Hes·so // VALAIS WALLIS
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences Western Switzerland

REQUIREMENTS AND DESIGN OF INTEROPERABILITY CASE STUDIES

(RCSO IHETOOLS Deliverable D.2.1)

IIG-TR 2011.02

Bruno Alves[1] Michael Schumacher[2]

JUNE, 2011

BUSINESS INFORMATION SYSTEMS TECHNICAL REPORT

Business Information Systems Institute • HES-SO // Wallis
University of Applied Sciences Western Switzerland
TechnoArk 3 • 3960 Sierre • Switzerland

phone +41 (27) 606 90 01 fax +41 (27) 606 90 00 iig@hevs.ch http://iig.hevs.ch

[1]IIG, HES-SO // Wallis, Sierre, Switzerland, bruno.alves@hevs.ch
[2]IIG, HES-SO // Wallis, Sierre, Switzerland, michael.schumacher@hevs.ch

Abstract

This document contains the high level requirements for three different scenarios making potential use of Integrating the Healthcare Enterprise (IHE) technologies. The objective is to give a detailed view of one scenario and briefly discuss about the two others. Every scenario is described in terms of background, objectives, requirements and architecture.

Keywords: IHE, Medicoordination, GDEMANDE, Drug-Exchange Platform

Contents

1 Introduction

1.1 Background

The goal of the IHETOOLS project is to propose a methodology and a catalog of architectural solutions targeting some known interoperability issues in the e-Health sector by leveraging IHE profiles (Integrating the Health Enterprise) in specific use cases.

As for the operational objectives, we seek providing three non-trivial prototype designs and one implementation of a particular use case in order to show the advantages and a discussion of IHE profiles. Concretely, as a first idea, we want to extend the current MediCoordination prototype (that currently uses only the XDS IHE Profile) with the following profiles: NAV (asynchronous notifications of availability), PIX (mapping of heterogeneous patient identifiers across different domains), ATNA (audit and trail), CT (consistent time between domains), together with better security integration (probably by leveraging EUA and XUA capabilities in a controlled environment).

In this report, we will explore three scenarios and detail the requirements in terms of the necessary IHE infrastructure.

1.2 Integrating the Healthcare Enterprise Profiles

There are numerous IHE profiles covering a wide selection of fields. However, the following scenarios will make use of a limited number among them. The required profiles are typically those used in the context of document sharing. The descriptions are directly taken from the "IHE IT Infrastructure (ITI) Technical Framework" books, available at [1][2] and [3].

1.3 XDS

Cross-Enterprise Document Sharing enables a number of healthcare delivery organizations belonging to an XDS Affinity Domain (e.g., a community of care) to cooperate in the care of a patient by sharing clinical records in the form of documents as they proceed with their patients' care delivery activities. Federated document repositories and a document registry create a longitudinal record of information about a patient within a given XDS Affinity Domain. This profile is based upon ebXML Registry standards and SOAP. It describes the configuration of an ebXML[1] Registry in sufficient detail to support Cross Enterprise Document Sharing.

1.4 PIX

The PIX profile supports the cross-referencing of patient identifiers from multiple Patient Identifier Domains. These cross-referenced patient identifiers can then be used by —identity consumer‖ systems to correlate information about a single patient from sources that —know the patient by different identifiers. This allows a clinician to have more complete view of the patient information.

1.5 PDQ

Patient Demographics Query provides ways for multiple distributed applications to query a patient information server for a list of patients, based on user-defined search criteria, and retrieve a patient's demographic (and, optionally, visit or visit-related) information directly into the application.

1.6 CT

Consistent Time Profile defines mechanisms to synchronize the time base between multiple actors and computers. Various infrastructure, security, and acquisition profiles require use of a consistent time base

[1] http://www.ebxml.org/ (viewed on 15.03.2011)

on multiple computers. The Consistent Time Profile provides median synchronization error of less than 1 second. Configuration options can provide better synchronization. The Consistent Time profile specifies the use of the Network Time Protocol (NTP) defined in RFC1305.

1.7 ATNA

Audit Trail and Node Authentication establishes the characteristics of a Basic Secure Node:

1. It describes the security environment (user identification, authentication, authorization, access control, etc.) assumed for the node so that security reviewers may decide whether 640 this matches their environments.
2. It defines basic auditing requirements for the node
3. It defines basic security requirements for the communications of the node using TLS or equivalent functionality.
4. It establishes the characteristics of the communication of audit messages between the 645 Basic Secure Nodes and Audit Repository nodes that collect audit information.
5. It defines a Secure Application actor for describing product configurations that are not able to meet all of the requirements of a Secure Node.

This profile has been designed so that specific domain frameworks may extend it through an option defined in the domain specific technical framework. Extensions are used to define additional audit event reporting requirements, especially actor specific requirements. The Radiology Audit Trail option in the IHE Radiology Technical Framework is an example of such an extension.

1.8 EUA

Enterprise User Authentication defines a means to establish one name per user that can then be 605 used on all of the devices and software that participate in this integration profile. It greatly facilitates centralized user authentication management and provides users with the convenience and speed of a single sign-on. This profile leverages Kerberos (RFC 1510) and the HL7 CCOW standard (user subject). User authentication is a necessary step for most application and data access operations and streamlines workflow for users. Future profiles will deal with other security issues, such as authorization management.

1.9 XUA

Cross-Enterprise User Assertion provides a means to communicate claims about the identity of an authenticated principal (user, application, system...) in transactions that cross-enterprise boundaries. To provide accountability in these cross enterprise transactions there is a need to identify the requesting principal in a way that enables the receiver to make access decisions and generate the proper audit entries. The XUA Profile supports enterprises that have chosen to have their own user directory with their own unique method of authenticating the users, as well as others that may have chosen to use a third party to perform the authentication.

2 Scenarios

The scenarios described in the following subsections are exploratory test cases for the usage of IHE technologies in concrete situations.

2.1 Extension of the Medicoordination Scenario

The MediCoordination[2] project tries to complement the Swiss eHealth strategy by collaborating mainly with regional medium-sized hospitals and smaller partners in the health system, where data exchange has not been an as important subject as in large University hospitals that often already exchange health data with external actors. By communicating with several actors in the health system, a few scenarios for health data exchange could be identified, where a simple implementation brings a clear added value for all partners. This allows for testing the infrastructures in parallel to the creation of the eHealth strategy also for smaller actors in the health system to gain experience with these tools and potential problem. This project, which has currently limited its scope to the French-speaking part of Switzerland, has successfully been deployed and was described in some papers [4][5] and [6]

The focus of this extension scenario is on adding security, patient management and attachment handling to the existing platform.

