

**Office of Information and Technology  
Product Development**

**Home Telehealth Capability Enhancements (HTCE)  
Integrated Home Telehealth Application (IHTA)**

**Software Architecture Document**



**February 2014**

**Version 1.8**



## Revision History

*The revision history cycle begins once changes or enhancements are requested after the Software Architecture Document has been baselined.*

Date	Revision	Description	Author
02/06/2014	1.8	Removed "future" from p. 7 regarding future integration of IHTA with HDR for DMP Response Reports and Census/Survey	
04/12/2013	1.7	Updated Figures 2 (removed External Systems), 15, 16 (added screenshots from the VCWI Prototype developed in Release 5.0), 17 (removed IVR; added HDR; added HT Reports), 21 (removed IVR from DMZ; added HDR to VA WAN), 32 (added HT Reports to IHTA Modules); updated Sections 5.2 and 9 to reference Martinsburg as Primary Production site; updated Section 2 to separate current modules from future modules (added a note that the CCCWI/VCWI will be delivered in future releases); updated Section 2.4.4 to include HDR as a SOR (removed IVR)	
03/07/2013	1.6	Numerous updates (e.g., images, lists, links, etc.) to conform to 508 standards; completed run of PAWS; confirmed 508 compliant 03/07/2013; transition to Martinsburg/Hines Production delayed – changed back to Falling Waters/Hines	
01/16/2013	1.5	Updated Sections 5.2 and 9; Figures 3, 24, 30, 31 to include changes in Production (Martinsburg & Hines)/Development Environments (Hines)	
08/13/2012	1.4	Updates to Figures 18, 24, 30, 31; added note to Approval Signatures section stating that approval signatures will be obtained in a subsequent release.	
06/15/2011	1.3	Added reference to the HTCE TSPR; updated IHTA screenshots to reflect UI Enhancements in R4S3; added to Section 4.1 text stating that the Care Coordinator's Common Web Interface and the Veteran's Web Interface will be delivered in subsequent releases	
04/15/2011	1.2	Changed project name to Home Telehealth Capability Enhancements (HTCE)	
11/16/2010	1.1	Updated section on My HealtheVet; changed Office of Enterprise Development to Product Development	
10/06/2010	1.0	Baseline release (HTIE 2.5x)	



## Table of Contents

<b>1. Introduction.....</b>	<b>1</b>
1.1. Scope.....	1
1.2. Definitions, Acronyms, and Abbreviations.....	2
1.3. References .....	2
<b>2. IHTA Architecture.....</b>	<b>3</b>
2.1. I3 Architecture.....	3
2.2. Framework Code .....	4
2.3. Common Code.....	4
2.4. Architectural.....	4
<b>3. Logical View.....</b>	<b>8</b>
3.1. Authentication .....	8
3.2. Authorization.....	9
3.3. Registration.....	10
3.4. Registration Approval .....	11
3.5. User Experience .....	12
3.7. Access to IHTA via MHV .....	15
<b>4. Process View .....</b>	<b>19</b>
4.1. Enterprise Context.....	19
4.2. Enterprise System Integration .....	20
4.3. Portlet Integration .....	22
4.4. Service Oriented Architecture.....	23
4.5. Event-Driven .....	23
4.6. Batch Processing.....	23
4.7. Business Intelligence.....	23
4.8. Logging .....	24
4.9. Exception Management.....	24
4.10. Session Management.....	24
<b>5. Deployment View.....</b>	<b>24</b>
5.1. Application Topology .....	24
5.2. Physical Topology.....	25
<b>6. Implementation View.....</b>	<b>30</b>
6.1. Application Layering .....	30
6.2. Technology Stack .....	31
6.3. Application Components.....	31
6.4. Application Context.....	32
6.5. Test Scaffolding.....	32
6.6. Development Software .....	33
<b>7. Data View.....</b>	<b>34</b>
<b>8. Quality .....</b>	<b>34</b>
<b>9. Architectural Mechanism.....</b>	<b>35</b>
<b>Attachment A - Approval Signatures .....</b>	<b>36</b>

