MITRE

Healthcare TrustHub: Capabilities and Web Services for Physician Identity Management

I3P Safeguarding Digital Identity Research Activity

Bruce Bakis
Maria Casipe
Jaya Tripathi
David Weitzel

May 2010

This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001, under the auspices of the Institute for Information Infrastructure Protection (I3P) research program. The I3P is managed by Dartmouth College. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security, the I3P, or Dartmouth College.

This page intentionally left blank

Approved By:

Bruce J. Bakis, Principal Investigator May 6, 2010

Name and Title of Approval Signature Date

Abstract

This document provides the design, technical architecture and implementation approach of a proof-of-concept Healthcare TrustHub that was prototyped by MITRE under an identity management research activity—Safeguarding Digital Identity—managed by the Institute for Information Infrastructure Protection (I3P).

The TrustHub is a set of open-source, standards-based capabilities and web services that demonstrate how a web services architecture solves the problem of physician/care provider identity interoperability and portability in a healthcare federation that is readily extensible, secure and preserves privacy.

Table of Contents

List of Figures

List of Tables

This page intentionally left blank

1 Introduction

1.1 Abstract

This document provides the design, technical architecture and implementation approach of a proof-of-concept Healthcare TrustHub that was prototyped by MITRE under an identity management research activity—Safeguarding Digital Identity—managed by the Institute for Information Infrastructure Protection (I3P).

The TrustHub is a set of open-source, standards-based capabilities and web services that demonstrate how a web services architecture solves the problem of physician/care provider identity interoperability and portability in a healthcare federation that is readily extensible, secure and preserves privacy.

1.2 Background

MITRE was engaged in a two-year digital identity management research activity (April 1, 2007 to July 31, 2009) that was managed by Dartmouth College's Institute for Information Infrastructure Protection (I3P) under a Department of Homeland Security, Science and Technology Directorate grant (2006-CS-001-000001).

MITRE was the Principal Investigator of the Safeguarding Digital Identity research collaboration with Cornell University, Georgia Institute of Technology, Purdue University, SRI International and the University of Illinois at Urbana-Champaign.

Each organization conducted research in accordance with this mission statement: *Research, analyze and prototype digital identity management (IdM) solutions that allow sharing of identity-related information across organizations in ways that are sufficiently accurate, cost-effective, secure and privacy respecting.*

The research directions taken by the member organizations are described on the I3P's website: http://www.thei3p.org.

MITRE conducted two threads of research:

Enabling Capabilities and Web Services for a Healthcare TrustHub.

A Trust Framework: The SPICI (Sharing Policy, Identity, and Control Information) Approach to Negotiating Identity Federation and Sharing Agreements.

1.3 Overview

The previous administration made a major policy effort to reduce the redundant information technology costs in the $2.7 trillion healthcare expenditure in the United States. Policy debates notwithstanding, the only technological response is to bring modern Web Services and Web 2.0 technologies to bear.

To help achieve some of the key healthcare-related goals in the Obama-Biden administration's technology agenda, care providing organizations are expected to increasingly enter into collaborative networks (e.g., Regional Health Information Organizations, or RHIOs) of providers, pharmacies, insurers, etc., that need to share information securely. A major concern in these networks is the assured digital identification of all participating entities.

In order to help address these issues, goals and challenges, we have leveraged the U.S. Department of Health and Human Services (HHS) American Health Information Community's (AHIC) use cases[1] in the area of identity management utilizing emerging Web Services technologies. We have addressed the section "Provider Authentication and Authorization Workflow" from the 2007 AHIC Emergency Health Responder use case by creating a set of enabling prototypic capabilities and web services—the Healthcare TrustHub—that implement: Provider Identity Management Services, Provider Authentication Services and Provider Authorization Services.

We have incorporated existing industry standards such as Health Level Seven (HL7) and identity management projects such as Project Higgins to provide a set of prototypic capabilities and Web Services to address physician digital identity interoperability and messaging issues.