2.2 Agent-based Drugs-Exchange Platform

Almost every hospital is well acquainted with the problem of drugs stacking up in stocks and ending in trash. These stocks are considered lost after drugs expire. This conceptual project strives to gather hospitals in a large drug-exchange platform. The objective is to organize continuous auctions in order to sell valid but unnecessary drug reserves to other institutions which may need them. Humans and institutions are impersonated by auctioning agents, which follow strict rules in order to maximize the value of the stocks according to local policies.

2.3 Gestational Diabetes Monitoring Platform

This use-case targets the specification of a pervasive healthcare system (PHS) based on a multi-agent system infrastructure to support pregnant women with Gestational Diabetes Mellitus (GDM). Such an infrastructure utilizes a mobile application to monitor patients affected by GDM, whose parameters are then sent to and analyzed by cognitive agents. A symbolic reasoning approach is used to formalize the events happening in the system, the entities participating in the interaction and the agent cognitive model for continuous monitoring of GDM. Such a cognitive model is based on deductive treatment adjustment rules to provide doctors with indications about the patient's treatment and on abductive rules to provide a diagnosis of the illness' current state.

[2] http://www.medicoordination.ch/

3 Scenario 1: Extension of the Medicoordination Architecture

This section presents the requirements and extended architecture of the Medicoordination project. The new architecture is partly based on IHE's whitepapers: "Access Control" [7] and "A Service-Oriented Architecture (SOA) View of IHE Profiles" [8].

3.1 Background

The architecture of the MediCoordination prototype is based on IHE Profiles, especially on the Cross-Enterprise Document Sharing (XDS.b)[1 pp. 69-99]. Document producers upload structured documents with metadata onto a server (registry and repository) and consumers download them from the repository. For this, existing open source tools were used: *iheprofiles*[3] from *OpenHealthTools*[4] and *XDS.b Document Registry and Document Repository Solution Accelerator*[5] from Microsoft.

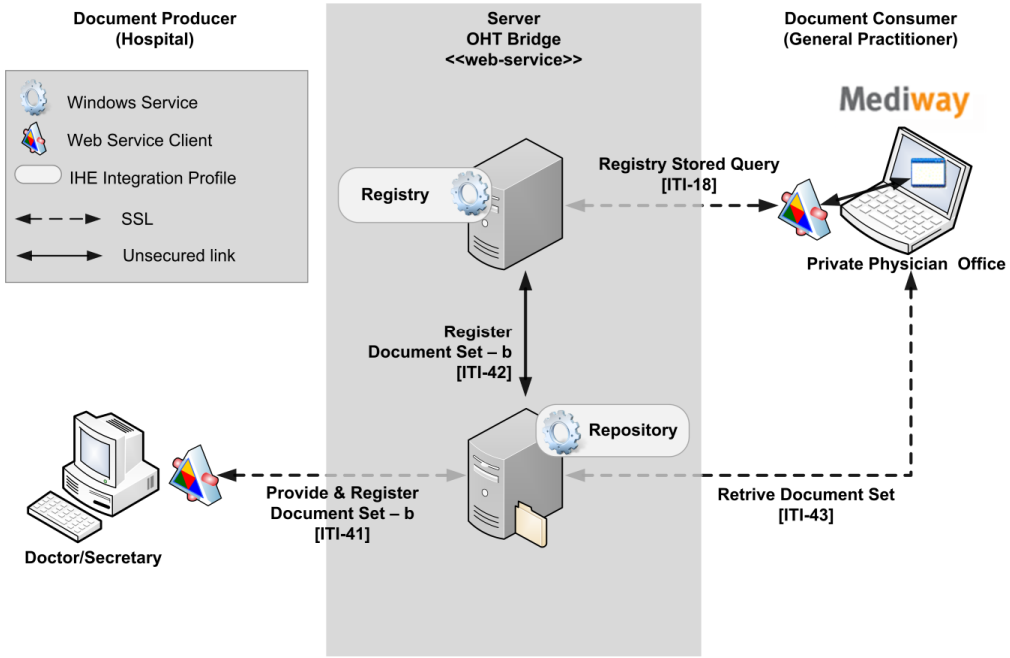

FIGURE 1. ARCHITECTURE OF THE MEDICOORDINATION PROTOTYPE

Medicoordination was primarily set as a test case to assess IHE technologies. The first prototype was lacking a sufficient security infrastructure and had no patient-management facilities. Furthermore, it provided no mechanism to embed attachments, important when storing radiological results, for instance. The current section describes the extension of Medicoordination towards a second prototype, which was effectively deployed and tested in real settings.

3.2 Objectives

The primary objective of this extension is to fill-in the gaps highlighted in the original architecture specification. The key objectives of this implementation are to provide solutions for the following points:

1. Patient management
2. Attachment handling

[3] https://www.projects.openhealthtools.org/sf/projects/iheprofiles/
[4] http://www.openhealthtools.org/index.htm
[5] http://ihe.codeplex.com/

3. Handle different document types
4. Security compliances

The first Medicoordination implementation didn't provide any patient management facilities. No mechanism to match identities coming from different domains was implemented and it was not possible to retrieve demographics either.

Attachment handling is definitely worth considering, since it permits attaching external files, such as radiological data to documents forming thus an episode of care.

The first prototype implementation did only accept "Discharge Summary" type files. The new extension provides a few predefined types and can perform access-control related verifications on these.

Since security is of paramount importance in healthcare, it is essential that this implementation is compliant with current security practices. Amongst all, it is important to provide a strong access-control mechanism, but also authentication and accountability primitives.

3.3 Requirements

This section details the basic requirements for implementing the proposed architecture taking into account the preceding points.

3.3.1 Patient Management

The patient management infrastructure must have the ability to both match identifiers coming from different hospital domains, but also to provide demographic information about them. This is naturally done with PIX and PDQ, that handle respectively identity matching and demographics querying.

Patient information is stored in a patient index often called Master Patient Index(MPI) or Enterprise Master Patient Index (eMPI). In this implementation, the index is called Local Patient Index (LPI) and its role is storing patient identifiers and demographics for a single domain. Since the repository may receive documents from unknown patient domains, LPI may also play a role in the matching of internal patient identifiers to the external ones. This component is often related as the patient identity cross-reference manager, or PIX Manager in IHE terminology.

3.3.2 Attachment handling

The XDS communication protocol already provides a nice way to link attachments to another document. This is done by appending new `Document` entries in the `ProvideAndRegisterDocumentSet-b` [3, ITI-41] request and linking them to the parent document.

Since it is preferable to hide specifics to the users of the platform, we need a frontend web service interface layer that provided primitives accepting the original document and its attachments as a single interface method call. This method should then create a HL7 CDA-CH R2 document and export it to the repository.