## List of Figures

Figure 1: “4+1” Architecture View Model shown with additional Data View.....	2
Figure 2: Load Balancing and Fail-Over .....	5
Figure 3: Transparency .....	6
Figure 4: VA User Authentication Process Flow .....	9
Figure 5: Non-VA User Authentication Process Flow .....	9
Figure 6: IHTA Authorization Process Flow .....	10
Figure 7: Registration Process Flow .....	11
Figure 8: Registration Approval Process Flow.....	12



Figure 9: IHTA Home Page .....	13
Figure 10: IHTA User Registration Page .....	13
Figure 11: IHTA Login Page.....	14
Figure 12: IHTA Portal Page .....	14
Figure 13: My HealtheVet Home Page .....	15
Figure 14: My HealtheVet Login Page .....	16
Figure 15: IHTA Access from My HealtheVet Portal Page .....	17
Figure 16: Veteran's Common Web Interface Module (IHTA/MHV Integration) .....	18
Figure 17: Enterprise Context .....	19
Figure 18: Vendor Server - Inventory Tracker .....	20
Figure 19: Home Telehealth Database and the VA Enterprise LDAP .....	21
Figure 20: My HealtheVet and IHTA .....	21
Figure 21: Common Data Aggregator.....	22
Figure 22: Application Topology.....	25
Figure 23: Environment Overview.....	25
Figure 24: Development Environment .....	26
Figure 25: SQA Environment .....	27
Figure 26: Production Environment .....	27
Figure 27: Disaster Recovery Environment .....	28
Figure 28: Development Data Center .....	29
Figure 29: Production Data Centers .....	29
Figure 30: IHTA Architectural Layers .....	30
Figure 31: IHTA Technology Stack .....	31
Figure 32: IHTA Application Components.....	32

### List of Tables

Table 1: Enterprise Service and Application Summary .....	19
Table 2: Application Topology .....	24
Table 3: IHTA Development Software and Tools.....	33



# 1. Introduction

The Veterans Health Administration (VHA) is mandated to provide non-institutional care services to Veteran patients with chronic conditions under the Millennium Bill. The VHA National Care Coordination/Home Telehealth Program (CCHT) is required to provide 50% of VHA's non-institutional care needs by 2011 (75,000 patients). Additionally, Veterans Integrated Service Networks (VISN) have instituted CCHT-supported services for acute care management, chronic care management, and health promotion/disease prevention because of the efficiency and cost-effectiveness of these services. CCHT, within the Department of Veterans Affairs (VA), uses health informatics, disease management, and Home Telehealth (HT) technologies to target care/case management efforts, thereby facilitating access to care, and improving the health of Veterans. CCHT changes the location where health care services are routinely provided and supports Veterans' preferences to live in the least restrictive settings possible. To support its mission and encourage a partnership with patients and their caregivers in self-managing their condition, the Office of Telehealth Services (OTS) in the Veterans Administration Central Office (VACO) is responsible for care coordination implementation throughout VA. It is anticipated that the population being served is likely to more than double over the next three years. This expansion places increased pressure on the existing information technology (IT) infrastructure, with resulting risk management issues, and compounds the costs and organizational barriers to expansion that avoidable data entry creates. Consequently, VHA needs a more robust infrastructure and reliable messaging support to provide risk mitigation for potential failure of vendor systems due to technology or financial problems.

The specific business objectives are as follows:

- Provide an efficient and cost-effective platform to support and maintain a projected 160% rise in the census of HT patients
- Mitigate risk in the event of a systems failure by a HT vendor (including Chapter 7 insolvency) by enabling continuity of technology-based patient monitoring and communication capacities
- Provide IT support for necessary clinical process re-engineering that will enhance care, ensure patient safety, and free clinician time by automating data processes that are currently performed manually

The Home Telehealth Capability Enhancements (HTCE) project was established to develop a means to meet these business objectives. The Integrated Home Telehealth Application (IHTA) is the proposed means for doing so. This document details the architecture of, interfaces to, and tools to support IHTA.

## 1.1. Scope

The IHTA architecture, interfaces, and tools will be documented in the following sections through a common view of software architecture known as the Philippe Kruchten's "4+1" architecture view model. The "4+1" architecture view model is widely recommended to articulate the details of a software architecture. Due to the potential of having multiple, enterprise databases in the HTCE context, it is important to add one additional view, the "Data View". The complete "4+1" architecture model (with the additional Data View) is illustrated in Figure 1.

