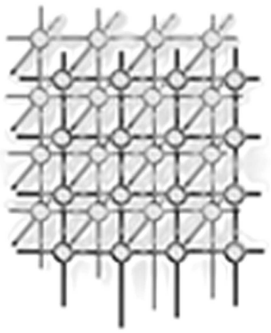
CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2009; **00**:1–7 *Prepared using* cpeauth.cls *[Version: 2002/09/19 v2.02]*



eMOLST: a Documentation Flow for Distributed Health Informatics

Gregor von Laszewski[†], Jai Dayal[‡], Lizhe Wang[†]

† *Pervasive Technology Institute, Indiana University at Bloomington Bloomington, IN 47404*
‡ *Service Oriented Cyberinfrastructure Laboratory, Rochester Institute of Technology Rochester, NY 14623*

SUMMARY

Electronic Health Records (EHRs) have many potential advantages over traditional paper records, such as wide scale access, error checking, and protection from physical damage to a record. As with any medical record, paper or electronic, both the patient's privacy and the document's integrity must be guaranteed. With initiatives such as Integrating the Healthcare Enterprise (IHE), computerized healthcare systems are able to share EHRs on a large scale, while protecting the patient's privacy rights. However, IHE does not yet meet the needs for all healthcare systems, as we will show with the eMOLST project.

The eMOLST project delivers software in support of Medical Order for Life Sustaining Treatment (MOLST) forms and uses IHE specifications for cross enterprise document storage and sharing, patient identification, and user authentication & authorization. The Web based system provides secure access to electronic MOLST documents regardless of the patient's or healthcare provider's location. The eMOLST project allows a user to have Single Sign On (SSO) access to the system from either the user's associated enterprise, or through a Web portal shared amongst all users across all enterprises. In this paper, we show a security solution to allow SSO from multiple access points for IHE compliant systems.

KEY WORDS: *Workflow, Health informatics, distributed system*

1. INTRODUCTION

A Medical Order for Life Sustaining Treatment (MOLST) form allows a patient to make end of life decisions, and is officially supported by the New York State Department of Health

*Correspondence to: 719 East 10th Street, Bloomington, Indiana 47408, U.S.
†E-mail: Lizhe.Wang@gmail.com

Received xx xxx 2009
Revised xx xxx 2009

(NYS DOH) [1]. The MOLST form goes beyond Do No Resuscitate (DNR) orders by specifying additional instructions for medical personnel, in case a patient can no longer make decisions for his/her self. For example, the MOLST form allows a patient to specify what comfort measures should take place, if feeding tubes should be employed, and whether or not to administer anti-biotics or other medications.

The NYS DOH has passed a number of initiatives to computerize medical records. As a result, the eMOLST project has been created to facilitate the large-scale sharing of computerized MOLST forms, and to make these forms available to patients and medical personnel regardless of their locations.

eMOLST is a distributed system that allows multiple healthcare providers to create and share electronic MOLST forms in a secure and scalable environment. In traditional healthcare systems, a user logs into the enterprise with which the user is associated. This can be problematic as physicians often change locations and would thus need credentials for each enterprise's system. Patients are even more likely to switch locations, so it is not feasible to require the patient to have so many different usernames and passwords. Currently, IHE allows credentials to be shared across enterprises for users who are authenticated at an enterprise's system. The eMOLST project extends this by allowing a user to have Single Sign On (SSO) access the system via multiple access points, either the user's associated enterprise, or through a shared Web portal. In order to provide such functionality, we have implemented a process to delegate authentication to the appropriate identity provider.

To support the secure creation and distribution of electronic MOLST documents, eMOLST uses Service Oriented Architectures (SOAs), Federated Identity Management (FIM) and Integrating Healthcare Enterprises (IHE) guidelines to provide location independent access to a patient's MOLST form, while retaining the patient's privacy and the data integrity of the MOLST form.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 provides background information on MOLST documents. Section 4 discusses Federated Identity and Section 5 discusses the IHE initiative in relation to the eMOLST project. Section 6 discusses eMOLST's federated design, Section 7 contains future work, and Section 8 has our conclusions.

2. Related Work

Leveraging IHE to facilitate computerized health records is not a new topic, and many companies and researchers have developed a number of software packages and products. [18] discusses their implementation experience with IHE guidelines and have even extended IHE to allow a patient to define his/her privacy requirements. This research only considers access to the system at the enterprise level, and does not discuss the authentication delegation to allow access via a Web portal.