3.3.3 Document types

Document types are assigned by setting a `code` element in the header of the CDA document in order to match the desired type. Content types are given as codes of a custom coding scheme and have no other meaning than the one given by the project team. As an example, to define a Discharge Summary inside the CDA document, one would insert the following line.

```
<urn:code code="DISCHARGE_SUMMARY" codeSystem="<MyCodeSystemOID>" />
```

`<MyCodeSystemOID>` matches the OID some user-defined coding scheme and the code attribute represents one of its values. Setting this value is essential, because it allows the access-control system to apply security decisions on a document type basis.

The `contentTypeCode` field of the XDS request must also match this code.

3.3.4 Security compliance

In order to improve over Medicoordination security, it is important to reconsider key aspects such as authentication, authorization and accountability mechanisms. The security infrastructure must also be revised in order to support and reflect these changes.

3.3.4.1 Authentication

Authentication requests were handled by the web container in the previous Medicoordination implementation and used to carry on credentials as a username/password pair. In this extension, authentication requests have to be made to a central authentication authority.

The authentication mechanism we want to implement is based on mutual certificates. The authentication service and the client must present certificates signed by the same certification authority. The authentication service can then extract parts of the user certificate and match them against user information stored in a LDAP repository, such as the email or certificate fingerprint, for instance.

The authentication server is also responsible for issuing the SAMLv2 tokens that will be sent along all XUA transactions flowing through the components of the system.

3.3.4.2 Authorization

In order to take patient consent into account, it is necessary to extend CDA documents with information about their confidentiality level. Users are associated to roles that may be authorized or denied some specific actions, like reading or modifying documents.

The Confidentiality code can be specified in a CDA document by setting the `confidentialityCode` element to match a code in a custom coding system. In this extension, we propose the following confidentiality levels:

public	Document contain does not contain any restricted material
normal	Document contains normal medical data subject to consent
restricted	Document contains data which is restricted to authorized roles only
sensitive	Document contains sensitive data that is accessible to trustes parties only
taboo	Document contains highly sensitive data that may not be exposed

Based on this designation, the access-controlling system can check whether a healthcare professional can access the data or not. Access controlling cannot be set for every professional, but only for roles. Those roles include: `medical_doctor`, `general_practitioner`, `trusted_entity` and `patient`.

- The `medical_doctor` role represents a physician working in a hospital. S/he can read, modify or post documents in the system, but is constrained to the rules applied by a patient on his/her data ;
- The `general_practitioner` also represents a physician, but working in a different setting, typically a care practice or care center. A general practitioner cannot modify or post documents, but can still read them. GP's are also constrained by access limitations imposed by the patient ;
- The `trusted_entity` role represents someone who is fully trusted by a patient ;
- A `patient` as the name implies is a physical person going to a care center in order to get a treatment. Patients can read all their record data and can also change access rights.

Roles can perform up to four operations, which are:

- READ: reading documents ;
- MODIFY: updating documents ;
- POST: posting new documents ;
- ACCESS_RIGHTS: modifying access rights for a particular type of data.

The table below synthesizes their rights:

	READ	MODIFY	POST	ACCESS_RIGHTS

medical_doctor	*	*	*	
general_practitioner	*			
trusted_entity	*	*	?	
patient	*			*

TABLEAU 1. ROLE-ACTION MAPPING MATRIX

Every user in the system is associated to one of the roles above. Roles are sent along every SOAP request, inside a SAMLv2 assertion, as stated in the XUA profile specification.

3.3.4.3 Accountability

Accountability must be performed at nodes that handle medical data, such as the repository, the registry and the web service front-end.

A separated Audit Record Repository (ARR) is kept in a separate server and is accessible to all the logging nodes. Events to be recorded are specified in the ATNA section of the XDS profile specification and also includes start or stop events.

3.3.4.4 Infrastructure

In order to build a secure infrastructure, it is essential to ensure the confidentiality, integrity and availability (CIA principle) of the data that is transferred.

Confidentiality is typically ensured with encryption. Communications between nodes must be encrypted to comply with the IHE ATNA profile directives. A secure TLSv1 channel must be established between all actor nodes participating in the profile. There are basic requirements regarding supported cipher suites. These are described in the WS-I Basic Security Profile 1.2 (WSIBasicSecurity). This specification describes what cipher suites are mandatory, recommended or discouraged.

Mandatory cipher suites are widely implemented, secure and interoperable. They include:

- TLS instances, which are not FIPS-compliant, must at least support the TLS_RSA_WITH_3DES_EDE_CBC_SHA cipher suite ;
- TLS instances, which are FIPS-compliant, must support the TLS_RSA_FIPS_WITH_3DES_EDE_CBC_SHA cipher suite.

These cipher suites are the default ones used by the extension prototype. Since the 3DES algorithm is to be superseded by the AES encryption algorithm, it is recommended to use the TLS_RSA_WITH_AES_128_CBC_SHA. In java, the AES encryption is subject to local jurisdiction restrictions and must be downloaded separately, as a JCE (Java Cryptography Extension) provider. Some cipher suites are known to be vulnerable to man-in-the-middle attacks and should be avoided:

- SSL_RSA_WITH_NULL_SHA
- TLS_RSA_WITH_NULL_SHA
- SSL_RSA_WITH_NULL_MD5
- TLS_RSA_WITH_NULL_MD5

Ciphers suites including MD5 digests or cipher suites using 40 or 56-bit keys must also be avoided, because their relative weakness can be broken by brute-force type attacks.

The IHE ATNA profile advocates that TLS authentication should preferably be bi-directional. The server and clients must provide X509 certificates to authenticate themselves in the system. Note that these certificates are not used for the authentication of the user, but rather for establishing a secure communication channel where credentials can be safely exchanged. Only the health professional and patient credentials between the external systems and the extended platform are accepted as credentials. All the ATNA secure node actors must be isolated into distinct servers and present distinct certificates signed by an approved CA (Certification Authority). Revoked certificates must be kept in a remotely-accessible point, preferably on the authentication service

Integrity is semi-automatically handled by the SOAP stack. It is possible to direct it to sign specific parts of a message and to encrypt it. Encryption at the message level is important even if data is already

encrypted by the TLS protocol. In the OSI stack, the SOAP message has to traverse a few layers before it gets encrypted or decrypted. Between these layers, the data is in clear-text and can thus be exploited.

Availability is not handled by this extension, since it is a proof-of-concept and not an already deployable infrastructure.

3.4 Architecture

The improved architecture presented below builds on Medicoordination, but adds user indexes, an access control server ACS (including PEP, PIP and PAP) and an authentication proxy.