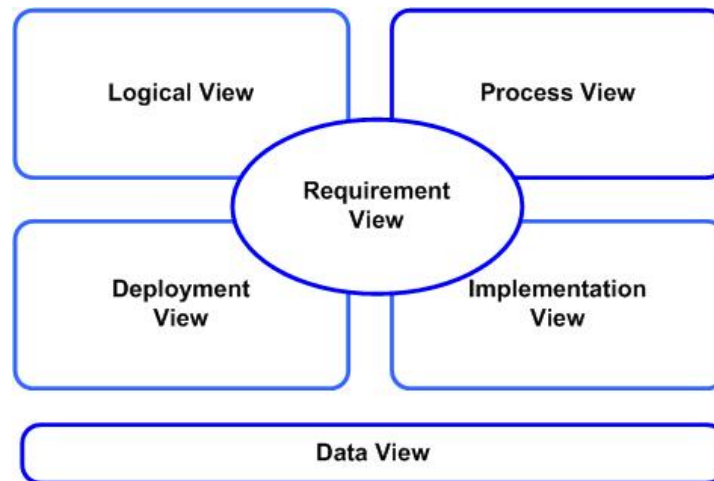


Figure 1: “4+1” Architecture View Model shown with additional Data View

Figure 1 identifies the following architectural views of a software application:

- Requirement View – Architectural functional, non-functional requirements of the system
- Logical View – Architectural functional elements that are exposed to end-users
- Process View – Application data flow to and from enterprise and external systems
- Deployment View – The application’s physical topology across supported environments
- Implementation View – Architectural design, software stack and artifacts of the application
- Data View – Application data ownership and separation

## 1.2. Definitions, Acronyms, and Abbreviations

For terms, definitions, and acronyms used in this document, please refer to the HTCE Terms, Definitions, and Acronyms (TDA) document available in the HTCE Technical Services Project Repository (TSPR) at the following link: [REDACTED]

For a listing of all acronyms used by VA, please refer to the VA Acronym Lookup site at the following link: [REDACTED]

## 1.3. References

This document references the following:

- HTCE TSPR – The central repository for all approved and released HTCE project documentation.
- HTCE Share Point – The repository for working HTCE project documentation, containing requirements, proposed architectures, and tools. For access, please contact the HTCE Project Manager.
- “4+1” View Model of Software Architecture – Presents a model for describing the architecture of software-intensive systems, based on the use of multiple, concurrent views.
- *HTCE Database Design Document*



## 2. IHTA Architecture

The architectural goal of IHTA is to provide a flexible, maintainable, and resilient platform for HT business functions. Each business function that leverages IHTA is constructed as an application module of IHTA and shares the same set of IHTA infrastructure and business services that interface with enterprise systems, such as the Enterprise Interactive Voice Response (IVR) System and My HealtheVet (MHV) System.

At the core of IHTA is an integrated internet and intranet application that provides business functionality needed by OTS. It is accessible through a Web portal interface with possible linkage from the MHV System. Current portlets (modules) of IHTA include the following: Administration, Inventory Tracker, Disease Management Protocol (DMP) Development Tool, and HT Reports. Future modules may include the Veteran's Common Web Interface (VCWI) and the Care Coordinator's Common Web Interface (CCCWI). Access to each IHTA module is restricted by user roles and permissions granted during the registration process.

To ensure that the IHTA architecture adheres to the Office of Product Development (PD) architectural standards, the IHTA architecture is based on the Clinical Information Support System (CISS) architecture, which has been certified as meeting the relevant PD standards. The IHTA architecture contains the key constructs discussed in the subsequent sections.

**NOTE:** The VCWI and the CCCWI will be delivered in subsequent releases following customer approval to move forward.

### 2.1. I3 Architecture

The IHTA architecture is built upon three main principles: **Integration**, **Isolation**, and **Innovation**. These principles encompass the “**I3 Architecture**”, as each principle plays a major role in the software architecture.

