dhs_logo_sm

Minnesota Department of Human Services

Provider Screening

Pre-Design & Technical Specifications Document

Prepared By: Greg Anderson

Prepared For: MN Dept of Human Services

Version: 2.0

Last Updated: March 7, 2012

Table of Contents

[Provider Screening 1](#_Toc318817497)

[Pre-Design & Technical Specifications Document 1](#_Toc318817498)

[Document Change History 3](#_Toc318817499)

[Overview 4](#_Toc318817500)

[1.1 Purpose of this Document 4](#_Toc318817501)

[1.2 Acronyms 4](#_Toc318817502)

[The Provider Screening Project And DHS General Technical Architecture 6](#_Toc318817503)

[1.4 Introduction 6](#_Toc318817504)

[1.5 DHS Development approach 6](#_Toc318817505)

[Technical Architecture REQUIREMENTS 8](#_Toc318817506)

[1.6 General Technical Architecture 8](#_Toc318817507)

[1.7 Summary 10](#_Toc318817508)

[APPENDIX A – DHS Architecture 11](#_Toc318817509)

[APPENDIX B – MITA Architecture 12](#_Toc318817510)

[APPENDIX C – DHS SOFTWARE STANDARDS LIST 16](#_Toc318817511)

[APPENDIX D – DHS Security Architecture 17](#_Toc318817512)

# 

# Document Change History

|  |  |  |  |
| --- | --- | --- | --- |
| **Release** | **Date** | **Person** | **Change** |
| 1.0 | March 6, 2012 | Greg Anderson | Incomplete draft |
| 2.0 | March 7, 2012 | Greg Anderson | Submitted version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Overview

## Purpose of this Document

This document describes the main aspects of the technical architecture for the Minnesota Department of Human Services, Health Care Systems (Medicaid) in preparation for the Provider Screening Application Project.

## Acronyms

Here are several commonly-used acronyms in this document.

| **Term** | **Meaning** | **Notes** |
| --- | --- | --- |
| AOP | Aspect-Oriented Programming |  |
| BRMS | Business Rule Management System |  |
| CMS | Center for Medicaid Services under the federal Department of Health and Human Services |  |
| MN-DHS | Minnesota Department of Human Services |  |
| EDMS | Enterprise Document Management System |  |
| ESB | Enterprise Service Bus |  |
| iLOG | Rules Engine Technology by IBM |  |
| JEE | Java Enterprise Edition, formerly J2EE |  |
| MITA | Medicaid Information Technology Architecture | As developed by CMS to provide enterprise architectural guidelines for state medicaids, currently mersion 3.0 is in process of being adopted |
| MN-ITS | DHS Health Care Provider Portal |  |
| NPI | National Provider Identifier |  |
| ORM | Object-Relational Mapping |  |
| OSS | Open Source Software |  |
| SIRS | Surveillance & Integrity Review |  |
| SCA | Service Component Architecture |  |
| SOAP | originally Simple Object Access Protocol, now just “SOAP” |  |
| SMO | Service Message Object |  |
| TCN | Transaction control number | Transaction tracking number for all claims |
| UDDI | Uniform Description, Discovery, and Integration | For service cataloguing and governance, particularly to facilitate dynamic client lookup of dependent service endpoints |
| UI | User Interface |  |
| WESB | IBM WebSphere ESB |  |
| WPS | IBM WebSphere Process Server |  |
| WS | Web Services |  |
| WSDL | Web Services Description Language |  |
| WSRR | IBM Web Services Registry and Repository | Similar to UDDI, but more tightly integrated with WPS/WESB. |
| XSD | XML Schema Document |  |

## 

# The Provider Screening Project And DHS General Technical Architecture

## Introduction

The proposed Provider Screening project consists of a web and workflow application that follows a path from a Provider enrollment application submission from a currently un-enrolled provider through the response of an unsuccessful or successful enrollment. There are three possible service calls that are escalated depending on the risk factor identified for the enrollee (in order of risk):

1. To make service calls to various external validation services validating the enrollee’s various statuses, e.g. licensure, exluded provider lists, etc.
2. To submit the enrollee to a SIRS review system.
3. To submit the enrollee to a background check system.

