A MEDICAL INTEROPERABILITY ARCHITECTURE

Bruno Alves, Michael Schumacher
University of Applied Sciences Western Switzerland (HES-SO)
{bruno.alves, michael.schumacher}@hevs.ch

ABSTRACT

Interoperability has the potential to improve care processes and decrease costs of the healthcare system. The advent of enterprise ICT solutions to replace costly and error-prone paper-based records did not fully convince practitioners, and many still prefer traditional methods for their simplicity and relative security. In this context, the MediCoordination research project proposes a Service-Oriented Architecture, combining a totally decentralized storage for patient records and a federated metadata infrastructure allowing semantic descriptions of medical documents. While taking a complementary approach to Integrating the Healthcare Enterprise (IHE) IT Profiles, among which IHE XDS, we were able to design an enterprise-level architecture based on the recommendations of the recent Swiss eHealth strategy on architecture components and standards. The Medicoordination Healthcare Infrastructure presented in this paper provides enough scalability to fit the fragmented nature of the Swiss healthcare industry. The component-based nature of its architecture enables a good separation of roles between storage and resource description, while enabling reusability in other projects. A prototype is implemented and deployed; and while partially incomplete, it already provides encouraging results in terms of security, scalability and efficiency. Experimental results highlight document storage and retrieval times in the range of milliseconds.

KEYWORDS

Interoperability; Security; eHealth; Healthcare; Component-based system development; Service Oriented Architecture; Information system architecture.

1. INTRODUCTION

Interoperability in data exchange has the potential to improve the care processes and decrease costs of the healthcare system. To tackle the high potential of the domain of medical interoperability but also respond to potential risks of data abuse, strategies for the interoperability exist in many countries [1, 2] and also on a European level [3].

In 2006, the *Swiss Federal Council* began elaborating a new strategy concerning the usage of Information and Communication Technologies (ICT) in the context of the Swiss e-Health [4]. It also highlighted the importance of the information technologies in cyber-administration (e-Gov). The new strategy intends to take Swiss e-Health towards an improvement of efficiency, quality and security while also improving the productivity.

The new e-Health strategy is intended to guarantee Swiss people the access to a health system, which is efficient, secure and cost-effective. The strategy must: influence favorably the costs; improve the competence of the population, which is now responsible for its medical data; reinforce the quality and security of the care services with a between knowledge management.

The new strategy tries to normalize the processes and standardize them. Cost reduction and efficiency improvement are possible by using electronic infrastructures, which also reduce errors and costly administrative tasks. The end goal of the strategy is to place the patient in the center of the healthcare system, i.e give the patient the full control of his data.

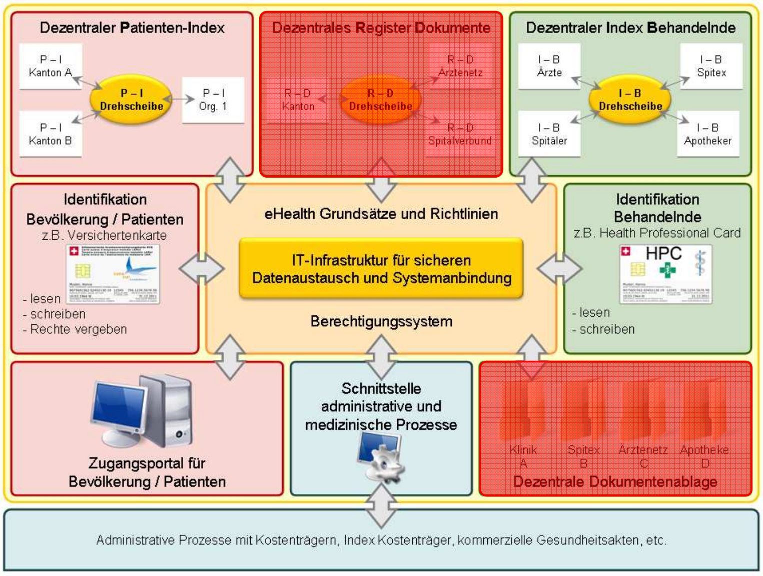


Figure 1. Swiss e-Health strategy

The modules composing the new Swiss e-Health strategy, illustrated in Figure 1 are about patient management (empower them with rights to their own data), decentralized document storage, registry, insurance cards, identification and administration. This paper presents the architecture developed for the modules in red: **decentralized registry** and **decentralized document storage**.