[11] discusses enabling EHR workflows through a Web based system, but this research does not leverage off of IHE guidelines. Additionally, this research does not implement any distributed storage mechanisms for EHRs as all EHRs are stored in a central repository.

[10] implements a similar architecture to the eMOLST project, in that it implements distributed XDS Repositories and allows access to the system via a Web portal. The research only discusses authentication to the system from the centralized Web portal, containing a single user directory, resulting in a more centralized authentication approach.

3. MOLST Overview

To fully understand the eMOLST project, we must first understand the MOLST document and the document's workflow. A workflow [4, 19, 3] is concerned with the automation of procedures where documents, information or tasks are passed between participants according to a defined set of rules to achieve, or contribute to, an overall business goal. Once we have an understanding of the MOLST document, we then provide a set of technical requirements. With the set of requirements, and the subsequent discussion on IHE, we can see how IHE does not fulfill all of eMOLST's needs.

3.1. MOLST Documents

MOLST is a form that is used to make end of life decisions and goes beyond DNR documents. This form helps specify exactly what will happen to you should you no longer be able to make decisions for yourself. In addition to standard DNR forms, MOLST specifies additional parameters, such as:

- What comfort measures should take place.
- Whether or not to administer feeding tubes.
- Whether or not the patient wishes to be placed on antibiotics, if needed.

MOLST also has supplemental forms, which further specify a patient's treatment.

A MOLST document also has different sub-forms for minors (individuals under 18), and also allows a patient guardian to sign the form for a patient [2]. MOLST forms are not considered valid until both the patient (or legal guardian) and physician sign the form. Electronically, these documents need to be maintained and stored, but should not be searchable by physicians other than the designated physician. Traditionally, the paper MOLST documents reside with the patient physically, and any changes either require the document to be faxed to the primary physician, or require the patient to change physicians. With the eMOLST project, a patient might wish to have a physical copy of the document also, so many copies of the document may distributed at any time [1].

Several different individuals may participate in filling out a MOLST form, as the process often spans several days. Any individual who makes a change to the MOLST form is required to add the changes to the revision history. Each sub-form in the MOLST document requires a set of signatures, including the patient or guardian, and the physician; the sub-form is not considered valid until the appropriate signatures are obtained.

In real life situations, many different care givers will need to view the document. For example, EMT staffers will need to see the patient's MOLST form, but do not have any authorization

to modify the form, where as a nurse in a hospital can update the form, but can not finalize the form with a signature.

For a more detailed discussion on MOLST documents, such as the various options, parameters, and the entire workflow, see [1, 2, 14].

3.2. eMOLST Functional Requirements

From the above discussion on MOLST documents, we get a sense of functionality needed to securely computerizing MOLST documents. We only formulate general functional requirements in this section, and not how these requirements are implemented. The implementation is discussed in Section VI.

The eMOLST system will need the following requirements:

1. the system must be able to uniquely identify patients
2. the system must be able to uniquely identify users
3. the system must be able to categorize these users into more general roles or groups
4. the decision to allow access to the documents, and to allow operations to be performed on the document, must consider both the individual user, and to which role the user belongs
5. the system must be able to uniquely identify the MOLST documents
6. a physician should be able to see all MOLST documents to which the physician is assigned
7. only the designated physician can see unsigned documents, while other users can only see MOLST documents that have been signed, unless the user is the designated physician
8. users and patients must be able to access the documents regardless of their location
9. all modifications to the documents must be logged

Storing MOLST documents can be attained in two ways. Each facility can maintain its own repository, but share these documents securely with other healthcare enterprises, or there can be a separate centralized system. Since most healthcare enterprises (hospitals, doctors offices, etc) maintain their own IT systems, it would be best to integrate eMOLST into these systems, and let each enterprise handle the security and the sharing of documents accordingly. To accomplish this, we must first understand Federated Identity Management (FIM), and Integrating the Healthcare Enterprise (IHE).

4. Federated Identity Management

To understand how IHE outlines how enterprises can exchange user identities and establish a trusted relationship, we must first understand Federated Identity Management (FIM). FIM is a set of technologies that allow various computer systems to unify identity information across the systems' security domains [12]. FIM provides Web base systems with cross domain SSO capability, meaning the user signs in once and has appropriate access to the necessary resources which may lie in different security domains.

The FIM model has four logical components [12]:

1. User, or person who assumes an identity.
2. User agent, or the user's interface such as a Web browser.
3. Service provider, which is the service the user is trying to access.
4. Identity provider, which is responsible for authenticating and providing users with an identity.

