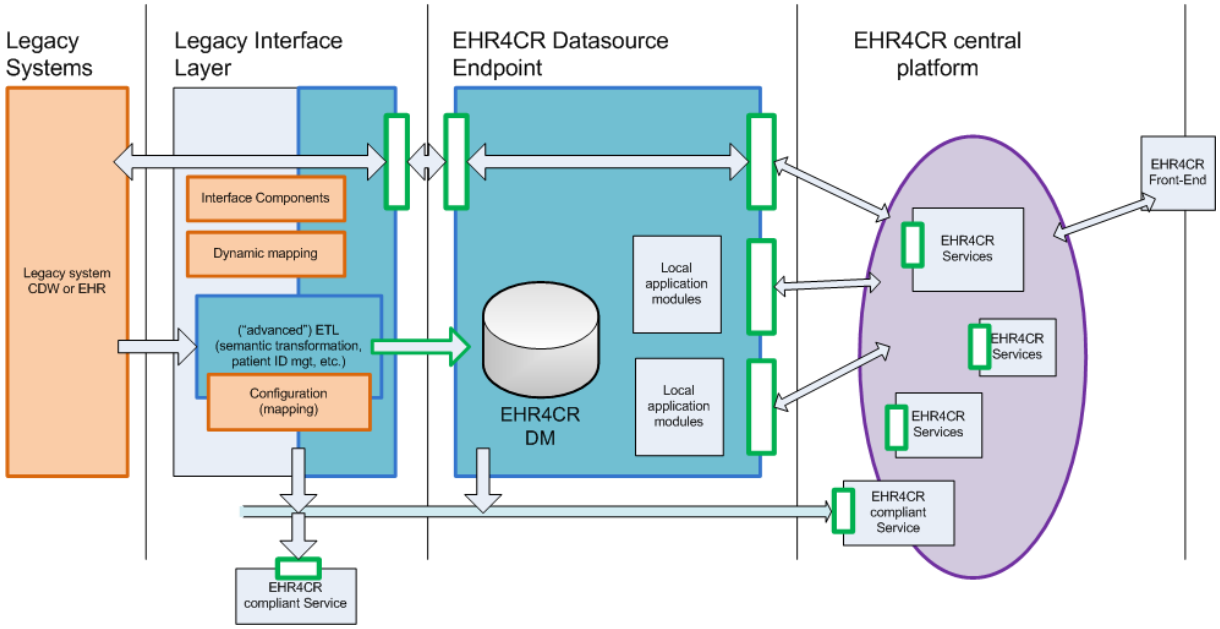

Figure 2 High-level architecture focusing on the data endpoints

The service-oriented character of the platform is reflected by the fact that services can be accessed from any point of the architecture provided that they are EHR4CR compliant (i.e. adhere to the standard interfaces). In order to support the different scenarios, the data endpoints will expose a number of services for the purpose of issuing local queries (e.g. in the protocol feasibility scenario for retrieving patient counts matching inclusion and exclusion criteria), workflow interaction (e.g. flagging eligible patients for clinical trial recruitment) and information exchange.

Figure 2 also illustrates two possible approaches for semantic data integration. The approach at the top illustrates the *mediation* approach where requests issued against the common information model of the platform are dynamically translate to operations against the local information model.

The approach displayed at the bottom of the diagram is based on an *Extract-Transform-Load* (ETL) methodology in order to transform the data model of the legacy system into a representation that is closer aligned with the common information model. This may include local terminology translation, patient record de-identification and data model restructuring. In this case, the data endpoint services will not be operating on the legacy system directly, but will be operating on a *data mart* that is optimized to support the EHR4CR information model and scenarios.

The blank rectangle boxes represent the standard EHR4CR interfaces at the different levels of the architecture.

3.3. Semantic interoperability

EHR4CR has opted (at least initially) to export hospital information to *hospital internal EHR4CR compliant Clinical Data Warehouses*[2] (CDW). An EHR4CR compliant CDW is structured (or at least query-able) according to the EHR4CR information model and EHR4CR terminologies.