There are three main components of the prototypic capabilities and Web Services that we have developed. Each component corresponds to a different trust domain:

> Capabilities that support physician authentication (i.e., log in) to his/her local healthcare portal (Domain/Hospital A), patient lookup and information retrieval in that portal, and secure communication of information requests and retrievals over the Internet from the local portal to a remote healthcare portal managed by an unaffiliated healthcare provider (Domain/Hospital B).
>
> Capabilities and Web Services—collectively called the TrustHub—to support the management of physician digital identity data by a certifying authority. These include a Standard Physician Identity Model and a Put Physician Data Web Service and a Get Physician Data Web Service to interact with physician identity information.
>
> Capabilities and Web Services at a healthcare portal to service information requests made over the Internet from an unaffiliated healthcare portal. These include a Patient Record Web Service to manage and fulfill healthcare information requests made over the Internet, a capability to authorize access to patient information requests made via the Patient Record Web Service, and a Connector that provides the interface between WorldVistA[2] and the Put Physician Data Web Service.

1.4 Need and Goal

The AHIC 2007 Emergency Health Responder and 2008 Patient-Provider Secure Messaging use cases establish the need for a core set of interoperable healthcare services. Specifically, the

[1] http://www.hhs.gov/healthit/usecases/
[2] The open source Electronic Health Record (EHR) and Information System developed by the U.S. Department of Veterans Affairs (VA).

Provider Authentication and Authorization Workflow section of AHIC 2007 Emergency Health Responder use case establishes the need for:

Provider Identity Management Services

Provider Authentication Services

Provider Authorization Services

Our goal was to standardize physician identity content by providing a standard data model, leveraging and incorporating existing industry standards such as HL7 and Higgins, and to provide a set of enabling Web Services and demonstration capabilities to provide a proof-of-concept solution to interoperability and messaging problems that arise when physician identity information is exchanged in a heterogeneous environment.

1.5 Purpose and Scope of this Document

This document provides the design, technical architecture and implementation approach of a prototype of an identity management system that provides enabling Web Services for physician identity management. Other capabilities and Web Services are described, but not to the same level of detail as the enabling Put and Get Physician Data Web Services.

1.6 Relationship with other Documents

This document is a companion to *Healthcare TrustHub Demonstration Walkthrough, V1.2*, which provides a walkthrough of a healthcare use scenario that demonstrates the TrustHub.

2 Architectural Design

2.1 Scope and Overview

This section provides a description of the architecture of the TrustHub. Web Services are implemented as Java 2 Platform, Enterprise Edition (J2EE) web applications and employ the Model-View-Controller (MVC) pattern for better separation of the architectural layers. The prototype is a service-oriented architecture built from software components that can be reused, centrally accessed, versioned, and managed.

As a result of architectural layering, database and workflow implementations do not leak out into the application layer, and transactions are encapsulated behind individual business services. The business service interfaces are governed by domain and industry standards and communicate through a Java Application Programming Interface (API) with an XML-based interface to a Remote Procedure Call (RPC) (JAX-RPC). Additional services (some examples are provider-patient messaging or health maintenance organization (HMO) registration or licensing verification), can be readily added to this framework, including third party services that use the same domain language.

2.2 Business Components

There are three main business components in our model (illustrated in the figure below). The three business components correspond to three different trust domains. They would be implemented in different application servers in a real-world situation and would function independent of each other. We have developed our prototype system so that these three components may be deployed as either three separate web applications or as one application running on one application server.