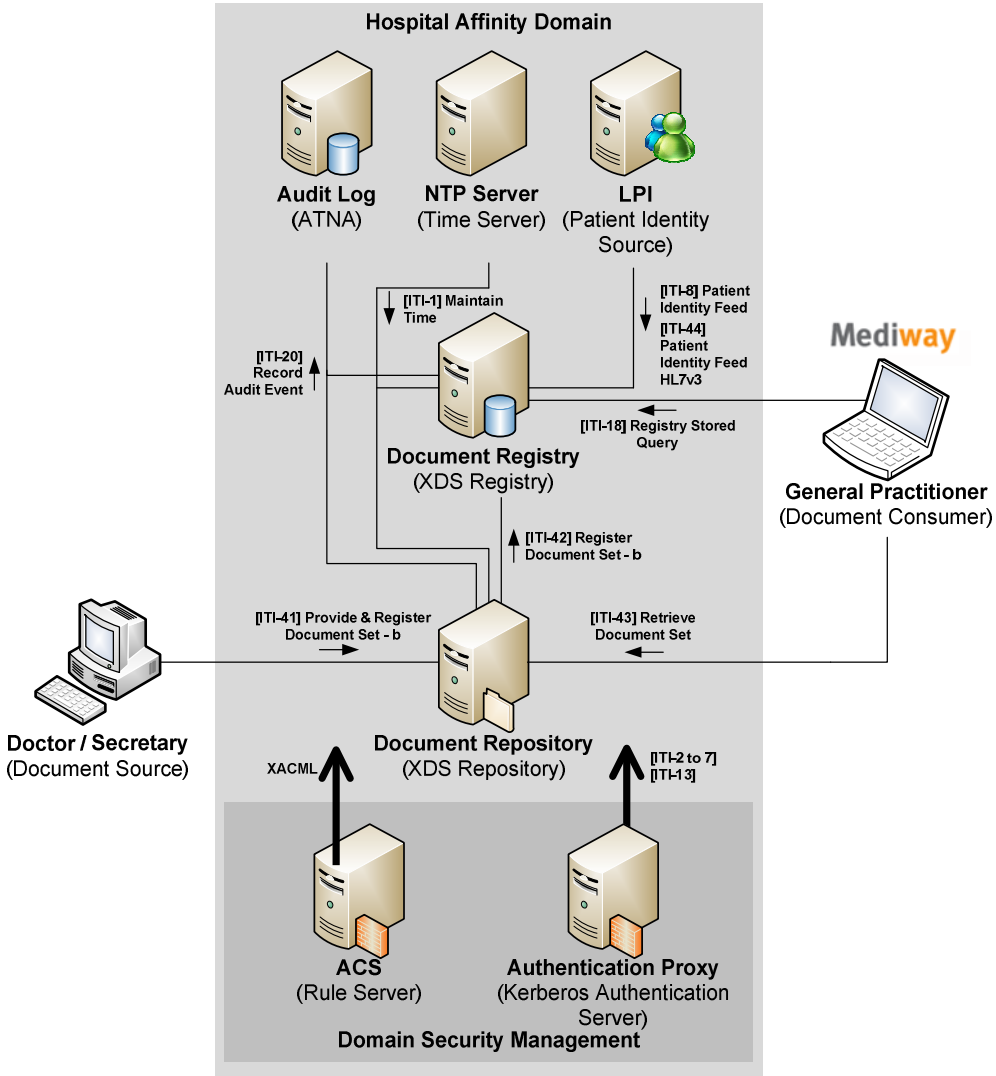

FIGURE 2. EXTENDED SCENARIO OVERVIEW

Medicoordination relied solely upon XDS. This extension proposal intends to leverage the functionality of CT, PIX, PDQ, EUA, XUA and ATNA for improving aspects like patient identification, health professional authorization and management and security.

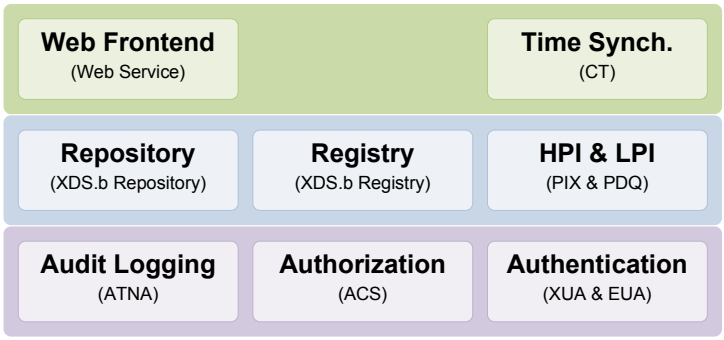

FIGURE 3. LINKING IHE PROFILES AND TECHNOLOGIES

The stack in the figure above shows the various parts of the prototype and their relationships with IHE Profiles and technologies. At the top of the stack, a Java web service proposes an interface exposing functionalities for submitting, retrieving and querying documents as well as functionalities for creating, deleting and searching for patients. A time synchronization server maintains a time reference, which is valid for every server participating in the extension.