#### 2.1.1. Integration

The IHTA architecture supports industry standards for application integration. IHTA is designed to allow for integration with external systems, such as MHV, as either a portlet or a Uniform Resource Locator (URL) link, depending on the business requirements of the target host system.

#### 2.1.2. Isolation

The IHTA modules can either be developed as an integrated application module or as a completely separate system with access restricted by the IHTA user registration progress. This isolation allows for parallel development, loose coupling with interfaces, and disparate release cycles. The level of isolation can range from simple (built by the HTCE project team but packaged in a separate Enterprise ARchived [EAR] artifact) to complex (built by different teams and hosted remotely).

#### 2.1.3. Innovation

Providing innovation in the user's experience is a major goal of the IHTA architecture. This architecture will take advantage of opportunities for innovation embedded in existing PD architectural standards. It will also advance existing PD technology in the following ways:

- Innovate, re-engineer, and optimize business processes by working closely with customers in an Agile approach
- Engineer innovative solutions to challenging requirements that lack enterprise solutions



- Provide a world-class user experience based on an innovative visual design that leverages a Section 508-compliant Rich Internet Application (RIA)

## **2.2. Framework Code**

The IHTA framework will leverage the existing CISS framework libraries and technology stacks to promote consistency and cohesiveness across the HTCE and all other projects that leverage the CISS framework. However, whenever IHTA business requirements call for the specific use of an industry-standard technology or a software component, such as a User Interface (UI) widget or business service that is not part of the CISS framework, the IHTA development team will either extend from the existing framework component or implement a new component.

## **2.3. Common Code**

All IHTA modules will share the same, common codebase packaged in a common Java ARchive (JAR) file. This codebase consists of business entities, business services, UI widgets, and utility classes that are common to IHTA. This common codebase promotes cohesiveness between application modules and consistent employment of technology stacks across the IHTA modules.