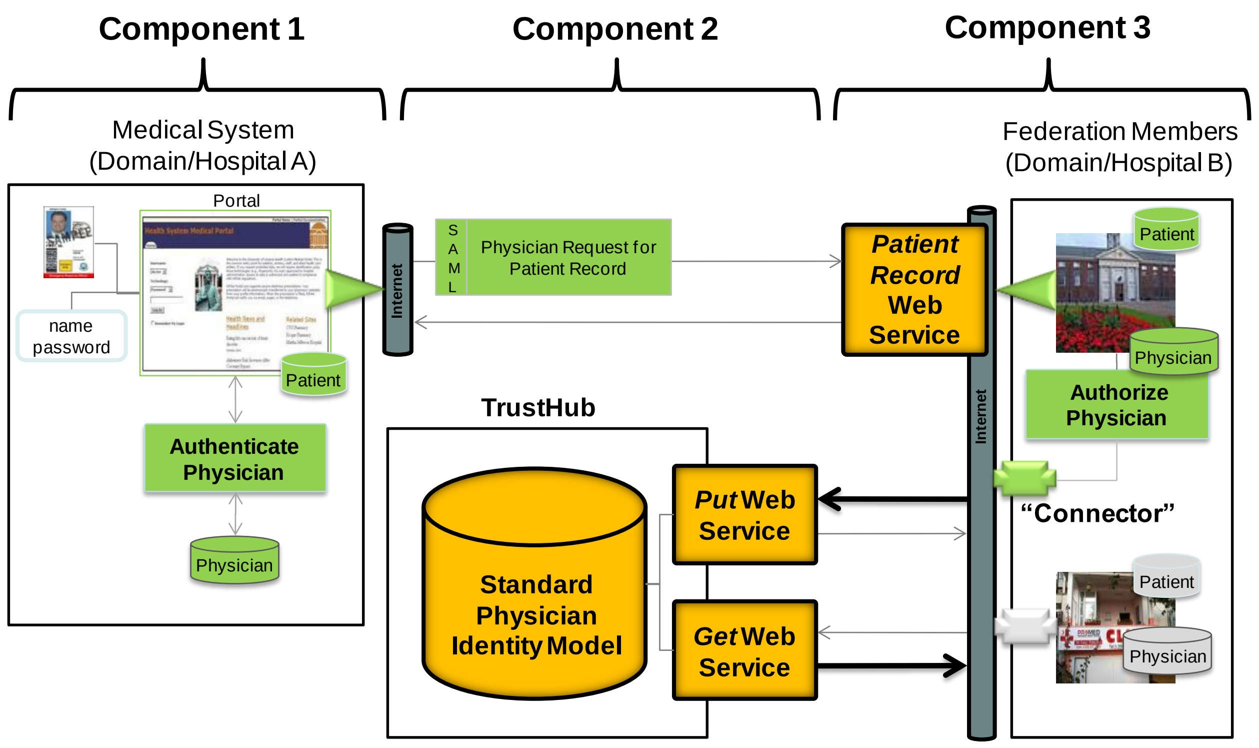


Figure 2-1 TrustHub Business Components

2.2.1 Component 1

This components provides capabilities that support physician authentication (i.e., log in) to his/her local healthcare portal (Domain/Hospital A), patient lookup and information retrieval in that portal, and secure communication of information requests and retrievals over the Internet from the local portal to a remote healthcare portal managed by an unaffiliated healthcare provider (Domain/Hospital B). The provided capabilities include:

A mock up of a healthcare portal.

A physician authentication (log in) capability that is supported by a small database of sample physicians (and their identifying information) who are authorized to access the healthcare portal.

A capability to authorize physicians to access patient information in the portal that is supported by a small database of sample patient information.

A capability to formulate a secure and certified request over the Internet using a Security Assertion Markup Language (SAML) assertion over a SOAP message to retrieve a patient medical record from an unaffiliated healthcare provider.

2.2.2 Component 2

This component provides services to support managing physician identity data by a central authority—the TrustHub. The provided services and capabilities include:

A Standard Physician Identity Model Database: This is a central repository of identity information for physicians that is managed and controlled by issuing certificates to participating third parties. This allows restricted access to the identity information and controls the release of attributes and assertions needed for the third parties to determine whether patient record information may be shared. The model is an extended version of Higgins Ontology Web Language (HOWL) and uses attributes taken from Health Level 7 (HL7), a standard language for sending information between healthcare systems. Only a few sample attributes are used in the prototype model: name, address, educational level, National Provider Identifier (NPI) number, and Social Security Number.