Medicoordination is a research project taking a complementary approach to the IHE Profile IT specifications [5]. It describes an enterprise-level Service-Oriented Architecture (SOA), which can be shared by different medical actors running different IT appliances, in order to make them interoperate. The objective of this project is to design a totally decentralized storage for Patient Electronic Health Records (PEHR). The Medicoordination project provides a federated metadata infrastructure allowing semantic descriptions of medical documents and a document-centric storage with versioning capabilities.

This paper provides an overview into an architecture for a federated metadata and storage Web platform, called Medicoordination Healthcare Infrastructure (MHI). It primarily aims at making possible the cooperation and interoperation of heterogeneous information systems in simple exchange scenarios. Its design is application agnostic, however, the end goal of this description is geared towards medical integration. This document describes the base concepts driving the architecture decisions as well as limitations and descriptions of the underlying models

A template is a set of styles and page layout settings that determine the appearance of a document. This template matches the printer settings that will be used in the proceeding and the CD-Rom. Use of the template is mandatory.

Clearly explain the nature of the problem, previous work, purpose, and contribution of the paper.

2. METHODS

The first MHI Architecture Specification (MHIAS), shown in this paper, describes a distributed and component-based system for managing patient records using semantic descriptions of medical documents. Documents belonging to an electronic health record (EHR) are henceforth called **fragments**. In this paper, we will first outline the global objectives of the infrastructure, show the models behind the components and provide hints for their implementation. This architecture focuses on modularity, interoperability and re-usability. It does not aim to provide a proper implementation, despite the fact that some modules have been implemented in this context, but a starting point for a concrete system.

A list of primary constraints was first elaborated as a bootstrapping process to give a clear direction project. First, in Switzerland, were data and privacy protection has always been considered as essentials, institutions and patients are empowered the responsibility of their own data (*distribution*) ; they might be able to access other repositories containing part of their data (*compatibility*) ; since documents may not be in a format accepted by all systems, their transformation into other formats must be feasible, when possible (*interoperability*) ; information should not be accessed by unauthorized parties (*authentication*) ; information should not be accessible to all care professionals, except in case of an emergency (*authorization & protection*) and finally, all systems should be independent of the underlying hardware (*independence*).

Our approach consisted in a preliminary survey on existing standards and storage formats. It was necessary to acknowledge potential fragment formats in order to tackle the requirements for the metadata. The existence of structured formats such as CEN EN13606 [6], also known as EHRcom and HL7 CDA [7] would be a good starting point for extracting and processing information about the documents and complete metadata descriptions. For simplicity and time-constraints, the first specification focused only on fragment properties, instead of its contents. Metadata within this architecture specification describes files properties like dates, issuer, intended recipients and summarizing information, as fragment types and document format. No clear semantization is achieved and no inferencing patterns are used.

During requirements analysis phase, several ideas were discussed about whether to make the platform distributed or centralized. Switzerland is a fragment country composed of regions (also called *cantons*). The same goes for its healthcare system. It is also fragmented. Decisions are mostly taken at the regional level (cantonal authority), although some coordination structures exist (like eHealthSuisse for instance) at the federal (country) level. In this context, a federative distributed architecture was foreseen. The advantage of such design was that data would preserve its locality (data is kept where it is created). For storage of the documents, a free form solution was selected. The storage was designed to be accesses as a web resource (addressed by an URL), with its credentials and security managed by the Medicoordination platform.

Subsystems were modeled and designed as independent components, but attention has been paid to their interconnections with other systems. Five models, about metadata, storage, coordination, security and identity services have been specified and will be later discussed in this paper. Each model highlighted specific challenges and requirements. They also provided some guidelines and starting points for a concrete system designs.

Models were at the base of a concrete specification. We designed a SOA architecture composed of services layers (subsystems), which were specified from their respective models. The result was an hierarchy or services coordinated by a single authority. Subsystems are, from the bottom-up: **metadata** and **storage** service layers, known as Metadata Service Layer (MSL) and Storage Service Layer (StoSL), providing registry and repository services; **identity services**, giving support for identities, **roles and authentication (security)**; and the **coordination** layer, also known as the Medicoordination Service Layer (MediSL), providing to glue to tie systems together. MediSL is a thin web client management platform, which has the role of federating the composing systems and providing a unified vision. Figure 1 summarizes the the different services.