- The structure and the content of these data warehouses are defined in WP4 and WP7. The physical model of CDW is defined by WP4 (and WP7) only for hospitals that do not already have an in-house CDW. In addition WP4 has defined a *global-as-view* schema.
- The initial approach feeds the EHR4CR data warehouses through an ETL process which takes care of the semantic transformation of a local information model into the agreed EHR4CR common information model. Alternatives to the ETL approach, e.g. a (dynamic) mediated approach are also considered. These different approaches do not affect the global architecture of the platform. For practical reasons it is however important is to decide where and when to transform the original data (if required). Centrally-managed mapping is more complicated due to governance and maintenance issues. Further, centralised mapping does not facilitate service composition because the end-points may not expose a uniform information model and thus service re-use and composition requires complex transformation workflows. We therefore consider conversion to the *EHR4CR platform model* a responsibility of the data endpoints.

Three different types of data sources were identified, all of which could be query-able through the EHR4CR platform:

- Existing data warehouse (already used by partners) addressed directly by the platform
- (Literal) copy of an EHR data repository
- EHR4CR data warehouse designed/proposed by the project feed through an ETL process and managed locally or in by a group of source data providers.

3.4. Summary of the protocol feasibility scenario

The AD currently focuses on the Protocol Feasibility Scenario (PFS) as this scenario has been elaborated in detail during the first year of the project.

The main services involved in this scenario are:

- Protocol feasibility tools in the form of a workbench for studying non-identifiable distributed patient data (*meso feasibility*[3]). Note that these tools focus on authoring and managing (computable) eligibility criteria queries rather than providing functionality for clinical trial protocol authoring;
- An orchestration service allowing distributed execution of eligibility criteria queries;

[2] The term data warehouse is used to identify any repository of EHR data that can be accessed for secondary use by any authorised person. **The data warehouse is in principle not 'centralised' and remains under authority of the healthcare providers and professionals.**

[3] Three types of feasibility can be considered: 1. Macro Feasibility: Program Level (prevalence of diseases and conditions in regions) 2. Meso Feasibility: Study Level (whether clinical study can be performed in a country or a region) 3. Micro Feasibility: Site or Investigator Level

- Endpoint (data access) services allowing eligibility criteria query execution on local clinical data warehouse facilities.
- Supporting semantic interoperability services (e.g. coding system value mapping), registry services (e.g. for dynamically discovering query endpoints) and security services (e.g. *single sign-on*)

The PFS consists of the following operational phases:

1. Definition of the goal of the protocol feasibility study
2. Definition and management of the "study queries"
3. Launching the query
4. Getting the results
5. Modifying the queries and measuring impact on results
6. Analysis of the results & simulations
7. Defining the final protocol

The conceptual data flows (not showing the physical data locations) involved in the PFS are depicted in the following figure:

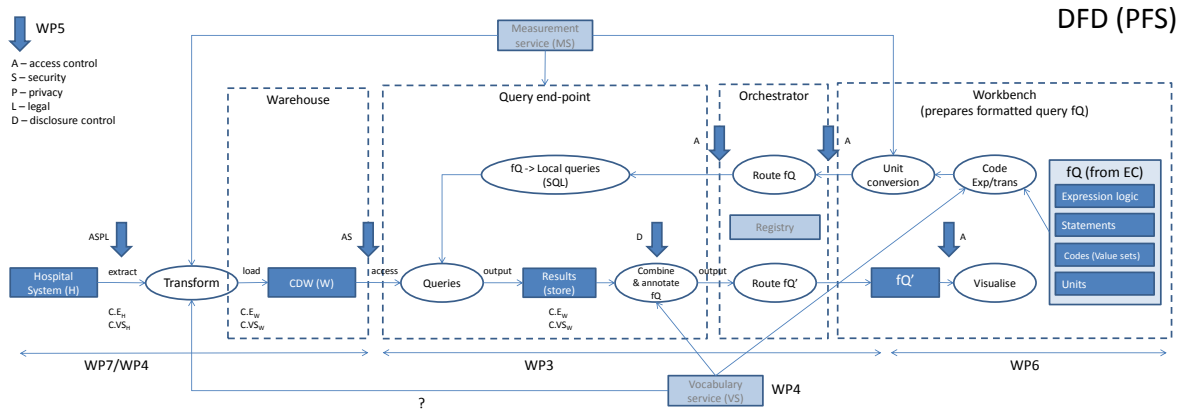

Figure 3 Conceptual Data Flow Diagram with WP boundaries and interaction points

As per the above diagram, the architecture is based on distributed CDWs remaining under the full control of their owners (meaning these will only be accessible through specific query endpoints and that no patient-identifiable information leaves the local hospital networks). The endpoints are query-able according to a common information model which implies that endpoints need to address semantic interoperability issues (due to different data structures and coding systems used, e.g. pilot sites will provide data using national coding systems where no international standards are available). Hence the importance of semantic interoperability services need to be stressed as previously discussed.

4. Summary of the other views addressed by the AD

We provide a summary of each of the other addressed views and include an illustration of the architecture models used in each view.

4.1. Functional view

This view describes the capabilities, structure, responsibilities and specifications of the EHR4CR components and services and explains how they interact with each other.

The functional view categorizes the EHR4CR services into three types: *application*, *semantic interoperability* and *platform* services.

The term *applications* encompasses all services that directly support one of the four main EHR4CR scenarios. *Semantic interoperability services* covers all services that can be used by any entity on the platform to facilitate semantic interoperability (e.g. can be included in ETL processes that prepare existing data for use in the platform) or to meet direct data interoperability requirements (e.g. concept expansion during semantic query evaluation). Finally, *platform services* covers all logic (including functional logic and infrastructure) that is common to multiple scenarios.

The following figure shows the main services and components involved in the PFS, focusing on the core functional aspects (e.g. security services not depicted):