Web Services:

Put Physician Data Web Service (or a Put Web Service): This service allows any medical information system to a convert its representation of a physician's identity to a standard physician identity model and deposit (put) it into the Standard Physician Identity Model Database.

Get Physician Data Web Service (or a Get Web Service): This service allows any medical information system to a retrieve (get) a physician's identity from the Standard Physician Identity Model Database.

Any interested party who has been issued the appropriate level of trust via a certificate would be able to be a consumer of the Put and Get web services.

2.2.3 Component 3

This component consists of capabilities and services to support restricted access to patient medical record information maintained in Domain/Hospital B. This component constitutes the services in the domain where the requested patient health record information is maintained. In order to determine access privileges, this component would need to be a trusted participant in the TrustHub. The rules governing access rights to patient information are maintained by the institution that owns the patient record. The provided services and capabilities in this component include:

A Patient Record Web Service: This service merely provides an externalized definition of a patient record to support patient record search and retrieval among different healthcare providers.

A capability to authorize access to patient information requests made via the Patient Record Web Service

A Connector that provides the interface between WorldVistA and the Put Physician Data Web Service.

2.3 Technical Components

The section above provides a description of the business components and services and how they interact with each other. This section describes the application from a development point of view.

The figure below represents a high-level overview of the application layers. The business components described above form three layers the figure: View, Business Service Layer and Model.