DHS architecture is based service oriented approach (see appendix A), with a decoupling of the web components (JSF 2.0 with Spring components) and the workflow components which specifically utilize an ESB (WESB) infrastructure with a workflow engine (WPS). In addition, a rules engine (iLOG) serves as the central repository of application rules that govern the overall processing. This architecture is aligned with the CMS MITA initiative (See Appendix B). See Appendix C for a complete list of system software utilized in the development of applications for DHS health care.

## DHS Development approach

Although there are some existing interfaces that this project requires, given that the Provider Screening project is a new project with little or no interfaces defined, the primary approach probably dictates a Top-down approach, along with some Meet-in-the-middle in certain cases.

In general DHS uses the following methodology within their SOA development:

* Determine the business requirements with well documented business rules.
* Identify the functional specifications as well as design and develop the user interface.
* Identify specific business rules and determine whether the application of rules engine technology is appropriate.
* The data model is determined from the derived business requirements, as well as the functional specifications.
* Model the workflow and develop the BPEL processes, both within the web component as well as within the overall workflow for WPS and WESB use.
* Identify and develop specifications for all of the services required to support the functional specifications.
* Identify and develop the interfaces required between each layer.
* Identify the error conditions and resulting workflows.

There are also global tasks that directly affect the development cycle as well:

* Design the security model from the requirements and design the technical implementation within the layers.
* Develop and normalize the data model from the functional specifications and business domain model.

The following describes how development is usually completed:

* Once the use case documents and the functional specifications are completed, the user interface tasks are developed. It should be noted that screens (UI) is usually de-coupled from the other layers in order to segment development among the UI, web workflow, services, and WESB mediations that the UI will utilize.
* Other services are identified (for each use case) and the services specifications are identified within the technical documentation
* Finally, the workflow is defined, together from the workflow business model, as well as from the user interfaces and services identified for each use case. Therefore, in reality, there are two workflows, one within the web pages as the user traverses the web pages and the transaction workflow, which manages the entire transaction.

DHS utilizes the following development teams in order to support the above methodology and is demonstrated in the following table:

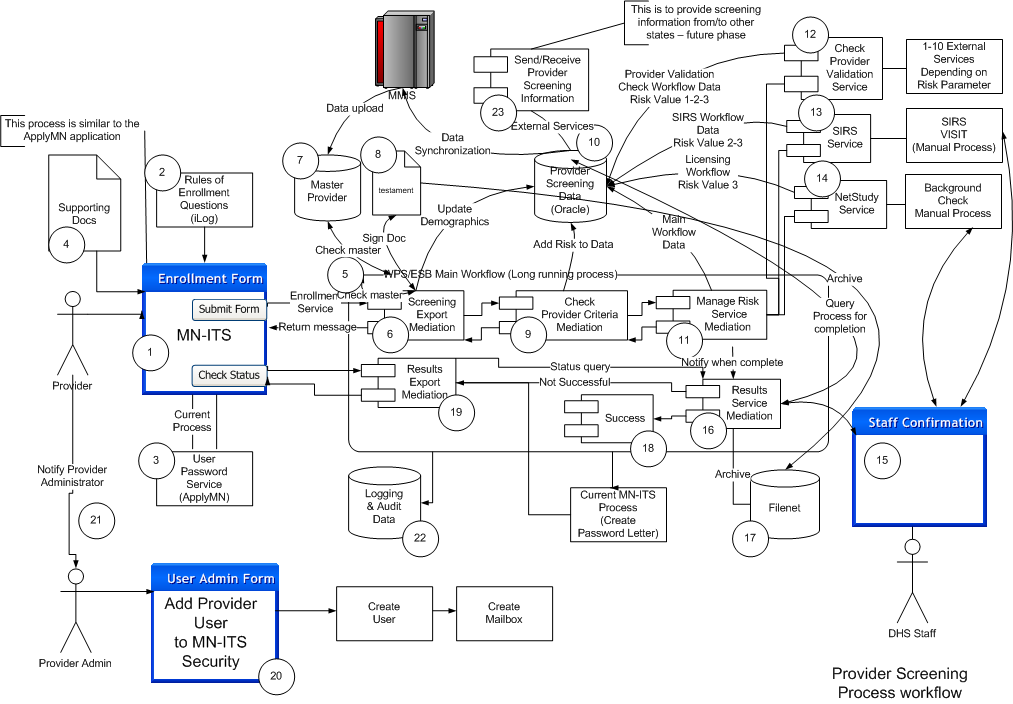
|  |  |
| --- | --- |
| **Team Required** | **Tasks** |
| User Interface Developer | * Design prototype based on use case; * develop JSF working model |
| Web Developer | * Link to Rules engine for specific use case(s); * Spring Web Flow for Use case work flow; * Map backer bean to Java object to XML transaction; * Map reverse transaction XML to Java Object to backer bean |
| Data Modeler | * Design Data model based on specs and requirements; * XML representation within flow |
| Integration Developer | * Design transaction mediation based on events and workflow; * Identify and connect interfaces required for connections among services (web service client development); * determine error handling |
| Services development team | * Design transaction services based on specifications; * Identify and define interfaces required for connections between services and client; * determine error handling. |
| Architectural | * Determine overall architecture; * Identify services required; * Determine overall workflow. |
| Deployment/Testing | * Deploy artifacts in four environments; * Conduct tests in four environments. |