Concrete implementations were done on some parts of the architecture. A distributed metadata registry based on a semantic federated RDF stores was implemented and deployed. The storage service layer was implemented in the context of a bachelor project [8] at the University of Applied Sciences Western Switzerland (HES-SO // Valais). Unfortunately, no concrete implementation was yet been done for Identity Services. When the project originally started, requirements or standards for managing identities were still left to specify by the Swiss Confederation.

3. RESULTS

The following section provides an insight into the different sub-systems of the Medicoordination infrastructure. Each sub-section gives some details on their respective models and specification.

The MHI is a distributed and component-based system for managing patient records using semantic descriptions of composing fragments. Each system exposes its functionality through a well defined web-service interface and is coordinated with the other subsystems by the MediSL. Security is applied globally on all levels and credentials provided for a subsystem are valid for the others.

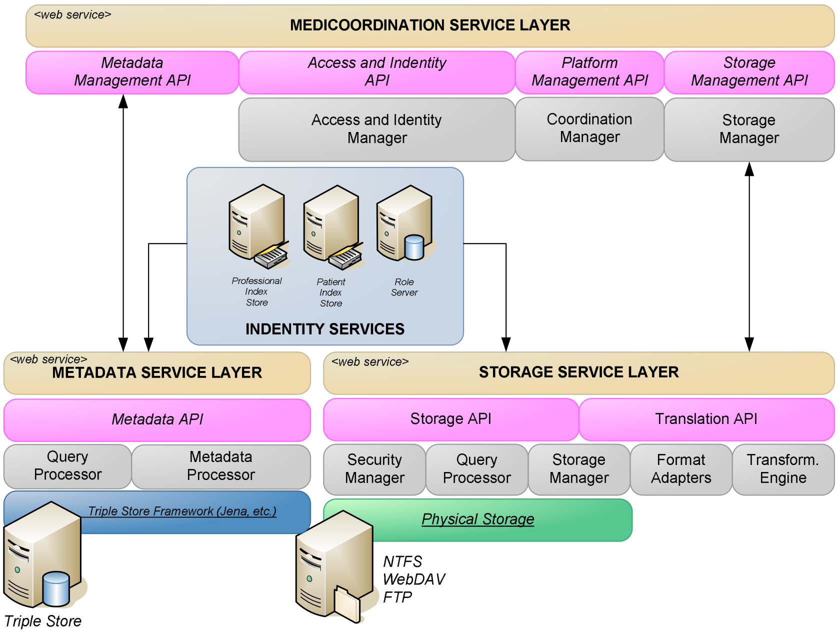


Figure 2. The Medicoordination Healthcare Infrastructure (MHI) Architecture

3.1. Describing documents with semantic metadata

Fragments are not independent documents structured in a directory. They are related or linked to other fragments in order to form an *electronic record (ER)*. Furthermore, they are associated to specific access rights, which are the set of authorizations of a patient or care professional concerning its content. Fragments are associated to both a patient, known as the subject of care and a medical professional, known as the author of the document. Fragment metadata represent attributes annotating the fragments to which they relate. These annotations are stored independently from the documents, in a registry**,** summarized in Figure 1 as the Metadata Service Layer. The minimal requirements for this system are the ability to **query**, **store** and **process** the metadata.

The specification of the *Metadata Service Layer* had to comply with constraints linked to the heterogeneous and complex nature of medical interactions and Swiss political concerns. Centralized metadata storage was not well adapted in this context. First, it would require a central authority for managing the servers, which is not a good idea in a state with a fragmented authority (canton). Furthermore, a centralized system is much more vulnerable to attacks, even though it is simpler to administrate. Finally, with a large number of connections and billions of entries, it would become a bottle neck. It was thus preferable to empower each institution with the management of its own data and link the independent nodes together, in a distributed pattern: a federation of metadata nodes.

The metadata system needed to comply with three main requirements. First, the metadata repository had to be accessible by systems, which belong to the same authentication domain (see section on authentication). Furthermore, each node had to carry the responsibility of managing its local metadata resources. Finally, there are cases were a patient needs to be treated in a region different from its residence region. For example, if a patient from Fribourg goes skiing in Valais and breaks a leg, care will be probably handled by the Hospital of Sion (Valais). Some data associated will be kept in Valais, and the other in Fribourg. It was thus necessary to specify a way to access resources inside another domain, by establishing inter-domain trust relationships.