There are several popular frameworks available for FIM, but the most common is the Security Assertion Markup Language (SAML). SAML is an OASIS standard and uses XML to organize and format identity information. SAML is composed of *assertions*, which are XML packets that contain the identity holder's identifier, authentication status, and attributes [12]. SAML is a flexible tool, as it can be transmitted via SOAP messages, and is often found in other FIM frameworks, such as WS-Federation and WS-Trust. Additionally, SAML is commonly used to identify non-human users such as other Web services or applications.

The eMOLST project uses SAML to transfer identities across security domains. In the case of sharing identity across federations, eMOLST will have to rely further on the WS-Interoperability protocol stack, with WS-Trust and WS-Federation in particular. eMOLST currently operates as one federation.

OpenID was created to be a lightweight and scalable framework to share identities [12]. OpenID leaves authorization to the service provider, and only serves as a means of authentication. The discussion on how OpenID handles authentication can be found in our Web Portal Authentication section.

While eMOLST does not use any OpenID Identity Providers explicitly, the project extends upon IHE's SSO by following OpenID's method of delegating authentication to a remote identity provider. In this case, the provider is the user's healthcare enterprise's IT system.

5. Integrating the Healthcare Enterprise

Integrating the Healthcare Enterprise (IHE) is an initiative designed to improve the interoperability between healthcare IT systems. IHE is not a set of standards, but rather a set of implementation guidelines that profile acceptable standards for healthcare IT systems [21]. For many different types of actors (users, applications, etc) and transactions (operations between actors) common to real life healthcare use cases, IHE has formulated these set of guidelines into integration profiles [10].

According to [7], integration profiles can be broken down into three categories: content, workflow, and infrastructure. Content profiles define how enterprises will manage (create, share, and store) specific content objects. Workflow profiles define each step in creation of the document, i.e., at each state in the workflow. Infrastructure profiles address basic authentication and authorization issues [7]. The eMOLST project makes use of several integration profiles, so we will discuss the IHE profiles related to the eMOLST project as a discussion of all IHE integration profiles is out of the scope of this paper.

As discussed above, the eMOLST system needs a way to identify both patient's and IT system users across healthcare enterprises. Not only does eMOLST need to know the identity of these users, but eMOLST must also know what operations the user can perform and what

documents the user can view, i.e., the user's authorization across enterprises. Additionally, each participating enterprise must know how to maintain audit trails for the accessed documents. Each healthcare enterprise must also agree on how the MOLST forms will be formatted, stored, and shared.

The Cross Enterprise Document Sharing (XDS) profile describes how the participating enterprises will manage these shared electronic MOLST documents. The Patient Identifier Cross-Referencing (PIX), Enterprise User Authentication (EUA), Cross Enterprise User Assertion (XUA), and Audit Trail and Node Authentication (ATNA) integration profiles are defined to specifically address the identity, security, and auditing issues mentioned above.

5.1. Cross Enterprise Document Sharing

XDS outlines how to share access to EHRs between healthcare enterprises. A group of enterprises that agree to share EHRs form an XDS Affinity Domain. Members of an XDS Affinity Domain must first agree on several parameters, such as how to uniquely identify patients and users, the structure and format of the documents, and access control [21].

To enable cross enterprise document sharing, XDS requires federated document repositories and a document registry. The federated document repositories simply store the documents and respond to document retrieval requests. The document registry maintains metadata about these documents so that each enterprise in the domain can locate these documents. It should also be noted that the XDS Integration Profile does not outline mechanisms to provide access to portions of a document, but only to the document as a whole. XDS does not restrictions the types of documents; they can be text-based documents, such as MOLST, or images such as X-RAY or MRI images [9].

eMOLST uses XDS to facilitate the sharing of documents across enterprises. eMOLST assumes that each enterprise maintains a document repository and all that all enterprises share a document registry. It is possible, however, for multiple enterprises to share the same repository, but this feature is not currently implemented in eMOLST. Also, eMOLST must be able to control access to specific portions of the document, but XDS only considers the document as a whole. It is possible to implement such controls with the OASIS Extensible Access Control Markup Language (XACML) [18], but eMOLST currently does this at the application layer.