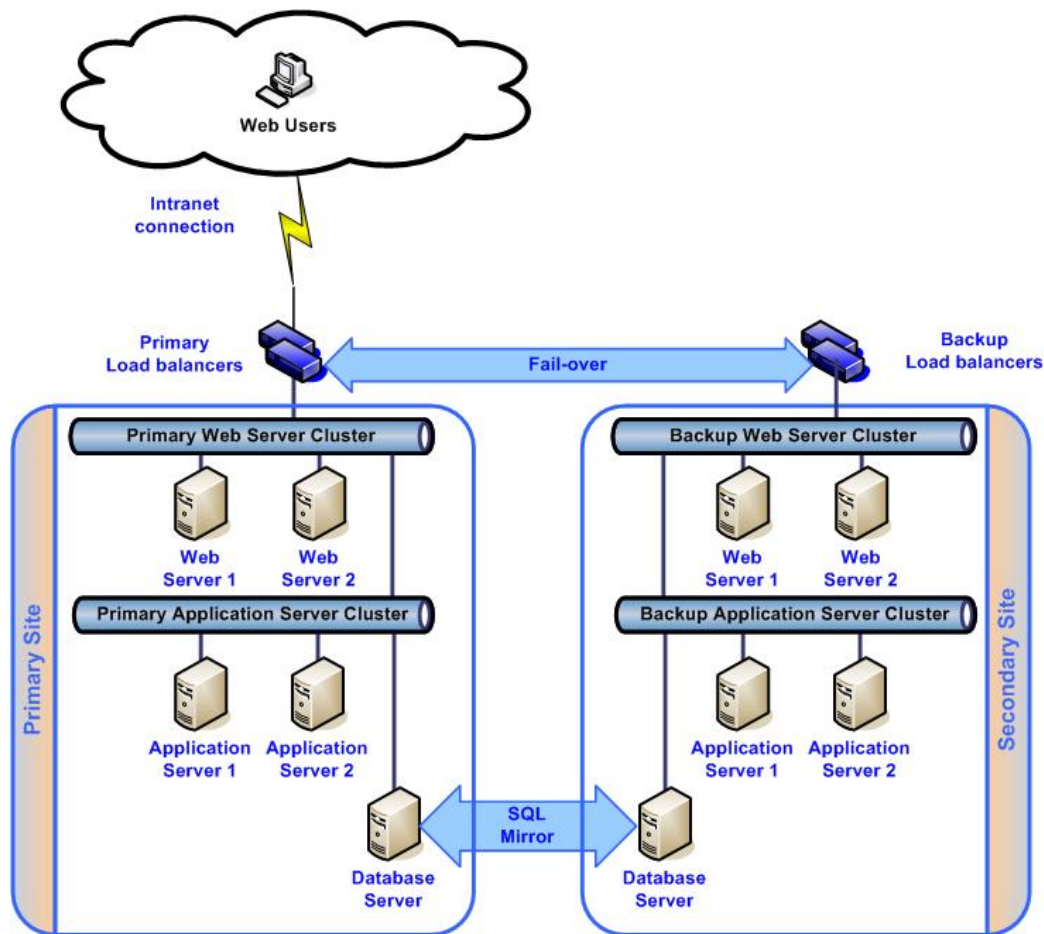
## **2.4. Architectural**

This section identifies the architectural requirements of IHTA.

### **2.4.1. Load Balancing and Fail-Over**

IHTA employs load-balancing techniques across a Web server, an application server, and database nodes within a cluster. A cluster consists of multiple server instances running simultaneously and working together to provide increased scalability and reliability. A cluster appears to clients as a single instance. Server instances in a cluster can run on the same machine or be located on different machines. A failure of one server instance in a cluster will not impact the cluster's operation. Instead, requests to the failed server instance are automatically failed over to the remaining server instances in a cluster for processing. Figure 2 depicts the IHTA Web server cluster, application server cluster, and database server cluster.





**Figure 2: Load Balancing and Fail-Over**

The load balancing and fail-over technique will be employed across geographic locations. In the Geographic Load Balancing (GLB) paradigm, an application is deployed to all production sites where the application's production environment is hosted. Application deployment is active and available for processing requests at all sites. This feature makes this model different than a standard, standby disaster recovery (DR) deployment at a remote site, which only has one active deployment.

IHTA will employ a standard deployment model, which has one active deployment at the production site and an inactive deployment at the secondary site. Load balancing and fail-over are implemented in the Web server and application server clusters where requests to a cluster will be distributed equally to all instances in a cluster. Failure to an instance in a cluster will not impact operation of the remaining instances in a cluster. In the event of catastrophic failure at the primary site, incoming requests to the primary site will be failed over to the secondary site, and the application will be activated manually to process incoming requests. Future enhancement will be to implement a database cluster across the production and DR sites so that the application, deployed at the secondary site, will be automatically activated in the event of failure at the primary site.

## 2.4.2. Near-Real-Time Data Replication

GLB involves the use of multiple deployments that are employed simultaneously. Replicating and/or partitioning near-real-time data across deployments is critical to successful application of the GLB paradigm.



Even without a GLB deployment, near-real-time data replication is critical to ensure a proper Continuity of Operations (COOP) strategy. The IHTA architecture implements a standard, standby DR deployment in which near-real-time data replication across the primary and secondary sites are ensured. Although IHTA is deployed to both the primary and secondary sites, IHTA deployment is only activated at the secondary site when there is catastrophic failure of IHTA at the primary site. In the event of catastrophic failure, incoming requests to the primary site's load balancers will be failed over to back-up load balancers at the secondary site. The primary site's main data store is synchronized with the secondary site's back-up data store through database log-file shifting to ensure near, real-time data replication.

### 2.4.3. Transparency

Due to the current state of the PD enterprise architecture, the IHTA architecture must be flexible enough to accommodate new features of the enterprise architecture as they are identified. This transparent nature of the IHTA architecture allows for changes in its business services and database mapping while minimizing impacts to the UI.

Clear separation and grouping of components based on their technical functions allow for technology changes in each tier while minimizing impacts to the overall architecture. The CISS Framework will be used as a backbone to ensure cohesiveness (see Figure 3).