As previously said, a metadata is just an extended file attribute. Each property can be described as a binary relationship between a subject (the fragment) and a specific value, linked together by a predicate (the attribute). In most cases, relationships in this project limit to: the fragment has a size of 500Kb, the fragment has an author named Peter, and so on. Resource Description Framework (RDF) [9] was a natural choice for their representation.

RDF is a graph model describing relationships between subjects and objects. It is not strictly tied to the XML, but it is often chosen as a natural representational language. RDF can be syntactically and semantically structured by ontologies. Ontologies provide a formalism describing the classes and objects in a data set. They use RDF as an underlying description language. Semantic descriptions and datasets are often stored in a special database, called RDF Store. These databases have support for rule-based inferencing and allow creating new entries from the existing ones. For instance, if X is brother of Y, and Y is brother of Z, then the RDF Store combined to an inference engine is able to deduce that X is also a brother of Z.

There exist a number of popular RDF databases, also named RDF stores. Sesame[1], Mulgara[2] and Jena[3] are open source options. AllegroGraph[4] is a commercial RDF store, which supports several add-in features, like reasoners and support for federated databases. All those RDF stores however do not fulfill all requirements for our metadata system. Especially, none of them implement a distributed (federated) store. With the MSL implementation, we opted out for federated RDF store based on WSDIR [10] for the network construction.

Medicoordination is intended to be area agnostic and is independent of any particular application domain. However, in the context of the first specification, a minimal set of attributes applied to the medical domain was created and listed in Table 1.

Table 1. **Minimal Attribute Set**

Attribute	Type	Multiplicity	Description
fragment_ood	Uri	1	Unique file identifier
patient_id	Identifier	1	Patient’s social security number
professional_id	Identifier	1	Care professional social security num
doc_type	Enumeration	1	Document type (exit letter, etc…)
doc_format	Enumeration	1	Document format (HL7 CDA, etc…)
doc_emission_date	Date	1	Date of the submission
doc_update_date	Date	1	Date of the last modification
role_read_id	Identifier	N	Identifier of the role, which can read
role_write_id	Identifier	N	Identifier of the role, which can write
storage_node	Uri	1	Node where the document is stored

The MSL provides a simple interface for reading, writing, deleting and querying entries. Internally, it is composed of a query processor and a metadata processor.

The query processor specifies all mechanisms and algorithms for searching data inside a federation of nodes. It provides the model for forwarding the queries to other nodes, as well to aggregate the results at each level. A metadata processor specifies how to handle the metadata, how to store it and how to make it faster for retrieval.

3.2. Storing and linking fragments in a medical record

The fragment repository has to assume the role of storing, linking and maintaining a medical history of the patients. The set of document is structured in an electronic record. The interface of the service was specified to be as simple as possible. It had to provide minimum capabilities for submitting and retrieving documents. It also has been specified to provide mechanisms for transforming documents into other formats, whereas applicable. This capability makes it interoperable with other medical ICT systems, which rely on specific and not always standard formats.

The storage system way made loosely coupled to the metadata service. It is able to work independently and is fully reusable. Contrary to the metadata nodes, which are stored locally, the specification does not put any constraint on the location schema of the storage nodes. They just need to be accessible as web resources through an URL and support the same security mechanisms.

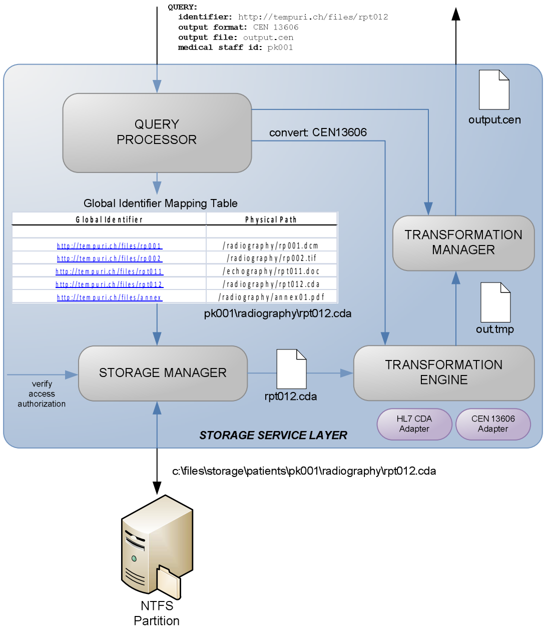
[1] OpenRDF.org, http://www.openrdf.org/about.jsp
[2] Mulgara Semantic Store, http://www.mulgara.org
[3] Jena Semantic Web Framework, http://jena.sourceforget.net
[4] AllegroGraph RDFStore (TM), http://agraph.franz.com/allegrograph/