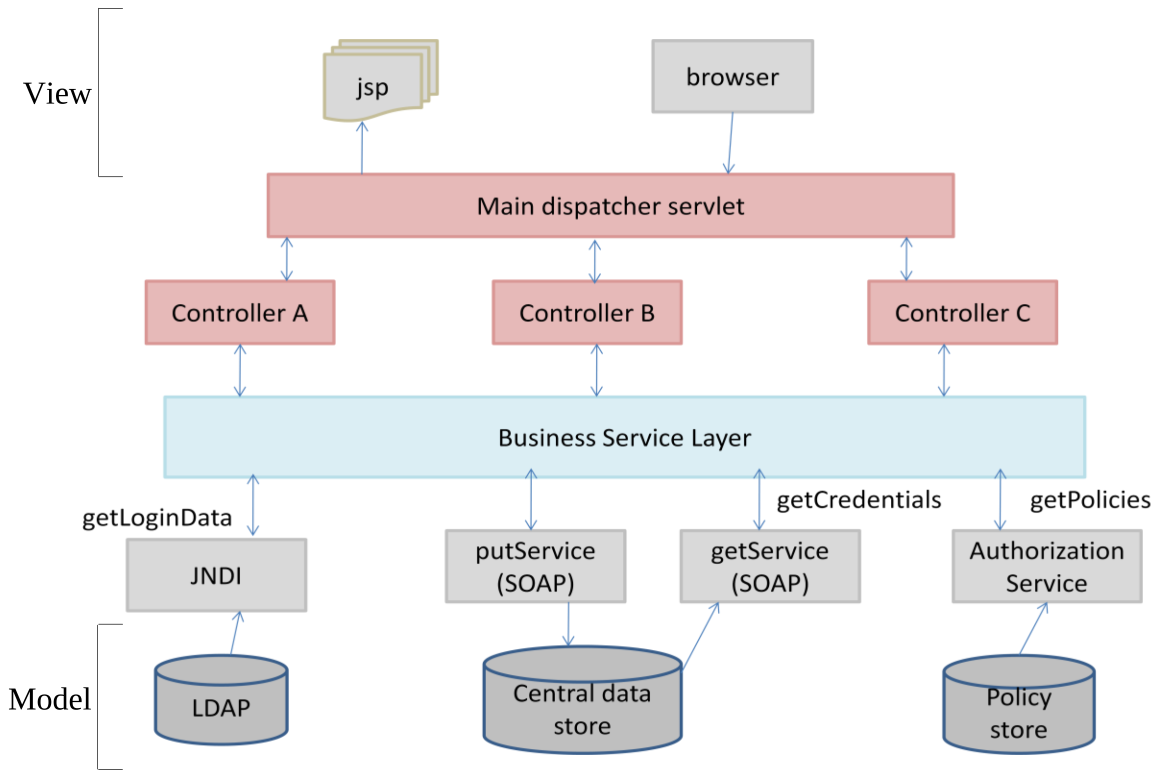


Figure 2-2 TrustHub Technical Components

2.3.1 View Layer

The look and feel of the View Layer is JSP-based and uses the Struts MVC framework with Tiles for presentation consistency and reusability. This provides the flexibility to alter the application look and feel to match portal-like or component-based requirements.

The View Layer is built using:

JSP v2.0 including the JavaSerever Pages Tag Library (JSTL) v1.1 tag library.

Struts Tiles 2 component for templating the (hospital's) portalized look.

2.3.2 Business Service Layer

All controllers interact with the interface definition of the Business Service Layer. Business services encapsulate and abstract the complexities of the underlying technical implementation. The interfaces are exported out as service descriptions in this layer, where they are exposed for use.

For example, getPhysicianService is the interface for physician lookup. This service provides a way to return a representation of the Physician Data Model as a Java Object.

2.3.3 Model Layer

Data for the various web parts are retrieved from the web services described above in section 2.2. The web service provider will return custom Java Objects as data and not as an XML representation of the model; therefore, schema information for that data as an XML Schema Language (XSD) is not needed.

The domain model consists of a set of types that represent the business objects within a particular business domain. The Physician Object is the key business object in the TrustHub. It is based on HL7. The HL7 Java SIG Project has created a Java application programming interface (API) for HL7,v3.0. From this, we built the physician object with the following data types:

> **Person:** A Person data type defines an object that describes an individual. It contains many attributes, including address, educational level, marital status, disability status, ethnic group, living arrangement, race, and religious affiliation. A Person data type has the following superinterfaces: BasicEntity, Cloneable, Entity, Hibernatable, InfrastructureRoot, LivingSubject, and RimObject. The Entity and LivingSubject types hold information about a Person. The other interfaces are for other uses of HL7. For our data model, the following are the most useful:
>
> > **Entity:** An Entity data type holds many attributes including: language, role, description, id, name, and role.
> >
> > **LivingSubject:** A LivingSubject data type holds many attributes including: gender, birth date, deceased time, and organ donor status.
>
> **LicensedEntity:** A LicensedEntity data type defines an object that describes a licensed individual. It contains an attribute for certification time. It has the following superinterfaces: BasicRole, Cloneable, Hibernatable, InfrastructureRoot, RimObject, and Role. The Role is most useful for our data model:
>
> > **Role:** A Role data type holds many attributes including address, participation, certification text, the kind of role, effective time, id, and name.

HL7 specifies useful data structures but is not intended to describe physician data alone. It has many other uses such as describing communication between institutions, patients, and other healthcare related functionalities. Because HL7 is not intended to describe physicians, we built our physician data object using the Higgins data model and incorporated Person and

LicensedEntity data structures described in HL7 as well as attributes taken from the WorldVistA "person" structure (listed below).

Table 2-1. WorldVistA Person Structure Attributes

Attribute	Description
name	Consists of the following: prefix, given first name, middle name, family last name, and suffix.
title	Category in which a person belongs to, e.g. physician, analyst, clinical coordinator, or nurse.
ssn	Social Security Number.
degree	Highest education earned.
dob	Date of birth.
division	Institution in which a person belongs to, e.g. VOE Office.
service_or_section	Type of service that a person practices (e.g. medicine) or section in which s/he belongs (e.g. pharmacy).
contact_info	Consists of the following: phone, office phone, commercial phone, voice pager, digital pager, fax number, and email address.
language	Language a person is fluent in.
class	A class in which a person belongs to, e.g. allopathic and osteopathic physicians, podiatric medicine and surgery service providers, chiropractic providers.
address	Consists of the following: street, city, state, zip code.
is_active_trainee	Indicates whether a person is a trainee or not.
vha_training_facility	Location in which a person is trained from.
training_date_start	Date in which a person started training.
last_training	Date (month and year) in which a person was last trained.
trainee_inactive_date	Date in which a person's training was deactivated.
program_of_study	The name of the program that a person is being trained for.
target_degree_level	The educational degree a person is trying to accomplish.