# 

# Technical Architecture REQUIREMENTS

## General Technical Architecture

The following diagram depicts the possible provider screening architectural workflow based on DHS’s current architecture and the MITA process:



**The following describes the above diagram:**

1. The UI consists of a series of dynamically produced questions based on answers given by the enrollee. These questions are tightly coupled to the rules engine, which dynamically determines the next set of questions to ask.

The UI, which is decoupled from the enrollment workflow process, has two possible event choices after all questions are answered through the click of a button within the screen:

* 1. To submit an enrollment form from a web screen based on JSF framework.
  2. To check the status of the entire transaction and return a success or failure of the enrollment process.

1. The rules engine (iLOG) contains all of the rules surround the questions and answers in order to determine which questions apply to any particular provider.
2. A temporary user name/password is entered by the enrollee using the DHS existing security services. This will enable the enrollee to enter an enrollment, upload documentation, save the information, and return later to edit the form, information, and check status. Once everything is complete the enrollee submit their enrollment form to the Enrollment Service via a SOAP message.
3. Supporting documentation includes scanned images such as licenses, etc.
4. The main workflow is developed using the DHS workflow engine (WPS) within the DHS ESB (WESB). This is a long running process by virtue of a human task intervention, as well as service searches of unkown timing.
5. The initial mediation will provide initial checks: a) will query DHS master provider database and if a known provider will add that information to the business object; b) will provide a testament screen to allow the provider to agree to the DHS standard provider agreement; c) will add the information to the provider screening database.
6. The Master Provider database is the current DHS master for MMIS.
7. The provider agreement currently exists within the MN-ITS system.
8. The second step within the workflow checks the risk criteria of the enrollee and adds the risk data to the provider screening database.
9. The provider screening database does not currently exist and will have to be modeled and created. This database also contains completion flags indicating what services have been executed and when the overall process has been completed.
10. The final step in the enrollment workflow manages the risk services and calls the various services depending on the risk. This step also determines whether the enrollment is complete, successful, or unsuccessful and updates the provider screening database and sends messages to the next workflow process.
11. The first service (Check Provider Validation Service) called for all risk type enrollees provides a check of 10 or more provider validation services, i.e. this a service that combines searchs of many services. The technical requirements include:
    1. The master service must allow for a parameter to check one or up to the maxmum number of validation services.
    2. The service must update the provider screening database as the service checks are completed.