The architecture specification doesn't make any assumption concerning the physical nature of the underlying storage mechanisms. It is abstract, in the sense that it presents a uniform interface to the users, independently of the storage solution beneath. It is possible to use physical partitions on a hard-drive or link the storage system to a web repository. However, it has to provide support for versioning and name transformation.

Versioning allows the system to track the timely changes in the fragments and thus provide means of reconstituting the patient history. No assumption was made concerning the choice of the versioning software. Name transformation is a mechanism that associates a particular identifier with a physical file name. Fragments are identified by an URI, which is known by the MSL and the StoSL. However, the medium where they are stored may not provide the same naming convention. It is necessary to implement a Global Mapping Table (sort of hash table), which makes to translation between the URI identifying the fragment and its physical name.

The storage system was designed to organize the fragments inside a single patient record. However, no assumption was made on the structure of folder and their naming conventions. It is possible to store fragments inside a complex folder structure or inside a database, as for Apache Jackrabbit[5]. Data pertaining to a patient may be spread among different repositories. It is the responsibility of the MediSL service to find resources (through metadata queries) and present them as if they were stored in the same location.

One of the greatest advantages of the Medicoordination Healthcare Infrastructure is allowing content negotiation and transformation. It is possible to translate between formats when retrieving documents. Figure 3 below illustrates this mechanism. A query process processes requests to the service. Once the file is found and retrieved, is it sent to the transformation pipeline if requested. However, all transformations are not possible. For example, it is possible to transform from a CEN EN13606 document to a HL7 CDA, but not the contrary. The specification does give some hint concerning these transformations. It is possible to use XSL Style Sheets [11] to transform between structured formats based on XML. For images, it is also possible to convert between formats. However, for DICOM images, it is more complicated, because structure information may be embedded within the document.



```
QUERY:
  identifier: http://tempuri.ch/files/rpt012
  output format: CEN 13606
  output file: output.cen
  medical staff id: pk001
```

Global Identifier Mapping Table

Global Identifier	Physical Path
http://tempuri.ch/files/rp001	/radiography/rp001.dcm
http://tempuri.ch/files/rp002	/radiography/rp002.tif
http://tempuri.ch/files/rpt011	/echography/rpt011.doc
http://tempuri.ch/files/rpt012	/radiography/rpt012.cda
http://tempuri.ch/files/annex	/radiography/annex01.pdf

Figure 3. Storage service layer overview

3.3. Authenticating and authorizing professionals and patients

[5] Apache Jackrabbit, http://jackrabbit.apache.org/

The security model of the specification was designed to account for the distributed nature of the project and for the probable technologies it might put in use. Security encompasses communication channel protection, message level encryption and signatures, inter-domain credentials validation, roles, identities, authentication and authorization. All components of the system are secured and imply at least a two-level security.

The Swiss Confederation is very protective in matters that concern information diffusion. Switzerland puts real efforts towards laws, which protect the citizens and their private life [12]. Each patient is given the full responsibility for his own data and are be empowered to accept or deny the diffusion of private medical information of his/her concern to medical staff. *Only* the patient can decide with whom to give fragments of the full record. This reality brings another layer of difficulty, because of the additional management structures it implies.

Security in the MHI essentially consists in providing means to authenticate users in the system, assign them to a role with access rights, manage exchanges between institutions in different domains, and handle security of the medium and authorization.

Authentication in the MHI is token-based. Authenticator servers are responsible for validating credentials and issue security tokens used for future transactions, including validating tokens coming from different domains. The Medicoordination Architecture Specification does not constraint the usage of a particular solution, but the Swiss e-Health strategy considers using SAML Tokens [13]. Trust is ensured by tokens with specific domain rights. Authenticators are also responsible for validating foreign tokens and accept them if coming from trusted domains. For example, a practitioner in Geneva could gain a temporary access to resources in Zürich (case of emergency for example). They are responsible to link principals to roles, and are thus considered as part of the role servers.