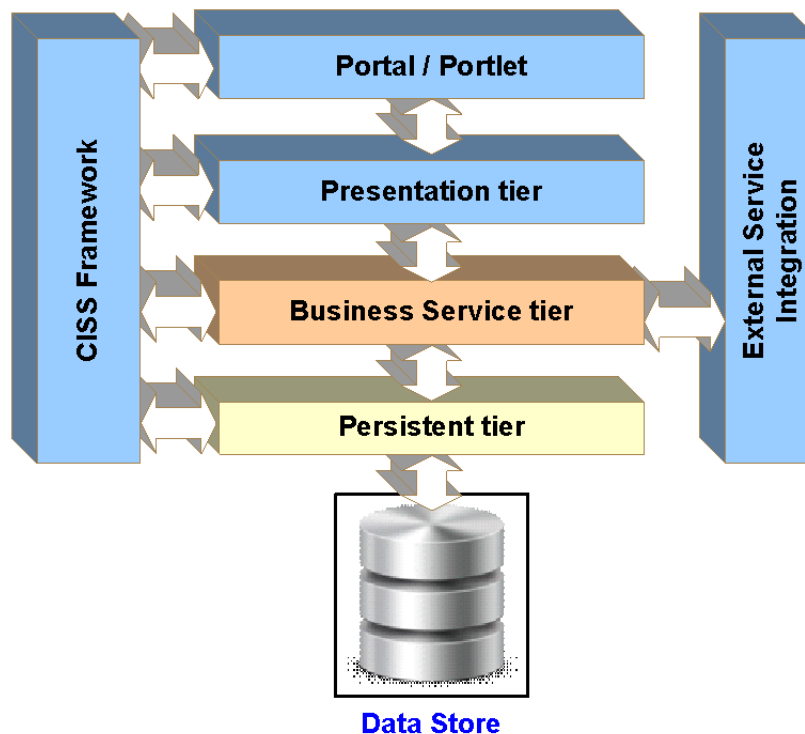


Figure 3: Transparency

### 2.4.4. Integration

IHTA will provide a consolidated view of its own modules and existing legacy HT applications. Given the investment spent on existing legacy HT applications, the IHTA development team will decide, on a case-by-case basis, the best strategy for the inclusion of HT applications into IHTA.

Generally, this will fall into one of the following main strategies:



- Legacy enablement – This can actually take many forms, although all are focused on providing a new UI (i.e., skin) on existing application services. For example, invoking services of the Census Application from within IHTA using a Web service.
- Embed “As-Is” into IHTA as a Java portlet. For example, the Census Application might be embedded in IHTA as a portlet.
- Re-implement the legacy application such that it can be either embedded in IHTA as an application module or hosted on the IHTA portal server as a portlet.

The IHTA architecture calls for integration with the following VA Enterprise systems: Health Data Repository (HDR), Lightweight Directory Access Protocol (LDAP), HT database, Vendor Servers, and MHV.

- HDR: Integration between IHTA and HDR will allow the exchange of vital sign, survey, and DMP data for the HT Report module, which will include DMP response reports and census/survey reports.
- LDAP system: Integration to LDAP allows IHTA to authenticate and to authorize VA user access to IHTA portlets and services.
- HT database: Integration to the HT database is critical for IHTA since HT data is centrally stored in the HT database.
- Vendor Servers: Integration to Vendor Servers will allow the exchange of vital sign, survey, and DMP data.
- MHV: Integration with the MHV system provides Veterans access to IHTA from the MHV portal (future).

**NOTE:** Integration with MHV will occur in future releases of IHTA pending a customer decision to move forward with the VCWI.



### 3. Logical View

The Logical View of the IHTA architecture will focus on functionality exposed to end users. These include authentication, authorization, user registration, registration approval, and the general user experience.

#### 3.1. Authentication

Authentication must address the target identity providers and process flow as described in the following sections.

##### 3.1.1. Identity Providers

The Solutions Analysis and Architecture (SAA) team recommends application authentication on the LDAP (implemented as Microsoft Active Directory) and Security Assertion Markup Language (SAML). While the VA Enterprise LDAP (LDAP) is in place, enterprise authentication using SAML is still under development. Until this capability is available, the IHTA architecture will leverage the existing Java Security Service (JSS) to authenticate VA users against LDAP.

Access to IHTA will be granted upon successful authentication against the existing LDAP. It is important to note that logging into IHTA will not grant access to all application modules or embedded systems in IHTA. There will be authorizations that govern access to each of the application modules or embedded systems. There will also be authorizations that govern access within each application module.