5.2. Patient Identifier Cross-Referencing

A Patient Identifier Domain (PID) is a group of enterprises that agree to share a common patient identity scheme. This domain could consist of multiple enterprises, or it could be multiple departments within a single enterprise. Within a PID, a Patient Identity Source system is responsible for assigning each patient with a unique identifier, and for maintaining a set of attributes associated with the patient. In some cases, it may be necessary to assign multiple unique identifiers to a patient, but this is not needed for the eMOLST project [9].

PIX extends on the PID model by allowing patient identities to be shared across PIDs by creating a Patient Identifier Cross-reference Domain. A Patient Identifier Cross-reference Manager maintains a list of each individual PID, and assigns each PID with it's own unique

identifier. The Patient Identifier Cross-reference Manager does not guarantee, or improve, the quality and accuracy of the individual PID identifiers. The manager simply provides a way for domains to locate patients between each other. Thus, each PID is responsible for the quality and accuracy of its associated patient identifiers [9].

The eMOLST project does not make use of the PIX integration profile at this time, and instead relies on a single PID for the entire system. In the future, if eMOLST is extended to a national scale, PIX should be implemented.

5.3. Enterprise User Authentication

A medical staff user commonly operates on several compute devices, such as PDAs, laptops, and desktops, to perform daily tasks. The user will also commonly need to access several different departmental components of the IT system, for example the radiology or billing sub-systems. EUA allows an enterprise to establish one identity scheme for the user thus enabling SSO through out the entire enterprise. IHE EUA uses Kerberos to establish an identity, and uses HL7 Clinical Context Object Workgroup (CCOW) to hide the underlying system's complexity from the user. For example, if an operation requires several system components, the user is kept unaware of all the underlying components [9].

eMOLST uses EUA for enterprise wide authentication, however EUA does not meet all of eMOLST's needs. For example, a user (perhaps a patient) needs to be able to log into the eMOLST system at home via a Web browser, thus operating outside any healthcare enterprise security domain. IHE does not provide an integration profile to delegate authentication from the Web portal to the user's associated enterprise (across security domains), but it is possible to implement something similar to OpenID [17]. In such a scenario, a user's identity will include both the user's associated enterprise and the user's unique identifier within that enterprise. When a user is accessing the system remotely, through a Web portal, the Web portal work similarly to OpenID by forwarding the authentication session to that enterprise's Kerberos authentication server. For a discussion on OpenID, see [17] and [15].

eMOLST only allows patients to access the system through the Web portal, thus patients will never access the healthcare enterprise directly. This scenario allows us to view the Web portal as a separate enterprise. This is made possible largely because eMOLST currently only has one PID for the entire system, so mapping the patient's Web portal username to the patient's unique identifier is trivial. The situation is a little different for physicians, and other medical staff, as their duties are directly related to specific healthcare enterprises. It is possible to provide a physician with two separate accounts, one for the associated enterprise, and one for the Web portal, but such a method is not scalable as physicians may hold positions at multiple enterprises. To handle medical staff authentication via the Web portal, eMOLST delegates the authentication process to the appropriate EUA. The authentication process details are further detailed in Section 6.

5.4. Cross Enterprise User Assertion

Based on SAML, XUA is a guideline for federated identity management when an entity wishes to perform operations that span across healthcare enterprises. SAML assertions contain

information about authenticated users, such as the role and enterprise to which a user belongs. XUA does not specify how enterprises in the security domain should authenticate and authorize users, it simply just outlines a way to transfer a user's identity information [18].

XUA Defines three actors in a cross-enterprise transaction: X-Service User, X-Service Provider, and X-Identity Provider. The X-Service User is the user attempting to access a resource in another enterprise. The X-Service Provider is the service the user is attempting to access, and the X-Identity Provider issues SAML assertions for authenticated users. The X-Identity Provider does not perform authentication, but rather assumes that the user is properly authenticated.

With the current implementation of eMOLST, each enterprise handles it's own user authentication and authorization to the MOLST documents. Also, the Affinity Domain (group of participating enterprises) for XUA contains the same set of enterprises as does the Affinity Domain for XDS. As patients frequently change locations, a physician often needs to access MOLST documents located at another enterprise. To provide this capability, the eMOLST project relies heavily on XUA. While it is possible to extend XUA to share identities across Affinity domains, eMOLST does not need this as currently operates as one Affinity Domain.