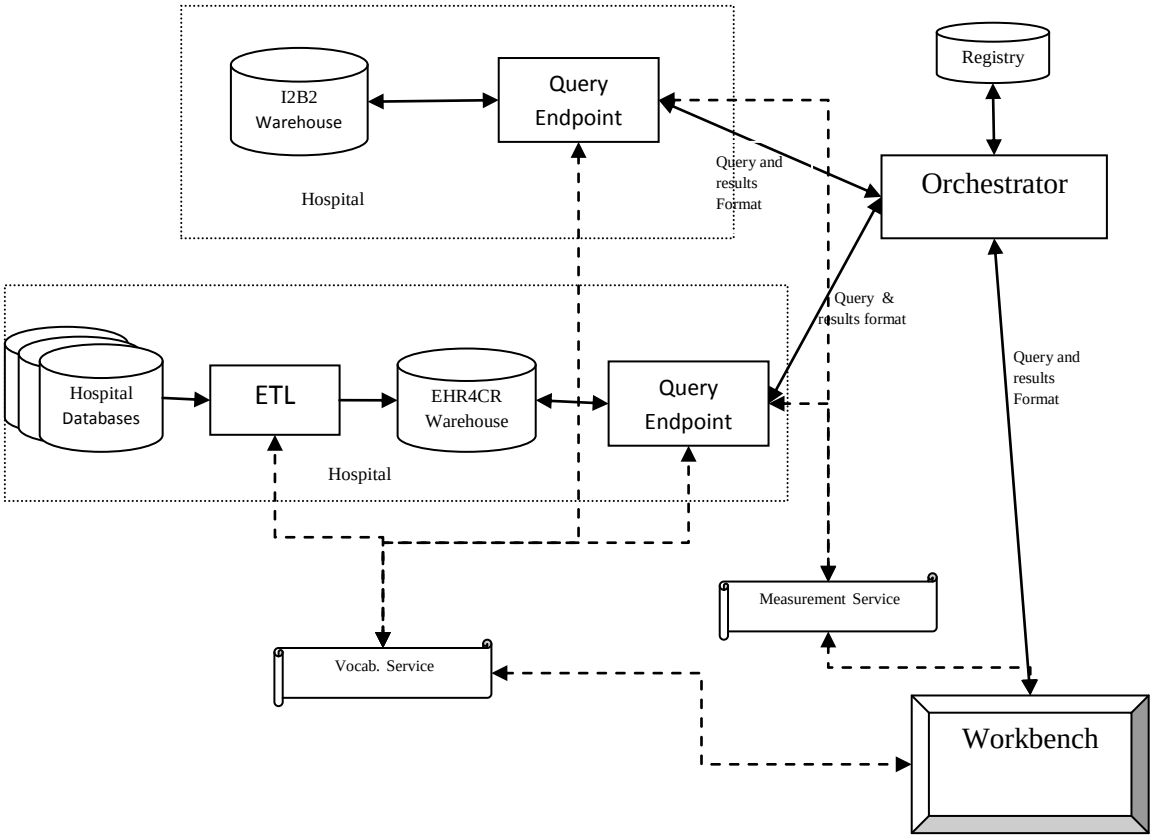

Figure 4 PFS Services and components interaction

4.2. Information view

An information view typically addresses three levels of information models:

- The conceptual model: describes the information that is relevant to the scenarios being addressed (*domain knowledge*);

- The logical model: describes how information flows through and is used in the different components and services;
- The physical model: describes how information is persisted (e.g. inside a relational database).

In the EHR4CR AD, we adopt the following strategy for documenting the information models:

- The conceptual models are described in the information view;
- The logical model is expressed by correspondence rules between functional view and information view;
- The physical model is expressed by correspondence rules between information view and deployment view.

In general the conceptual models are represented using UML static structure diagrams as is shown in the following example:

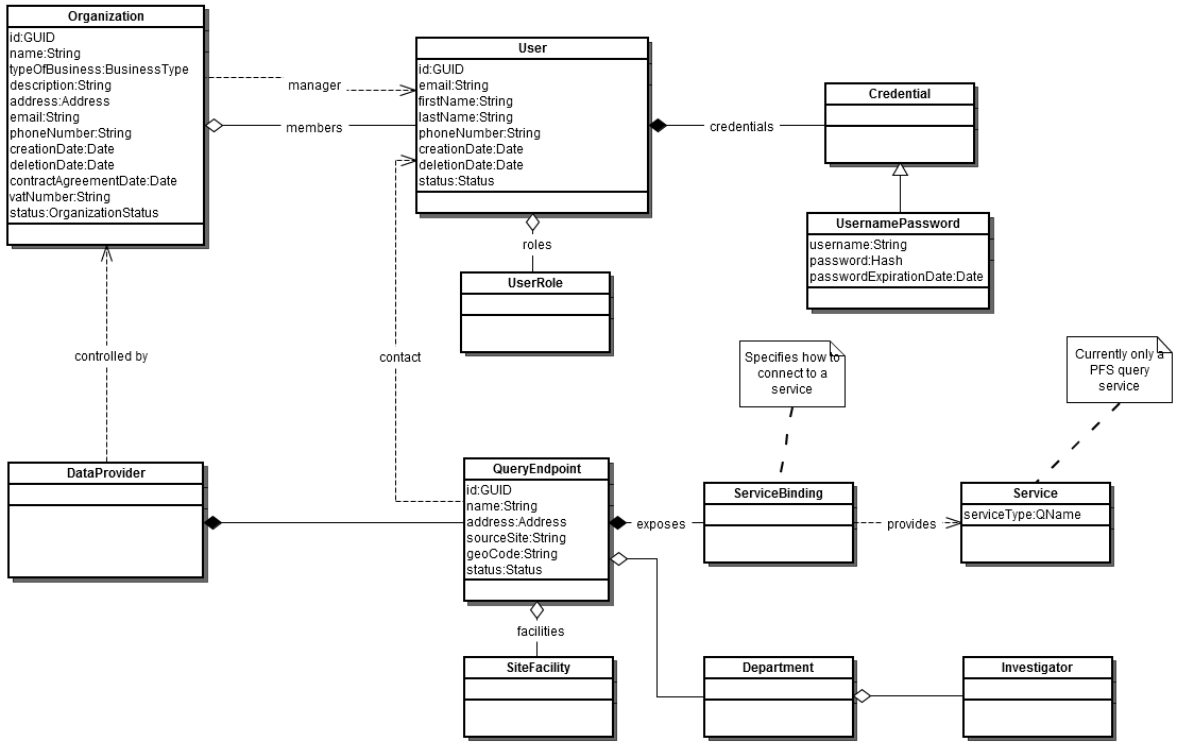

Figure 5 Technical platform governance model

4.3. Data protection view

The data protection view describes how data security constraints are addressed by the EHR4CR platform architecture. This includes an overview of all data protection related (high level) requirements as described in the EHR4CR *Software Requirements Specification* (WP1, Task 1.3) and the translation of these requirements into technical guidelines or measures that must be undertaken in order to fulfill them. The application of these technical measures to the different architecture elements is part of the consistency view in the form of correspondence rules (see 4.5).