IHTA will authenticate its users against LDAP, but not implement direct SAML. Since users at each VA Medical Center are present in LDAP at the VISN level (e.g., vXX.med.va.gov, etc.), IHTA will establish a one-way trust with the VA Enterprise LDAP. This will allow all VISN users access to IHTA.

##### 3.1.2. Process Flow

LDAP will be used to authenticate VA users of IHTA. IHTA database tables will be *used* to store credentials for authenticating non-VA users.

###### 3.1.2.1. IHTA Authentication Process Flow

IHTA users are grouped into two groups: VA Users and Non-VA users. VA users are those who can access the VA network with a valid VA network ID and those whose credentials are stored in LDAP. Non-VA-users are those who do not have a VA network ID. IHTA will authenticate VA users using LDAP, while authentication of Non-VA users will be done using credentials stored in IHTA database tables. Figures 5 and 6 depict the VA and Non-VA authentication processes respectively.

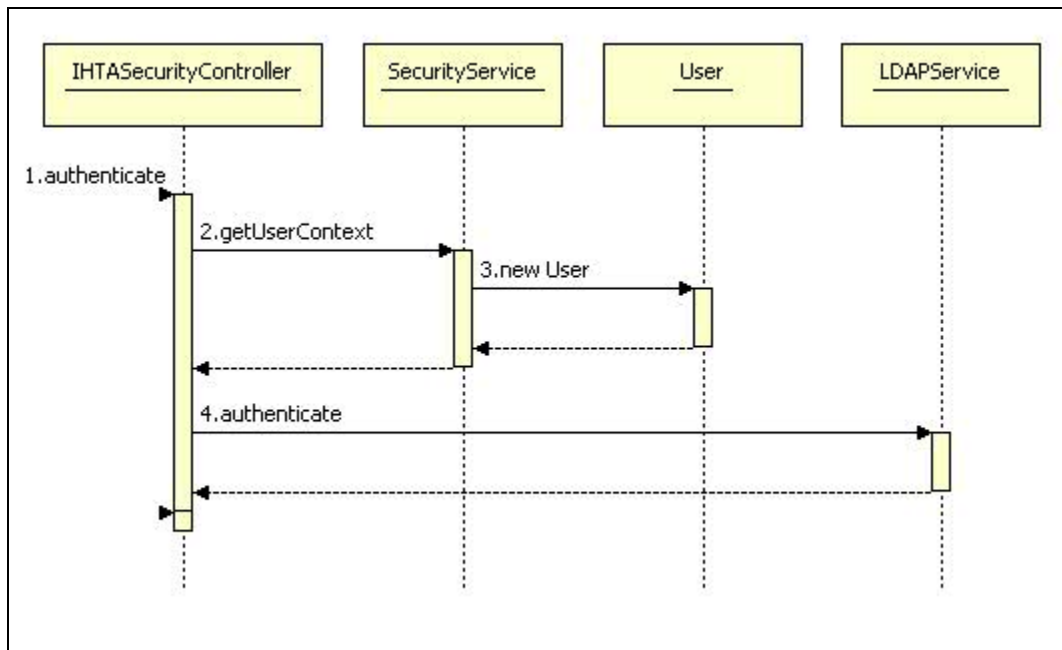


Figure 4: VA User Authentication Process Flow

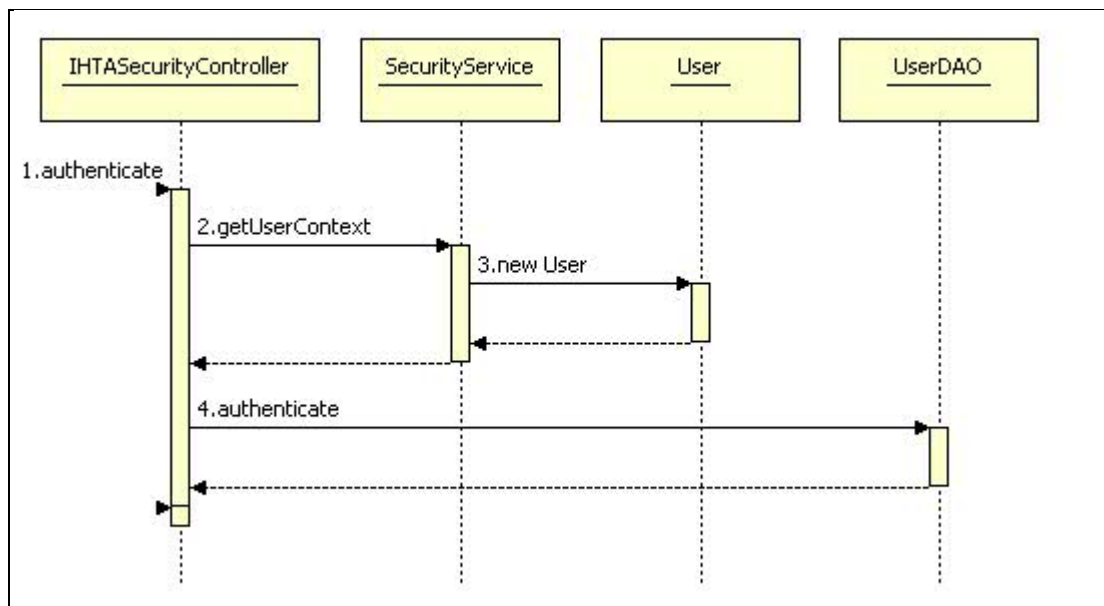


Figure 5: Non-VA User Authentication Process Flow