5.5. Audit Trail and Node Authentication

ATNA provides patient privacy, data integrity, and user accountability with in a secure domain, by specifying node level security and auditing parameters. A secure domain may span from individual system components, to multiple enterprises, but generally includes the enterprises in the XDS Affinity Domain. A node is considered to be any network addressable component of the system [9]. ATNA provides the following:

1. Describes the security environment (identification, authentication, and authorization) for a node.
2. Defines auditing requirements for a node.
3. Defines the communication security environment, for example if the node is using TLS.
4. Describes the communication of audit messages between the node and an Auditing Repository.
5. Defines additional configurations for a task specific nodes. The Radiology Audit Trail is an example.

eMOLST incorporates the MOLST document's history into the document itself and only audits information pertaining directly to the MOLST documents. ATNA is responsible for maintaining system usage logs, not necessarily document specific logs. In the future, the eMOLST project will require system logging capabilities using ATNA, but the eMOLST project does not currently use ATNA, and instead, uses its own application level document revision mechanism.

6. eMOLST Implementation

From the above discussions on MOLST, FIM, and IHE, we can see how IHE uses FIM, and other frameworks, to provide large-scale interoperability across healthcare enterprises. We can also see that IHE does not discuss how to allow for a user to have SSO access to eMOLST through multiple access points.

eMOLST needs a Web portal because physicians, and especially patients, commonly change locations and it is infeasible to require a patient to obtain a new user ID for each location. Also, we've decided to allow SSO access to eMOLST through the enterprise level IT systems because it is much more convenient for users, such as medical staffers, so sign into the system once, and be able to carry out their daily job functions. Without enterprise wide SSO, the medical staffer would need to authenticate with both the enterprise IT system, and with the eMOLST system through the Web portal.

In this section, we will discuss eMOLST's high-level design, eMOLST's access control, and then we discuss our SSO security solution.

6.1. eMOLST Architecture

Figure 1 shows eMOLST's high level design of the affinity domain and Web portal This image only depicts IHE related components.

The affinity domain consists of several independent enterprises, an XDS Registry, an XUA Service, and a Patient Identity Service. Each enterprise, whether a hospital or nursing home, contains an eMOLST instance composed the following:

- **EUA Service:** This represents the enterprise's authentication and authorization service. This also contains the director of all the enterprise's users, which is not shown in the diagram.
- **Enterprise Patient Registry:** In order to associate MOLST documents to patients, each enterprise must have a registry containing information about the patients. This information ranges anywhere from contact information and a social security number, to insurance and billing information. This service interacts with the eMOLST Patient Identity Service, which is discussed below.
- **eMOLST XDS Repository:** This repository contains stores the eMOLST documents. This repository is shared with the entire enterprise and can contain other EHRs, but for eMOLST, we only consider MOLST documents.
- **Departmental Services:** An enterprise typically consists of many other department services, for example the radiology department. This is only shown to more accurately represent a healthcare enterprise.

The eMOLST software contains many other services, for example, the service that actually retrieves documents, but these are excluded for simplicity. Users access the enterprise level system through a local Web portal. The Web server is configured to only allow connections from with in the enterprise, the shared Web portal, or from the other eMOLST services in the affinity domain. Thus, the Web portal is not exposed outside of the enterprise.

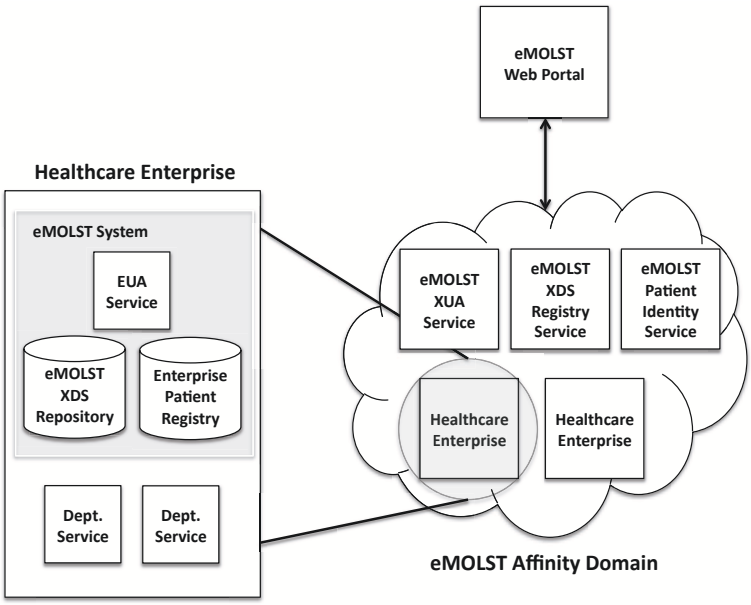


Figure 1. eMOLST Architecture

These following components are services shared by each enterprise in the affinity domain. These services may operate at any location; through a third-party vendor, a state run healthcare agency, or even at a designated high-capacity enterprise.