Most of the attributes from WorldVistA have the same name in HL7. For those attributes that do not have a direct mapping from WorldVistA to HL7, an "additional_info" element is added in our physician data model that holds a list of the name and value of these attributes. It will be up to an external application to process them. However, for our web services, the additional_info element will be ignored for simplicity.

2.3.4 Controller Layer

The main servlet, the DispatcherServlet and the various controllers are all implemented in Component 1 of the Medical System in Domain A. This component is built using the Struts MVC framework which includes the Central Dispatcher and Action Controllers.

The framework provides a clean division between the various (business) controllers, JavaBean models and views. It comprises a main servlet, the DispatcherServlet, which routes all incoming requests to the appropriate business controller or handler. The routing is based on the mapping defined in the application servlet.xml file. This file exists in the /WEB-INF directory. It is the application context definition for the Dispatcher Servlet (named "idmapp" in our identity management application) defined in the web.xml file.

There is one controller for each major action performed on the site. Currently there are no direct references to JSP pages from the controllers, (default index.jsp excluded) as they are protected in WEB-INF/jsp.

The controllers are all in the org.mitre.i3pidm.action package.

All the controllers interact with the interface definition of the services.

3 Technologies Employed

3.1 Server

Deployed to a J2EE1.4 container (JBoss v4.2.3 GA)

3.2 Software Components and Libraries

Runs on JDK 1.5

Ant 1.6 to build

Struts 2.1 framework

log4j-1.2.8 for logging

saaj-api-2005Q1

Database: MySql Community Server v5.0

WorldVista Open SemiViVa

3.3 Client Requirements

The application should support the following client configuration:

Operating system: Windows XP Pro or Windows 2000.

Browser: IE version 6.x, Firefox and other mozilla type browsers. Note: cookies cannot be disabled and Javascript need not be enabled.