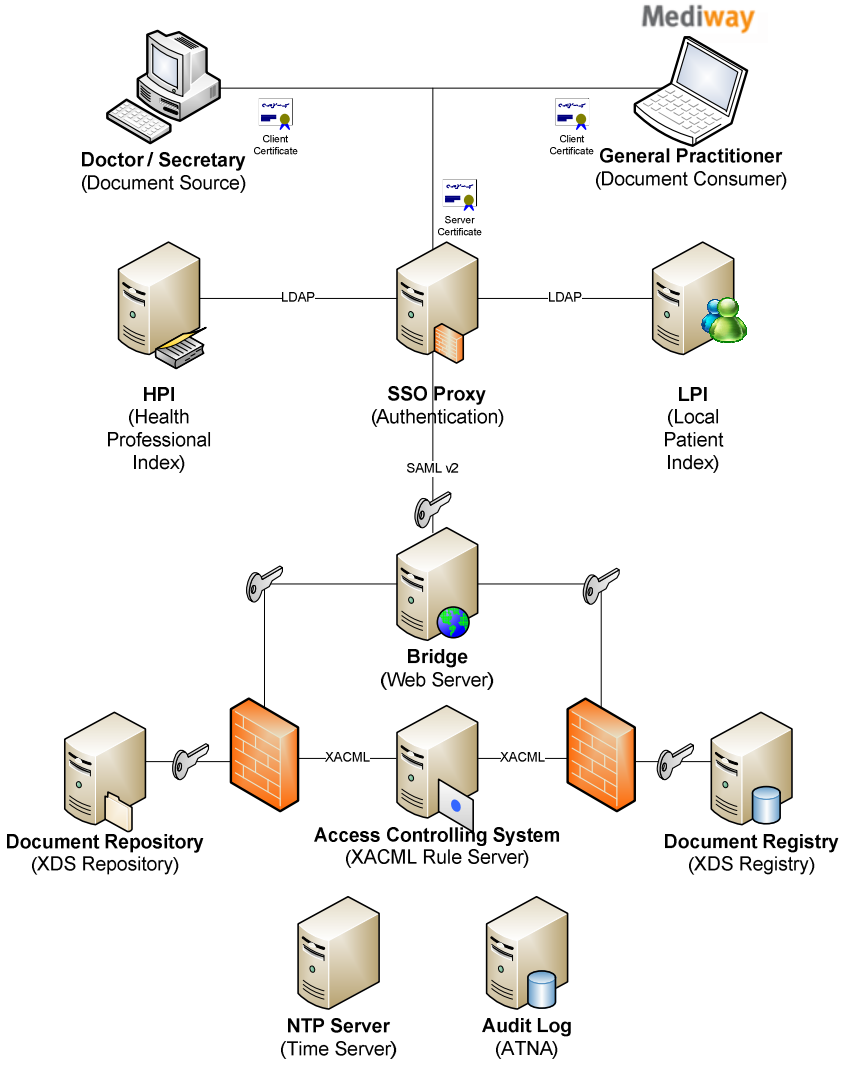

FIGURE 4. EXTENDED ARCHITECTURE OVERVIEW

The architecture illustrated above relies on many subsystems, which are described below.

3.4.1 Single Sign-On Reverse Proxy

The Signle Sign-On Reverse Proxy (SSO Proxy) is a proxy server in a reverse configuration. It channels all the traffic through a single point and is in charge of sending authentication requests to the authentication service and forwarding requests along with the returned security token (SAMLv2).

3.4.2 Authentication Service

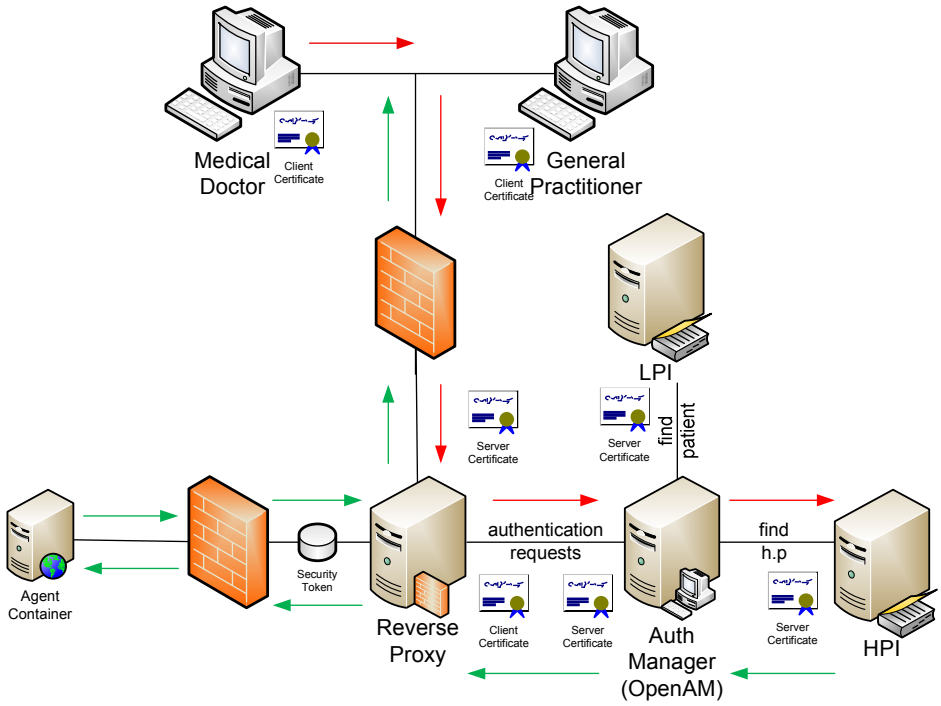

FIGURE 5. OVERVIEW OF THE AUTHENTICATION SYSTEM

The authentication service is shown here as a part of the SSO Reverse Proxy, but it typically sits in a different system, although it is tightly connected with the proxy. The authentication service uses the HPI source to match user certificate attributes in authentication requests. All connections from the reverse proxy to the internal subsystems (web containers) are then impersonated with security tokens.

3.4.3 Health Professional and Local Patient Indexes

The Health Professional Index (HPI) and Local Patient Index (LPI) are both Policy Information Points for the ACS subsystem. These repositories keep respectively information about health professionals and about patients.

3.4.4 Patient and HP Portals

The bridge is a web server exposing a web service for accessing all functionality. It is hidden from the internet by the reverse proxy. The portals should exhibit minimal functionality, since they are showcases for the functionality.

3.4.5 Document Repository

The document repository is responsible for storing data along with attachments.

3.4.6 Document Registry

The document registry is responsible for registering new documents.

3.4.7 Access Control System

The Access Control System (ACS) checks that specific users have the required access privilege to data they request. This system is composed of a rule server and of policy enforcement point acting as firewall to

the repository and registry and one or more information points, streaming attributes to security assertions. The ACS is composed of three parts, which can be distinct or not:

- The Policy Administration Point (PAP) is an endpoint that allows users to change ACS policies and specify which roles have access to what ;
- The Policy Information Point (PIP) is generally associated to the LDAP repository and can stream attributes from the LDAP repository down to the ACS in order to match request's declared attributes to the ones stored in the system, that help deciding on the status of the request ;
- The Policy Enforcement Point (PEP) receives the original request to the XDS subsystem. It extracts the SAMLv2 token and forwards it to the ACS along with attributes about the document. After receiving the decision, it either forwards the request or rejects it.

3.4.8 Time Synch Server

In order for the audit logs to be meaningful, all the subsystems need to be perfectly synchronized.

3.4.9 Audit Log server

An audit log server records every major event in the system.

4 Scenario 2: Agent-based Drugs-Exchange Platform

This scenario specifies requirements for the implementation of an agent-based auctioning drugs-exchange platform based on IHE.

4.1 Background

The topic of rising drug costs has received significant attention in the last couple of years. To tackle this problem, several strategic alternatives have been in study:

- Network pharmacy contract renegotiation ;
- Preferred drugs lists ;
- Purchasing coalitions.

One key domain where cost reduction can be significant is the management of the drug stock.

4.2 Objectives

The objective of this scenario is to provide a supplementary motivation to use IHE technologies coupled to agents in order to take advantage of existing infrastructures.

The goal of this scenario is to create an architecture where agents are distributed and their policies are controlled locally. Auctions remain centralized. It is thus necessary to explicit requirements for the agents and for the auction house.

4.3 Requirements

The requirements elicited in this section are informative and do not go further down into much detail. The objective here is to give an idea how to solve this scenario with a simple architecture based on IHE.