The **Patient Identity Service** is responsible for creating and transferring a patient's identity across enterprises. Before creating a unique identifier for a patient, the enterprise level patient ID service first queries the eMOLST Patient Identity Service. If this patient already exists in the eMOLST system, the patients existing unique identifier and necessary information, and forwarded back to the enterprise level system. If this patient does not exist, then the enterprise level service creates the new patient, stores the information, and then registers the new patient with the eMOLST Patient Identity Service. Upon registration, the eMOLST Patient Identity Service creates and returns a new unique patient identifier. This is made possible, because all enterprises in the affinity domain are also in the same PID, as discussed earlier. IHE integration profiles do not specify how to create new identities. The PIX integration profile only specifies how to share identities across PIDs.

The **eMOLST XUA Service** is responsible for transferring identities across enterprises. When a user attempts to access a MOLST document and the enterprise's eMOLST system determines that the document may be stored in another enterprise, the system obtains a SAML token for the user from the eMOLST XUA Service. This SAML token contains information

about the user as well as the user's role. In the current implementation, the eMOLST roles are the same for all enterprises.

eMOLST XDS Registries do not store documents, but maintain pointers to the correct XDS Repository. When a user attempts to retrieve a document in a remote repository, the eMOLST system must first obtain a SAML token for the user from the eMOLST XUA Service. The XDS Registries are not responsible for performing authorization. Authorization is handled by the service containing the XDS Repository. The XDS Registry will receive the user's SAML token, and will forward this along with the query request to the XDS Repository. Naturally, if the user is authorized, the Repository returns the document to the Registry, which then forwards the MOLST document to the user.

6.2. Users and Roles

eMOLST uses hierarchical Role Based Access Control (RBAC) to classify eMOLST users and permissions associated to the users. RBAC has the following basic components. A role classifies functional responsibilities users carry out within an enterprise, a user is either a person or process, and a permission is an access rule [13] [6]. More formally,

- $ROLES = \{r_1, r_2, ..., r_l\}$, the set of all roles, where l is the total number of roles;
- $USERS = \{u_1, u_2, ..., u_m\}$, the set of all users, where m is the total number of users;
- $PERMISSIONS = \{p_1, p_2, ...p_n\}$, the set of all possible access permissions where n is the total number of permissions;
- $UR \subseteq USERS \times ROLES$, the set of user-role relationships, i.e., assigning users to a roles;
- $PR \subseteq PERMISSIONS \times ROLES$, the set of permission-role relationships, i.e., associating permissions to roles;
- $RR \subseteq ROLES \times ROLES$, the set of role-role relationships, i.e., the role hierarchies. The senior, or parent, role has the permissions of each junior, or child, node.

Furthermore, a permission

$$p \in PERMISSIONS = \{identifier, mode, action, target, constraints, exception\}.$$

When a permission is assigned to a role, we get

$$pr \in PR = \{identifier, r, mode, action, target, constraints, exception\}.$$

The $identifier$ uniquely identifies the permission, while r represents to which role this permission applies. A mode is either an *obligation* or an *authorization*. An obligation represents the actions the user must perform, where as an authorization represents the actions a user is allowed to perform. A + represents an allowed permission, and a − represents a prohibited permission. The *action* is the operation while the $target$ is the data object to which the operation is applied. In this case, the $target$ is either signed or unsigned MOLST forms. $constraints$ set limitations on the permissions, for example, only after 5PM and *exceptions* specify exception cases in which the permission may be applied.

Table I lists some of the roles and permissions for the eMOLST project in regards to MOLST forms. For the action field, C = create, R = read, M = modify, and S = sign. For the target field,

Role	Permission
Physician	(phy1,a+,C,u-MOLST,,) (phy2,a+,R,s-MOLST,,) (phy3,a+,R,u-MOLST,form → physician,) (phy4,a+,M,s-MOLST,,) (phy5,a+,M,u-MOLST,form → physician,) (phy6,a+,S,s-MOLST, form → physician,) (phy7,a+,S,u-MOLST, form → physician,)
Patient	(pat1,a+,R,s-MOLST,form → patient,) (pat2,a+,R,u-MOLST,form → patient,)
Physician's Assistant	(pha1,a+,C,u-MOLST,,) (pha2,a+,R,s-MOLST,,) (pha3,a+,M,s-MOLST,,)
EMT	(emt1,a+,R,s-MOLST,,)