    3. Search keys include: NPI or UMPI (if available), SSN, FEIN, or Name.
12. The second service (SIRS Check Service) called provides a human task intervention in order to physically check provider’s facilities. This service waits for a secondary workflow and allows screen entry by DHS internal staff when the inspections are complete. The service updates the provider screening database upon completion.
13. The NetStudy service calls for a background check, which waits for confirmation from DHS staff that a background check is complete. The provider screening database is updated upon completion. Note that the enrollee has the option of entering the background (NetStudy) information at the point of data entry. This service is a verification service based on the response at data entry time. If the enrollee indicated that they have not completed this service, this should be validated during the application entry process and the enrollment submission should be delayed until the check is performed for this risk type candidates (this will be within the rules engine criteria).
14. This step represents staff workflow for human task interventions. There are three separate workflows: a) Service results queries for provider enrollment staff; b) SIRS visit tracking for SIRS staff; and c) Background check completion check for provider enrollment staff. Several JSF screens and perhaps separate workflows may have to be developed.
15. The Services Results Mediation is a timed process (daily), or a invoked process (from the Manage Risk Mediation), that completes the enrollment process and responds to “Check Status” queries. There may be a human task intervention to flag enrollees “complete” or “to be held”. The interaction with the provider screening database is key to the results. A NPI or UMPI is also allocated at this point, through the manual intervtion (UMPI), or through the notice to the enrollee to obtain an NPI.
16. Filenet document management system is used to archive all documents and is accessed through an existing web service.
17. The successful workflow creates a letter with a temporary MN-ITS password that allows the enrollee to register with MN-ITS. This can be displayed through the check status flow.
18. The mediation that manages the flow for: a) Checking status; b) Receiving letter of acceptance/password.
19. The current MN-ITS security process through delegated management (see Appendix D).
20. Once the enrollee receives a successful enrollment status, then they will notify their administrator, who will add their user name to the MN-ITS security system (if they are the administrator, then they will set up their own security).
21. All processes will be logged using our current logging methodology, i.e DHS application logging service and WPS message logging.
22. For future phases an external facing service provides screening information from and to external data sources, proposed to be other states and/or a national repository.

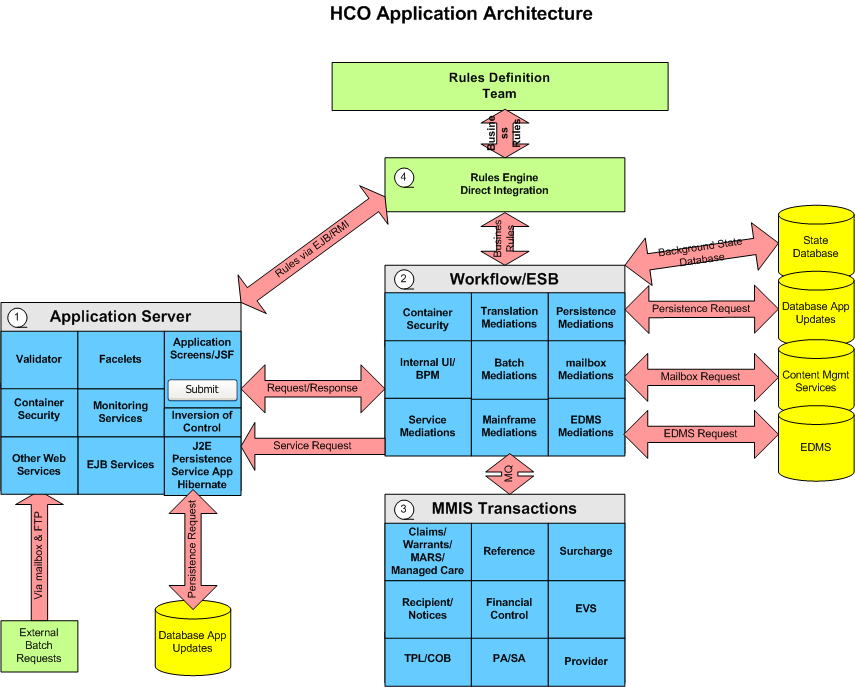
## Summary

While this document is not meant to be a final architecture design document, it does address the limitations of design for the provider screening project. Given that WPS/WESB is the central part of our SOA development as well as the future of DHS’s MMIS modernization plan, the project must conform to our existing architecture as well as our future planning.