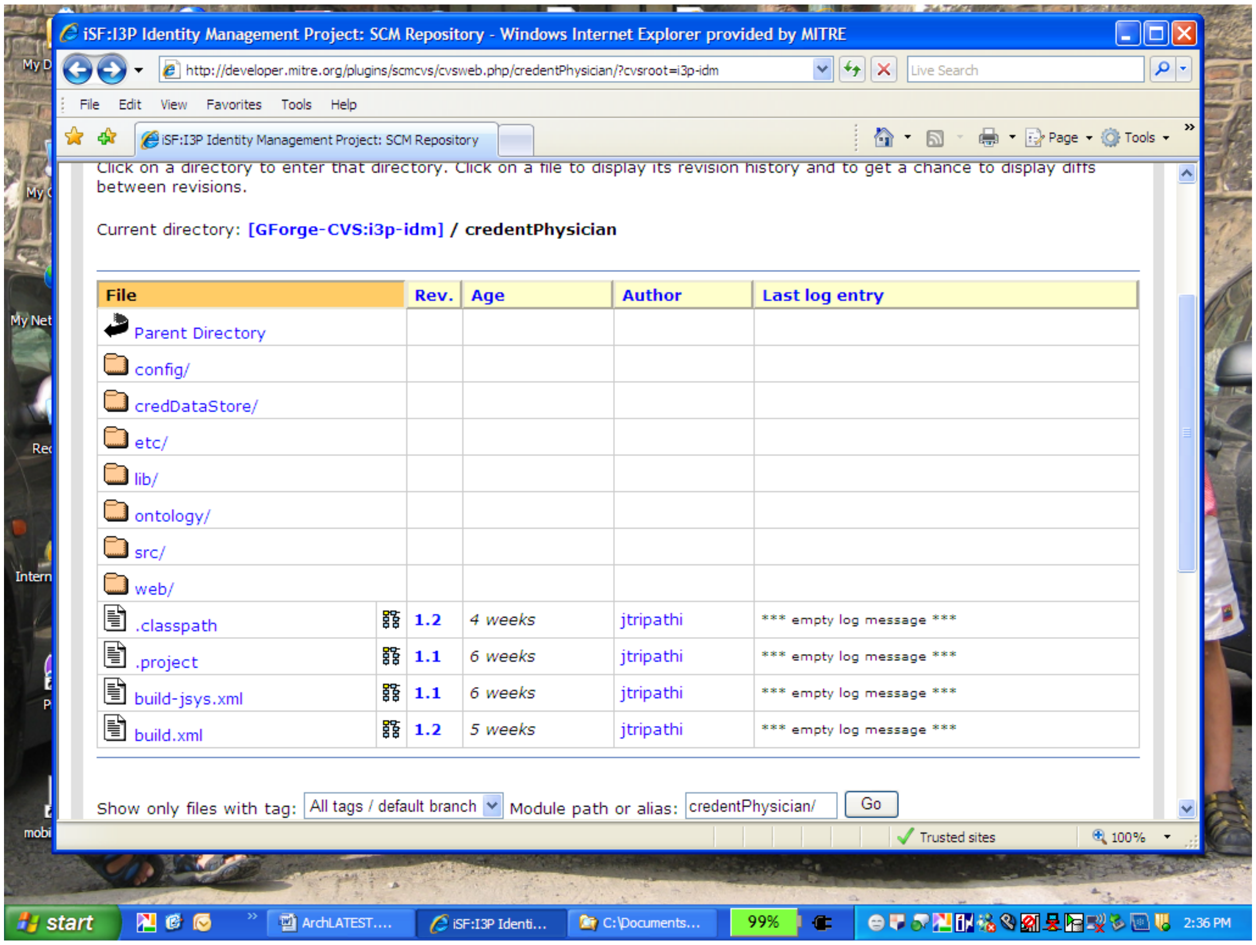
4 Project Standards and Practices

4.1 Coding

The development team established and followed a set of core Java and Web Services development standards that were based on industry-wide accepted practices, as described in *Effective Java - Programming Language Guide*, Joshua Bloch and *Patterns of Enterprise Application Architectures*, Martin Fowler.

4.2 Directory Structure

The figure below is a screen snapshot of the TrustHub directory structure. The TrustHub is deployed as three war files, which can be built using the ant script in the root directory.


iSF:I3P Identity Management Project: SCM Repository - Windows Internet Explorer provided by MITRE
http://developer.mitre.org/plugins/scmcvs/cvsweb.php/credentPhysician/?cvsroot=i3p-idm
Live Search
File Edit View Favorites Tools Help
iSF:I3P Identity Management Project: SCM Repository
Page Tools
Trusted sites
100%
start
ArchLATEST....
iSF:I3P Identi...
C:\Documents...
2:36 PM

Click on a directory to enter that directory. Click on a file to display its revision history and to get a chance to display diffs between revisions.

Current directory: **[GForge-CVS:i3p-idm] / credentPhysician**

File	Rev.	Age	Author	Last log entry
Parent Directory				
config/				
credDataStore/				
etc/				
lib/				
ontology/				
src/				
web/				
.classpath	1.2	4 weeks	jtripathi	*** empty log message ***
.project	1.1	6 weeks	jtripathi	*** empty log message ***
build-jsys.xml	1.1	6 weeks	jtripathi	*** empty log message ***
build.xml	1.2	5 weeks	jtripathi	*** empty log message ***

Show only files with tag: All tags / default branch Module path or alias: credentPhysician/ Go

Figure 4-1 TrustHub Directory Structure