Roles in the MHI aggregate identities under a single identity sharing the same rights. The granularity of the roles depends on the context, but they can include either entire organizations or single individuals. Information about patients and professionals is stored in so-called Patient and Professional Index Stores. These stores contain basic information about users and often consist in LDAP[6] endpoints. Both patients and medical professionals are assigned to roles. However, patients may only access to they own data, while professionals might obtain read/write rights in different patient records.

Securing data channels is essential to prevent eavesdropping. Communication channels must be encrypted in order to ensure *confidentiality, protection* and *integrity* over the Internet. Although the MHIAS does not force the use of a particular solution, SSL/TLS may represent a good choice. The problem, however, is that SSL/TLS applies security to the communication channel end-to-end until the session layers. It does not prevent message modification and forging between the application and session layer of the OSI Model[7]. In order to protect the SOAP messages, it is necessary to complement SSL/TLS with message-level protection technologies, such as Web Services Security (WS-Security) [14] and XML signatures [15]. WS-Trust [16] and WS-Federation [17] are particularly useful in the context of federated domains using different authentication methods.

In Switzerland, patients are responsible for the management their medical data. They are empowered the rights to refuse the diffusion of their personal data to a particular medical professional. Authorization consists in ensuring that data is not accessed (read or write) by unauthorized parties. In the current specification, authorization is done using Access Control Lists (ACLs[8]) stored as metadata in the MSL. Doing so was necessary to prevent problems when migrating fragments from a server to another, which would require rights migration.

3.4. Federating systems and presentation

The role Medicoordination Service Layer is to federate the resources and present a mashed-up vision of the results to the user. In this sense, it is composed of a thin web presentation layer and several coordination modules for communicating with the role server, repositories and registries. It is responsible for presenting the uniform vision of the electronic record, whether fragments are stored in different servers or not. The coordination modules are responsible for the coordination of the metadata and the storage services.

[6] Lightweight Directory Access Protocol

[7] Open System Interconnection Reference Model, http://en.wikipedia.org/wiki/OSI_model

[8] http://blogs.msdn.com/brian_dewey/archive/2004/01/20/60902.aspx

The long-term objective we started to tackle with the first specification, was the design of a full management platform for electronic health records with an advanced semantic search engine. However, in this version of the design specification, the search engine was only designed to query fragments matching a certain criteria. However, the detail in the metadata allows for relatively advanced searches.

3.5. The MSL implementation

An implementation of the Metadata Service Layer exists and has already been published. It focuses on federated RDF store based on WSDIR[18] and Jena[9].

3.6. The StoSL implementation

An implementation of the Storage Service Layer was made in the context of a bachelor student project in 2009. It is based on Apache Jackrabbit. It is compliant with the MHI specifications about data protection, accessibility and data versioning. The coordination of the MSL and the StoSL remains yet to be made.

4. DISCUSSION

The Medicoordination Health Infrastructure is intended to bridge to gap between heterogeneous institutions. We wanted to design a system which is simple, interoperable and not tied to a particular technology. The Swiss Confederation recommendations severely constrained our choices, because of the security level and data-loss prevention constraints it imposed.

Our architecture is important, because it can be used in situations when classic interoperability solutions do not work. Even if we did not go into deeps implementation details, the Medicoordination Healthcare Infrastructure may represent a solid starting point to an enterprise-level implementation of a semantic electronic record. The independence of its components makes it scalable. The security components described in this paper make the architecture compliant with requirements of the Swiss Confederation [19] and with the security and protection standards [20] of the medical industry.

In the next specifications, we are planning to improve the existing specification base and propose a semantic search engine providing concise information about patient condition from specific questions asked by the practitioner. For example, we should be able to obtain information about all allergies of a patient by querying fragments, extracting and parsing the necessary information, and to present a mashed up version readable by a human. The goal is to radically improve the time necessary to obtain important pieces of information, without having to browse a vast amount of related files. This is extremely important in the context of healthcare and can save lifes in cases of emergency.