It is recognized that this complicates re-use in other juristictions, however a general application that is truly independent of DHS architecture could not take advantage of DHS existing process and services without a major overhaul of our existing architecture and applications. The architecture does lend itself to a potential cloud application that other states could utilize.

# APPENDIX A – DHS Architecture

The following diagram represents the overall architecture of Health Care Systems:



1. The Application server hosts the J2E applications and provides the central core for the web based applications.
2. The ESB/Workflow (WPS/WESB) engine provides the central point for implementation of web services and workflow applications (BPEL).
3. The mainframe is the core processor for health care claims, as well as various other sub-systems.
4. The rules engine (iLog) provides the rules management system to the ESB as well as to other applications.
5. Other data repositories include: Filenent, Alfresco, and custom databases.

# APPENDIX B – MITA Architecture

The MITA AA connects MITA business services with technical services, as shown below. States tailor MITA business services to their environment’s needs. Business services have a common core for all MITA processes, which can adapt and extend to meet States’ special policies, rules, and deployment requirements. The MITA Framework defines its services from the abstract level to the design level of the system development life-cycle. This allows States to build service interfaces as standard interfaces without dialects caused by interpretations.

**Conceptual Technical Architecture Diagram**



The service infrastructure uses standards-based elements that allow intrastate service process integration and data sharing with other organizations and agencies. The MITA Framework is compatible with the Federal Health Architecture (FHA), the Nationwide Health Information Network (NwHIN), regional and national shared data sources, and the network on Regional Health Information Organizations (RHIOs). The MITA Framework defines a series of interoperability services based on Web Services (WS) and Extensible Markup Language (XML) message formats and protocols. The tools States need to establish interoperability, data capabilities, and other support requirements are available individually to States in groups using common facilities.

The figure below depicts some of the fundamental pieces of a SOA-based Business Service. Each service has Interface Components that include security attributes that inspect incoming messages to verify the message originator has authorization to invoke the service (e.g., only designated Medicaid staff members have authorization to approve claims.)



The Business Logic portion of the Business Service examines the received message and coordinates the execution of underlying custom, COTS, or legacy applications to provide the necessary business results for the received message. The business logic process accesses miscellaneous and enables Technical Services to perform its processing responsibilities.

The application portion of the Business Service accesses and uses a set of enterprise data management services to provide uniform access to data by using standard definitions for all shared and externally accessible data in the State Medicaid Enterprise. Data contained in COTS and legacy systems requires special handling. In the case of legacy data stores, data access routines tend to be specific to the underlying technology. Data stores used by COTS packages often have proprietary data structures; data access routines and only vendor-approved Application Programming Interfaces (API) access them. In addition, not all COTS data stores are externally accessible, not even via API. The enterprise data management services provide transparent data management services to all accessible data stores in State Medicaid Enterprise. The data access services are platform independent, so that, as legacy systems are phased out, or databases restructured, no changes to the enterprise data management services are visible to the consumer.

Using SOA standard application methods, systems invoke services at different architecture layers. To maximize reuse across the State Medicaid Enterprise, the State Medicaid Agency (SMA) should standardize the service descriptions, invoking messages for all of the services connected to the Enterprise Service Bus (ESB), and as many of the lower level Technical Services as possible.

**Application Architecture Key Components**

The figure belowdepicts the relationship between the MITA infrastructure and services. Business and technical services link by service infrastructure elements using the important integration element known as the Enterprise ESB. David Chappell in *Enterprise Service Bus* (O’Reilly Media, 2004) describes an ESB as:

“a standards-based integration platform that combines messaging, Web services, data transformation, and intelligent routing to reliably connect and coordinate the interaction of significant numbers of diverse applications across extended enterprises with transactional integrity.”