4.3.1 Agent

An agent is a self-containing infrastructure composed of a document repository and of a web container holding a copy of the agent service executable. The service configures itself from the local configuration repository (XDS Repository) and shares information with the Auctioning House using a very specific protocol, which has yet to be defined.

4.3.1.1 XDS Infrastructure

Agents are entities acting on behalf of the hospitals. They make decisions according to the local policies that are usually stores along with configuration data. Since configuration and data for the agents are stored in a decentralized fashion, records are kept in a XDS Repository at the hospital's node and are thus under hospital's responsibility.

In order for the Auctioning House to be able to share information with the agents, documents are registered in a centralized registry. It should be noted that ATNA does not define any particular event format for auctioning events. It solely proposes event structures for IHE transactions. However, it is possible to define new events and log them into the centralized Audit Record Repository. Every agent node can also define local audit repositories.

4.3.1.2 Security

Key security aspects that are needed are authentication, authorization and communication security.

First, agent nodes must be authenticated in order to participate to auctions. This is a strong requirement, because otherwise, it is not possible to track transactions they perform (anonymous transactions). As for the other scenarios, it is an advantage to use a SSO paradigm in order to avoid multiple authentication requests that could overwhelm the server.

An authorization mechanism should be deployed in order to limit the possibilities of the agents inside the auction, and prevent thus potentially malicious agents from taking advantage of security breaches.

Channel security should be applied conformingly to the ATNA specification. Agents should only communicate though protected channels and all transactions accordingly audited.

4.3.2 Auction House

The auction house must provide tools and agents for handling multiple auctions. It is thus composed on a configuration repository also linked to a central registry and multiple agents, each handling a single auction. All the agents exchange messages through the auction bus. The bus routes the messages according to the target address and local policies.

4.3.2.1 XDS Infrastructure

The XDS infrastructure is composed of a central registry holding configuration entries for each agent record. There is also a repository holding the configuration of the auctioning house itself.

4.4 Architecture

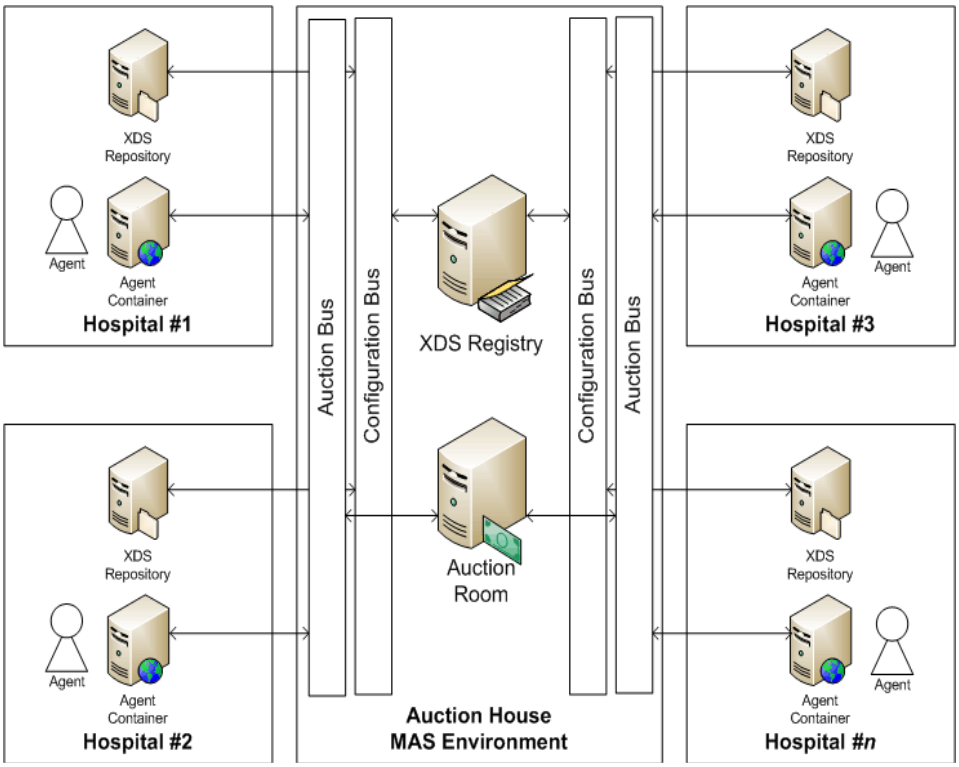

FIGURE 6. DRUGS EXCHANGE PLATFORM

The architecture illustrated above allows agents to be independent, by keeping intelligence locally. However they are allowed to participate in remote auctions. This architecture prevents other agents from cheating on behalf of other agents.

4.4.1 Hospital Agent

The hospital agent environment comprises a XDS Repository for the configuration and local data (including policies) and a web container running the agent executable.

4.4.1.1 Configuration

The configuration is inserted into the local repository, but is registered in the centralized XDS Registry of the auction house. Auctioning houses have a permanent knowledge of every agent.

4.4.1.2 Container

The web container holds the executable code of the agent. The agent uses local stored knowledge and policies in order to maximum its return value.

4.4.2 Auction House

4.4.2.1 Registry

A central registry keeps track of every agent transaction. Configuration data is registered here and can then be accessed by the auction house agent.

4.4.2.2 Auction House Agent

The auctioning agent is in charge of organizing the multiple auctions. Every auction request is started by an hospital and accepted by peers, but only the auctioning agent has the control on them

4.4.2.3 Auctioning Agents

An auctioning agent is in charge of a single auction. The agent handles the various steps of the auction and enforces that other agents play their roles.

4.4.2.4 Audit Record Repository

Every auction house contains one or more audit repositories that can be used for all auditing operations, even though they are designed to log auctioning events.

4.4.3 Communication channels

Agents communicate with the auction house via a secure custom SOAP protocol. The auction bus is a middleware component in charge of routing the requests, according to the address of delivery.

Agents may also share configuration with the central auction house via a configuration bus, which also uses a custom SOAP protocol.

5 Scenario 3: Gestational Diabetes Monitoring Platform

GDEMANDE [9][10] is a personal health system designed to help women suffering from gestational diabetes. This condition is treatable and patients can live normally with some diet adjustments and specific medicine. Correct treatment requires detailed information about physiological values such as glucose level or blood pressure.

5.1 Background

The scenario presented herein is based on the work described here (Bromuri, Michael, Kostas, & Ruiz, 2011) and here(Bromuri, Schumacher, & Stathis, Pervasing Healthcare using Self-Healing Agent Environments, 2011).