Our solution is similar to IHE XDS [5 p.67-97] repositories, however its specification is more patient-centric. We do not just want to propose a registry/repository, but we also want to give control to patients over their data. -he MHIAS is more abstract and does not prevent the use of IHE profiles for the implementation of its subsystems.

In relation to the TripCom project [22], it could be interesting to adapt the direction of the Medicoordination project to integrate the TripCom communication technologies and paradigms. Could TripCom ever be the web for computers, as its authors claim, it would be an important addition to the project that would allow an even greater independence and decoupling of the modules.

5. CONCLUSION

This document introduced an architecture to be used in situations where the heterogeneity of systems prevents classic interoperability solutions to work. We did not dig into low-level concepts to remain

[9] Jena Semantic Framework, http://jena.sourceforge.net

independent of any architecture. The implementation of systems based on Medicoordination requires careful thought on how to get different parts working together.

Medicoordination, as a research project is intended to give some guidelines about a possible architecture for electronic healthcare infrastructure cooperation, which empowers each healthcare actor to manage its own data, while providing a flexible platform, which adapts to existing standards and infrastructures, with a strong focus on security.

REFERENCES

1. Ruotsalainen P, Iivari AK, Doupi P, Finland’s strategy implementation of citizens’ access to health information, Std Health Technol Inform. 137 (2008) 379-85.
2. Austrian E-Health-Initiative. Austrian E-Health Strategy (in German). [homepage on the Internet] 2005, (Last access 2010-01-31).
3. Commission of the European Communities: e-Health – making healthcare better for European citizens: An action plan for a European e-ealth, Com (2004) 356.
4. Conférération Suisse: Stratégie nationale en matière de cybersanté (eHealth), December 2008.
5. IT Infrastructure Technical Framework, VOL. 1 (ITI TF-1): Integration Profiles, IHE International, August 2009.
6. Health Informatics, Electronic health record communication, Part 1: Reference Model, prEN 13606-1, CEN/TC 251, June 2006.
7. HL7 Version 3 Standard: Clinical Document Architecture, Release 1, 24th October 2000.
8. Alves S., Bachelor project : Storage Manager Implementation, HES-SO // Valais, 2009.
9. W3C. Resource Description Framework (RDF). *Semantic Web Activity.* [Online] 2004. http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/deploy/dgbe_sec_ztsn.mspx?mfr=true (Last access 2010-01-31).
10. M. Schumacher, T. Van Pelt, I. Constantinescu, de A. de Oliveira e Sousa, B. Faltings, *WSDir: a Federated Directory System of Semantic Web Services*, Proc. of 16th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2007), Paris, France, 18-20.06.2007.
11. W3C : XSL Transformation (XSLT), Version 1.0, W3C Recommendation, 16th November 1999, http://www.w3.org/TR/xslt.
12. Confédération Suisse. Constitution Fédérale de la Confédération Suisse. Avril 1999. RS 101 art. 13 alinea 1.
13. Mark O’Neil, Web Services Security, McGraw-Hill Osborne Media, January 2003
14. OASIS Web Services Security (WSS) TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss.
15. IETF/W3C XML-DSig Working Group: XML Signature, http://www.w3.org/Signature/
16. OASIS WS-Trust Specification 1.3, OASIS Standard, 19th March 2007.
17. Lockhart H., Andersen S., Bohren J., Sverdlov Y., Hondo M., Maruyama H., Nadalin A., Nagataram N., Boubez T., Morrison K.S., Kaler C., Nanda A., Schmidt D., Walters D., Wilson H., Burch L., Earl D., Baja S., Prafullchandra H., Web Services Federation Language (WS-Federation), Version 1.1, December 2006.
18. M. Schumacher, A. de Oliveira e Sousa, I. Constantinescu, T. van Pelt, B. Faltings, *Distributed Directories of Web Services*, Book CASCOM: Intelligent Service Coordination in the Semantic Web, Whitestein Series in Software Agent Technologies and Autonomic Computing, Birkhäuser Verlag, Basel.
19. Cybersanté Suisse: Normes et Architectures, Premières recommandations. 19 March 2009
20. Dantu R., Oosterwijk H., Kolan P., Husna H., Securing medical records in Network Security, Volume 2007, Isue 6, June 2007, Pages 13-16.
21. The official website of the TripCom project, http://www.tripcom.org.