Consider for example the following high-level requirement:

Name	Description
Confidentiality	No information must be exchanged over the Internet in unencrypted form.

Figure 5 Data protection concern

This requirement is translated into the following technical guideline:

The EHR4CR is a distributed platform involving many physical sites operated by different organizations. Each of these sites will expose a number of applications and services available to the EHR4CR platform and its users. As these endpoints will be interconnected over the public Internet, provisions are required to ensure confidentiality between communicating entities.

Provisions at the application layer (layer 7 of the OSI model) should be favored above provisions at lower layers since the former ensures true end-to-end confidentiality. Examples include WS-Security encryption for SOAP messages and XML Encryption for REST services exchanging XML messages.

Figure 6 Data protection concern technical guideline

4.4. Standards view

The standards view lists all IT standards (including both general purpose and specific health informatics standards) that have been considered relevant in the context of EHR4CR, irrespective of the fact whether they have actually been selected as part of the solution. For each identified standard, a recorded rationale clarifies the reason for adopting or avoiding standards compliance in the architecture. In the former case correspondence rules describe the relationship (e.g. adopted compliance level) between a particular architecture element and a selected standard (see further).

Name	*Category*	*Description*	*Relevance to EHR4CR*
OASIS UDDI v3	Platform management: Service discovery	Universal Description Discovery and Integration. A mechanism to register and discover web services (in order to promote loose coupling) , based on the metadata that is registered with it to describe the capabilities of these services and/or the organizations publishing them. http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm#uddiv3	Service discovery capabilities are an essential feature of a service oriented architecture. As such, standard specifications exist that provide such capabilities, most notably the UDDI v3 and ebXML RIM specifications. Although both the UDDI and ebXML specifications provide support for managing metadata as part of the endpoint reference information, such information could also be managed by a separate entity. It is to be determined whether the capabilities offered by these specifications will be sufficient for use within the EHR4CR PFS context. If this is not the case, a non-standard solution (e.g. Mule Galaxy, WSO2 Registry) could be used or a custom solution will need to be designed and implemented. The choice for service registry technology also depends on the type of services to be registered, e.g. UDDI was intended for publishing SOAP web services.

Figure 7 Example of an identified relevant standard

The standards view also encompasses a *standards forecast*. This is a list of upcoming relevant standards that may be considered for adoption or to which contributions from within the EHR4CR community could be considered.

4.5. Consistency

As the different views capture architecture elements from a particular viewpoint, overlap unavoidably occurs. Correspondence rules capture the correspondence between related architecture elements in the different views and describe their relationship. In this way, inconsistency problems can be avoided and architecture aspects that would have otherwise not been captured (based on the viewpoints alone) can be addressed. As such, this *special view* allows weaving the different views in order to form a complete and coherent understanding of the architecture.

The AD specifies relationships between views in the form of correspondence rule tables. The following table illustrates the correspondence relationships between functional and data protection view elements:

Functional view element	Data protection view element	Correspondence
All services and their operations as specified under section 10.4.2.5	12.2 Confidentiality	All services must require either transport-layer (end-to-end) or message-level encryption (preferred). Unencrypted connections must be rejected.
All services and their operations as specified under section 10.4.2.5	12.1 Authentication	All services must enforce authentication and authorization.

Figure 8 Example correspondence rules between functional and data protection view elements

5. Conclusion

The EHR4CR Architecture Description (AD) serves many purposes. First and foremost, it ensures a common understanding amongst the EHR4CR stakeholders. Next, it ensures that important concerns have been addressed by the architecture and documents how (in the form of a documented rationale (?), architecture models and consistency rules). Finally, it provides a blue-print to platform implementation entities (requirements, constraints) and guidance to platform adopters (interoperability profiles, operations and procedures).