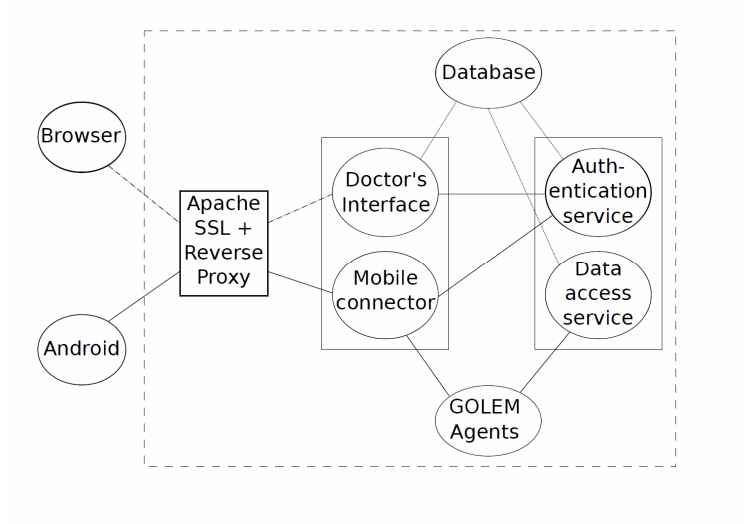

FIGURE 7. CURRENT ARCHITECTURE OVERVIEW

The current architecture of the GDEMANDE project consists of:

- A web interface for monitoring patients ;
- An Android mobile platform sending physiological values ;
- A GOLEM [11] instance reasoning on the input data ;
- A storage for historical values.

As part of the infrastructure, several services are available:

- A data access service shared by the agents and exposed web services ;
- An authentication service ;
- A mobile connector providing an abstract interface between the mobile and the agent platforms ;
- A web-application back-end allowing healthcare professionals and patients to monitor values ;
- A reverse proxy based on Apache2.

Connections to the services are channeled through the reverse proxy/firewall and are TLS-secured with single point (server) certificates. Credentials are sent as a username/password pair.

5.2 Objectives

The architecture described in this scenario is an advisory improvement over the current implementation. The improvements target foremost security and information control. Specific parts that are candidates to IHE implementations are: *accountability* and *security*.

5.3 Requirements

This section details the technological and security requirements for the extension scenario.

5.3.1 Accountability

Accountably is a subject where IHE technologies perform greatly. Accountability in this extension scenario is mainly targeted at auditing and channel security and would include ATNA and XUA profiles. The ATNA profile specifies how to secure the connection between two different nodes of a safe system and how to produce auditing events. The XUA profile is used by internal systems requiring auditing capabilities to carry information about roles, users and rights. It makes heavy use of SAMLv2 assertions.

5.3.2 Security

As a requirement for secure systems, all connections between subsystems or with external systems must support at least line encryption though protocols such as TLS. The minimal requirements in terms of security should at least meet those specified in the WS-I Basic Security Profile 1.0 specification and the ATNA profile.

Users and roles must clearly be defined and used within the system. This is an essential constraint, because it allows transactions to be audited. A central authentication server may be used in conjunction with a token service providing SAML tokens carrying authenticated credentials. Authentication credentials may be given by a user either by a username/password pair or by a client certificate or smartcard token (PKCS#11).

Authentication is used to know which actor is carrying which action and should not guarantee full access to every kind of data. It is necessary to ensure that every health professional has only access to data, which has been tagged as exposable by a patient. Of course professional advice may be necessary in order to prevent patients from hiding information which disclosure would be vital in the diagnosis.

Furthermore, in order to apply its deductive logic, it is necessary that the Multi-Agent System (MAS) has access to every piece of data. Thus, it is mandatory to strongly secure the MAS and disallow unwanted accesses.

5.4 Architecture

The figure below illustrates the proposed architecture. The first major change is the adding of the User Management and Authentication services, that include a MPI, HPI and the authentication service. These services are available from the reverse proxy. The second change is the adding of a Policy Enforcement Point between the Healt Professional Portal and the database. The role of the PEP is to make sure that a user cannot access unauthorized data and that a patient can decide who has access and to which data.

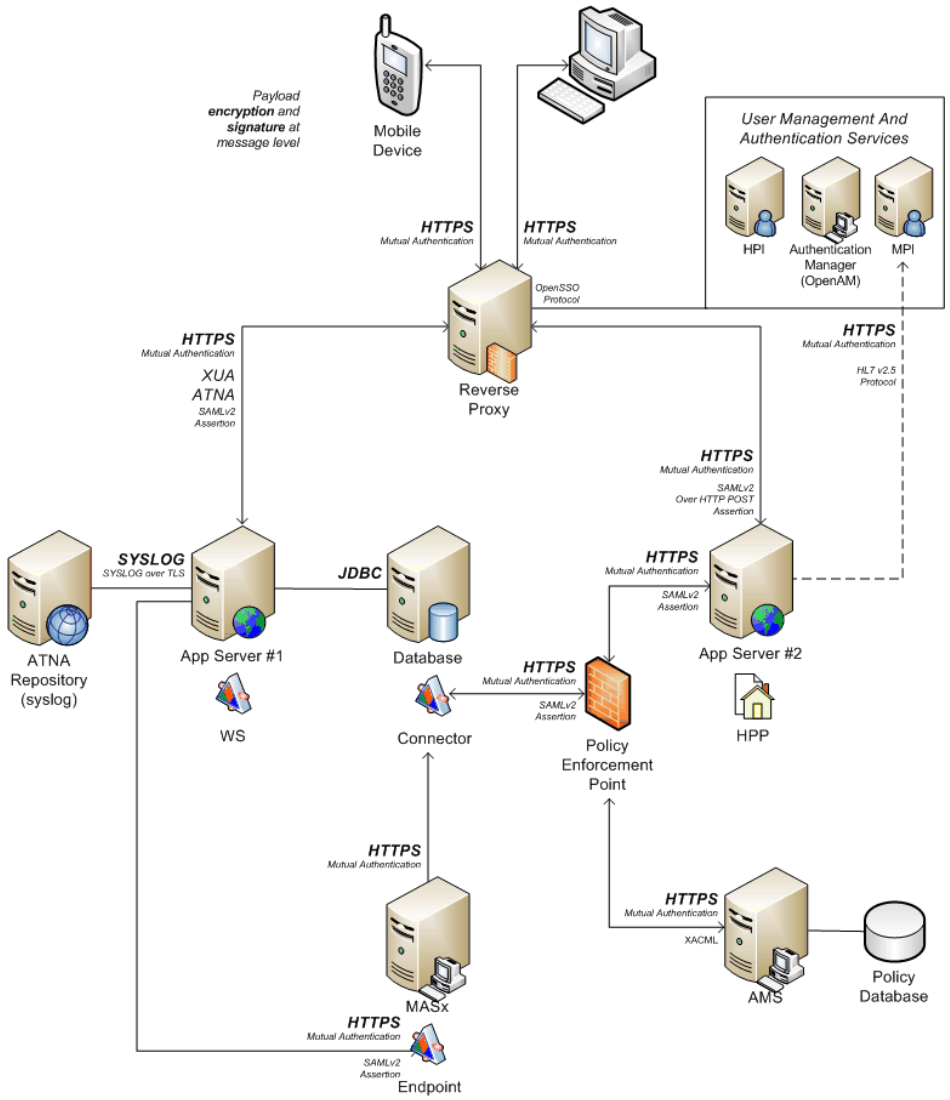

FIGURE 8. IHE-BASED ARCHITECTURE

In the architecture illustrated in the figure above, most systems already introduced in the GDEMANDE project remain at the same place. Improvements consist in an additional information access control system between the web interface and the data access layer (database connector) and separated servers for the authentication, health professional and patient identity stores.