Table I. eMOLST Roles & Permissions

u-MOST = unsigned MOLST form and s-MOLST = signed MOLST form. For the constraints field, form → physician means only if the physician is the form's designated physician. Likewise, form → patient means only the patient's form. It is possible to explicitly prohibit operations, such as (pat4,a-,S,u-MOLST,,), which means a physician's assistant cannot sign an unsigned MOLST form, but this is not needed as we only specify the actions permitted; all other actions are inherently prohibited.

6.3. Security Solution

The portal contains a set of interfaces to interact with the enterprise level EUA Services, the eMOLST XUA Service, and the eMOLST XDS Registry. The Web portal also maintains its own EUA Service, which contains a registry for all patients who have created accounts.

Current IHE integration profiles do not provide a discussion on how to allow a user to access the system through multiple points; they only detail how handle authentication at the enterprise system. To handle this, eMOLST has implemented a federated authentication mechanism by extending upon EUA. The Web portal has its own EUA and user identity registry for patients, but when a medical staffer attempts to login, eMOLST will forward the authentication session to the user's enterprise's EUA service, which is the identity provider. The process is as follows [17, 5]:

1. User, behind a Web browser, initiates authentication with Web portal, providing his/her user name and associated enterprise;
2. Web portal and enterprise EUA establish contact and agree to use numbers p and g, where $p \in \mathbb{P}$ and $g \in \mathbb{N}$;
3. Web portal selects a random number a, where $a \in \mathbb{N}$;
4. Web portal sends the enterprise EUA A, where $A = (g^a \ mod \ p)$;

5. Enterprise EUA selects a random number b, where $b \in \mathbb{N}$;
6. Enterprise EUA sends the Web portal B, where $B = (g^b \ mod \ p)$;
7. Web portal computes k, where $k = (B^a \ mod \ p)$;
8. Enterprise EUA computes k, where $k = (A^b \ mod \ p)$;
9. Web portal redirects user's browser to the enterprise's EUA service to perform the authentication;
10. User provides password to the enterprise's EUA service;
11. The enterprise's EUA asks the user for permission to forward the user's identity to the Web portal;
12. If the user does not allow, then the session ends;
13. If the user does allow, the enterprise EUA uses k to encrypt the identity information;
14. The Web portal then uses k to decrypt and obtain the user's identity information.

The most important benefit from this scheme is that the Web portal does not have to conform to any enterprises authentication mechanism. Each enterprise is fully capable of having its own authentication mechanism. For example, some enterprises might require two-factor authentication using a username/password and a time based key.

When an authenticated user needs to perform cross-enterprise transactions, the Web portal, which holds the user's identity, will obtain a SAML token from the XUA service, and will forward this SAML token to the necessary enterprises as needed. Thus, in this case, the Web portal appears to be no different than a standard enterprise.

Naturally, the Web portal and enterprise must have some level of established trust, otherwise, a malicious user could provide the Web portal with a fake enterprise, which would thus fraudulently authenticate the malicious user. Much work has been done to facilitate the establishment of trust, [16], [8], and [20], but the eMOLST system is configured to only communicate with a set of enterprises. In the future, eMOLST should implement a more robust and scalable way to establish trust.

7. Future Work

eMOLST leverages off of several of IHE's integration profiles, but to be able to operate within existing IHE systems, further work must be done. Firstly, eMOLST must be extended to use the PIX integration profile, as it is uncommon for enterprises to lie in the same PID.

Transaction logging and auditing is extremely important in the healthcare industry. Auditing and logging is often preferred over more strict security mechanisms since in emergency situations, a patient's survival is more important the the patient's privacy. Currently, eMOLST does not follow IHE's ATNA auditing profile, but instead handles this at the application layer.

Currently, eMOLST operates within a single affinity domain. In real life scenarios, eMOLST should be able to operate across affinity domains as it is unlikely that a nationwide affinity domain will be feasible.

8. Conclusions

IHE integration profiles provide implementation guidelines for many of the functions needed to computerize EHRs in a distributed environment. The XDS concept is straightforward and gives healthcare providers a great deal of flexibility for the technologies used, and the types of documents the provider wishes to share. Likewise, XUA outlines how to allow access to these documents when healthcare enterprises do not share the same security domain.