***There is currently no universal agreement within the industry regarding the components and functionality of an ESB. The MITA Framework concept of an ESB follows Chappell’s definition and includes common elements and those critical to linking business services and technical services.***

The service integration and interoperability methods provide loose connectivity and are essential enablers of flexibility. The consistent use of this service approach is a important element to designing the SOA. The significant components of the MITA AA are:

* ESB and Access Channels
* Service Management Engine
* Service Gateways and Mediators
* Externally Distributed Computing and Data Access (i.e., Cloud Computing
* MITA Framework Documents
* Interoperable Services
* Security and Privacy (S&P)



# APPENDIX C – DHS SOFTWARE STANDARDS LIST

The following is a list of software and tools used in the development of Health Care applications:

|  |  |  |
| --- | --- | --- |
| **Software/Tools** | **Version** | **As of Date** |
| **Servers/Systems** | | |
| iPlanet 6.1.5 web server | 7.0 | 1-1-2012 |
| Websphere Process Server | 7.0.04 | 1-1-2012 |
| Websphere Enterprise Service Bus | 7.0.04 | 1-1-2012 |
| Websphere Server (WAS***)*** | 7.0.0.21 | 1-1-2012 |
| Oracle RAC | 11g | 1-1-2011 |
| MQ Series | 6.0 | 7-1-2007 |
| PM4Data | 8.5.1 | 1-1-2009 |
| Oracle COREid (Oblix) | 7.0.4 | 7-1-2007 |
| **Development Tools** | | |
| Rational Architect | 7.5.5.1 | 1-1-2012 |
| Rational Application Developer | 7.5.5.1 | 1-1-2012 |
| Websphere Integration Developer | 7.0.04 | 1-1-2012 |
| WTX Design Studio | 8.3.0.3 | 1-1-2012 |
| Metastorm Workbench | 3.0 | 1-1-2012 |
| UltraEdit-32 | 10.10a | 2004 |
| Toad | 9.0.1 | 2004 |
| WinSCP3 |  | 2004 |
| WinZip |  | 2004 |
| **Development Frameworks** | | |
| J2EE | 6 SDK 5.0.22 | 1-1-2012 |
| EJB | 3.0 | 1-1-2008 |
| JSF | 2.0 | 6-1-2008 |
| Struts (if required) | 1.2 (2.0.11) | 1-1-2008 |
| DHSBase (internal framework) | 1.2.7 | 2004 |
| Hibernate | 3.6.3 | 1-1-2012 |
| Apache Axis | 1.4 (not used) | 7-1-2007 |
| Spring | 3.0.5 | 1-1-2012 |
| Jaxb | SDK | 1-1-2012 |
| Jax-ws | SDK | 1-1-2012 |
| **Source Control/Build Management** | | |
| Visual Sourcesafe | 5.0 | 2004 (to be phased out) |
| Rational Clearquest | 7.0.1.0 | 1-1-2010 |
| Rational Clearcase | 7.0 | 1-1-2010 |

# APPENDIX D – DHS Security Architecture



Process overview: Login using the WAS with Access System single

sign-on

1. The user attempts to access an WebSphere resource that is protected by the Access System.

2. WebGate (or AccessGate) intercepts the request and prompts for a username and password, using the Basic challenge method.

3. WebGate passes the user's credentials to the Access Server.

The Access Server checks the user data store (the directory) and authenticates the user. WebGate sets an ObSSOCookie in the request.

4. The Web server forwards the user request to the WAS.

The Oracle Access Manager TAI gets the request and confirms that the user has been authenticated.

5. WAS recognizes that the Access System has authenticated the user and creates an LTPA token.

The remaining steps in this process are the same as steps 5-7 in Scenario 1. The WAS and both Web servers must belong to the same domain.

6. The Federated Repository queries the LDAP for a list of groups to which the user belongs.

7. The Connector for WebSphere returns this information to the WAS.

If you are not using the CMR, the WAS checks the deployment descriptor for a user-security or group-security role mapping. If the user or group belongs to a security role that is allowed to access the resource, the WAS allows the user to access the resource.