The service accepting connections from the mobile appliances is to be connected to an audit repository and is in charge of logging specific events. The audit repository should be generic enough to accept logging events, which do not conform to the IHE ATNA specification.

5.4.1 External appliances

These appliances are terminals connecting to the internal subsystems from the Internet. Mobile appliances feed the system with physiological parameters, such as glucose level or blood pressure. These values are then used by the multi-agent system, which role is to analyze the flow of data and extract patterns that may trigger alerts.

Health professionals can connect to a web portal and monitor their patients. Observable data must be regulated by an access controlling subsystem, so that patients may decide whether particular values or pieces of information are too sensitive to be disclosed to the medical staff.

5.4.2 Authentication proxy

The authentication proxy is composed of a proxy in a reverse configuration coupled to an external authentication service. Attempts to connect to the web service or to the portal immediately trigger

authentication requests. The scenario is however a bit different, depending on the type of accessed resource.

When accessing the web service, the user must first provide its credentials. If the authentication succeeds, the authentication service sends a security token back to the proxy (a SAMLv2 token). This token is then used along XUA assertions in order to carry information for accountability.

On the other part, when a user tries to access the portal, the proxy first checks that there is an ongoing session. If this is not the case, then the user's browser is redirected to a login page. After the authentication is successful, the authentication service sends back a security token (also a SAMLv2 token) and the communication can proceed. However, the assertion is sent to the web container in SAMLv2 POST requests.

5.4.3 Multi-agent system

The multi-agent system takes values from the mobile appliances and from patient's history to search patterns that would trigger alerts and would require specific handling by the physicians. Even though it does not store values, this subsystem is sensitive because it handles private data that can otherwise be accessed by unauthorized parties.

5.4.4 Database system

The database system is the central element of the architecture and is shared by the MAS, the HPP and Web Service (WS). It is directly accessed through a JDBC driver connector by the WS, while an interface is proposed between it and the MAS and HPP. Since this component is of paramount importance, it is essential to keep it clearly separated from the other subsystems. All connections and transactions MUST be controlled and secured.

5.4.5 Healthcare Professional web interface

The Health Professional Portal (HPP) is web interface that enables caretakers and patients to monitor values and alerts. The frontend is tightly connected with the database and is accessible from the Internet, so its access must be secured and regulated.

5.4.6 Mobile data submission web service

This web service allows the mobile appliances to share the collected physiological values with the multi-agent subsystem. It does so by first storing the values on the database. Unlike other subsystems, the web service has a direct interface to the database (without a connector). All transactions must be logged according to the ATNA audit format specification.

6 Conclusion

This document presented three different scenarios that are potential candidates for using IHE technologies. As a result it is observable that profiles such as ATNA and XUA are easily portable to many very different scenarios. For the other profiles presented here, the situation is quite different, because most of them are very specific to a certain domain. Even if XDS can be used for all document sharing tasks it is not optimal and requires a great deal of efforts to deploy such infrastructures.

As a conclusion, designers should keep in mind that IHE profiles are designed with some very specific use cases in mind and are generally hard to adapt to different scenarios. So, it may be preferable in some cases to rely on different technologies or specifications.

References

[1] *Various authors,* **IHE IT Infrastructure (ITI) Technical Framework : Integration Profiles***,* IHE International, 2010, Vol. 1. ITI TF-1.

[2] *Various authors,* **IHE IT Infrastructure (ITI) Technical Framework : Transactions ITI-1 through ITI-28***,* IHE International, 2010, Vol. 1. ITI TF-2a.

[3] *Various authors,* **IHE IT Infrastructure (ITI) Technical Framework : Transactions (cont'd) ITI-29 through ITI-50***.* IHE International, 2010, Vol. 1. ITI TF-2b.

[4] *Bruno Alves, Henning Müller, Michael Schumacher, David Godel* and *Omar Abou Khaled*, **Interoperability prototype between hospitals and general practitioners in Switzerland,** in: Medinfo 2010, Cape Town, South Africa, pages 366-370, IOS press, 2010.

[5] *Bruno Alves, Michael Schumacher, David Godel, Philippe Richard, Abu Khaled Omar* and *Henning Müller*, **Prototype d'interopérabilité entre hôpitaux et médecins traitants**, in: Actes de GISEH 2010, conférence francophone "Gestion et Ingénierie des Systèmes Hospitaliers", Clermont-Ferrand, France, 2010.

[6] *Henning Müller, Michael Schumacher, David Godel, Abu Khaled Omar, Francois Mooser* and *Sandrine Ding*, **MediCoordination: A practical approach to interoperability in the Swiss health system**, in: The Medical Informatics Europe Conference (MIE 2009), 2009.

[7] *Jörg Caumanns, Raik Kuhlisch, Oliver Pfaff* and *Olaf Rode*, **IHE IT-Infrastructure (ITI) White Paper: Access Control**, IHE International, 2009.

[8] *Joshua Painter, Alean Kirnak* and *John Moehrke*, **IHE IT-Infrastructure (ITI) White Paper: A Service-Oriented Architecture (SOA) View of the IHE Profiles**, IHE International, 2009.

[9] *Stefano Bromuri, Michael Schumacher, Kostas Stathis* and *Juan Ruiz*, **Monitoring Gestational Diabetes Mellitus with Cognitive Agents and Agent Environments**, in: Proceedings of the 2011th IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2011), 2011

[10] Stefano *Bromuri, Michael Schumacher* and *Kostas Stathis*, **Pervasive Healthcare using Self-Healing Agent Environments**, in: 9th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS'11), Springer Verlag, 2011

[11] *Stefano Bromuri, Visara Urovi, P. Contreras* and *Kostas Stathis*, **Situating Cognitive Agents In GOLEM**, in: EEMMAS, pages 115-134, Springer, 2007