While IHE's federated identity management guidelines handle cross-enterprise transactions, they do not discuss the distributed delegation of authentication through a centralized Web portal. As patients often change locations and need access to MOLST documents, a centralized Web portal allows them to access the system, without needing security credentials at multiple enterprises. By adding functionality at the application layer, we show how it is possible to provide users with the appropriate access to the system, regardless of the user's or healthcare provider's location.

REFERENCES

1. Patricia Bomba, "Excellus BlueCross", and "New York State Department of Health". Molst - medical orders for life-sustaining treatment. Health Form, August 2008.
2. Patricia Bomba, "Excellus BlueCross", and "New York State Department of Health". Molst - supplemental documentation form for adults. Health Form, August 2008.
3. J. Chen and Y. Yang. Activity completion duration based checkpoint selection for dynamic verification of temporal constraints in grid workflow systems. *International Journal of High Performance Computing Applications*, 2009.
4. J. Chen and Y. Yang. Temporal dependency based checkpoint selection for dagoynamic verification of temporal constraints in scientific workflow systems. *ACM Transactions on Software Engineering and Methodology*, 2009.
5. Yannick Chevalier, Ralf Küsters, Michaël Rusinowitch, and Mathieu Turuani. Complexity results for security protocols with diffie-hellman exponentiation and commuting public key encryption. *ACM Trans. Comput. Logic*, 9(4):1–52, 2008.
6. Alessandro Colantonio, Roberto Di Pietro, Alberto Ocello, and Nino Vincenzo Verde. A formal framework to elicit roles with business meaning in rbac systems. In *SACMAT '09: Proceedings of the 14th ACM symposium on Access control models and technologies*, pages 85–94, New York, NY, USA, 2009. ACM.
7. Asuman Dogac, Veli Bicer, and Alper Okcan. Collaborative business process support in ihe xds through ebxml business processes. In *ICDE*, page 91, 2006.
8. Evren Eryilmaz, Mitch Cochran, and Sumonta Kasemvilas. Establishing trust management in an open source collaborative information repository: An emergency response information system case study. In *HICSS*, pages 1–10, 2009.
9. IHE International. Ihe it infrastructure (iti) technical framework. Technical report, IHE International, December 2008.
10. Basel Katt, Ruth Breu, Michael Hafner, Thomas Schabetsberger, Richard Mair, and Florian Wozak. Privacy and access control for ihe-based systems. In *eHealth*, pages 145–153, 2008.
11. Anant R. Koppar and V. Sridhar. A workflow solution for electronic health records to improve healthcare delivery efficiency in rural india. In *eTELEMED*, pages 227–232, 2009.
12. Eve Maler and Drummond Reed. The venn of identity: Options and issues in federated identity management. *IEEE Security & Privacy*, 6(2):16–23, 2008.
13. SangYeob Na and SuhHyun Cheon. Role delegation in role-based access control. In *RBAC '00: Proceedings of the fifth ACM workshop on Role-based access control*, pages 39–44, New York, NY, USA, 2000. ACM.
14. New York State Department of Health. Medical orders for life sustaining treatment (molst). Website.
15. Hyun-Kyung Oh and Seung-Hun Jin. The security limitations of sso in openid. In *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on*, volume 3, pages 1608–1611, Feb. 2008.

16. Arun Panayappan, Sriram Anand, and Raghu P. Pushpakath. Intelligent agents as trust negotiators for federated security. In *ICWS*, pages 1183–1184, 2007.
17. David Recordon and Drummond Reed. Openid 2.0: a platform for user-centric identity management. In *Digital Identity Management*, pages 11–16, 2006.
18. Asuman Dogac Tuncay Namli. Implementation experiences on ihe xua and bppc. Technical report, Software Research and Development Center, Middle East Technical University, Ankara, Turkey, December 2006.
19. M. Wang, R. Kotagiri, and J. Chen. Trust-based robust scheduling and runtime adaptation of scientific workflow. *Concurrency and Computation: Practice and Experience*, 2009.
20. Alfred C. Weaver and Zhengping Wu. Using web service enhancements to establish trust relationships with privacy protection (extended and invited from icws 2006 with id 47). *Int. J. Web Service Res.*, 6(1):49–68, 2009.
21. Kim Wuyts, Riccardo Scandariato, Geert Claeys, and Wouter Joosen. Hardening xds-based architectures. In *ARES*, pages 18–25, 2008.