## 3.2. Authorization

This section addresses the target policy store and process flow for authorization.

### 3.2.1. Authorization Policy Store

For VA users, IHTA will use the existing LDAP for authentication storage. For non-VA users, IHTA will utilize the IHTA database tables to store authorization information, such as credentials and successful and failed log in attempts. The Manage Users Screens of the Administrative module of IHTA allow applicable administrators to assign authorizations to users with different roles and permissions.



### 3.2.2. IHTA Authorization Process Flow

Figure 6 illustrates a sequence of authorization checks to grant user access to IHTA.

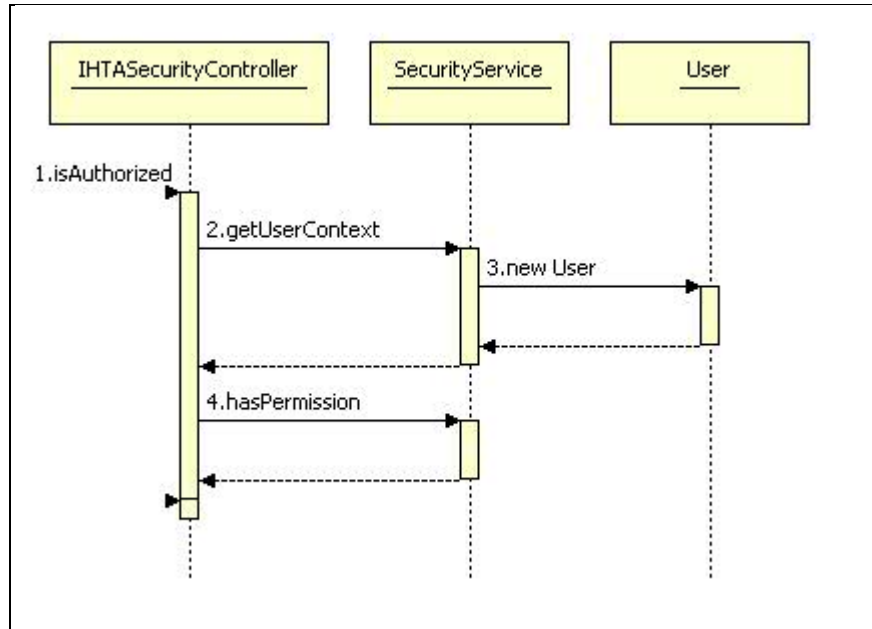


Figure 6: IHTA Authorization Process Flow

## 3.3. Registration

This section addresses the target policy store and process flow for registration.

### 3.3.1. Registration Policy Store

For VA users, the IHTA Registration Screens capture a user's VA network ID to store it in the designated IHTA database table.

### 3.3.2. Registration Process Flow

Figure 7 illustrates the registration process flow for IHTA.

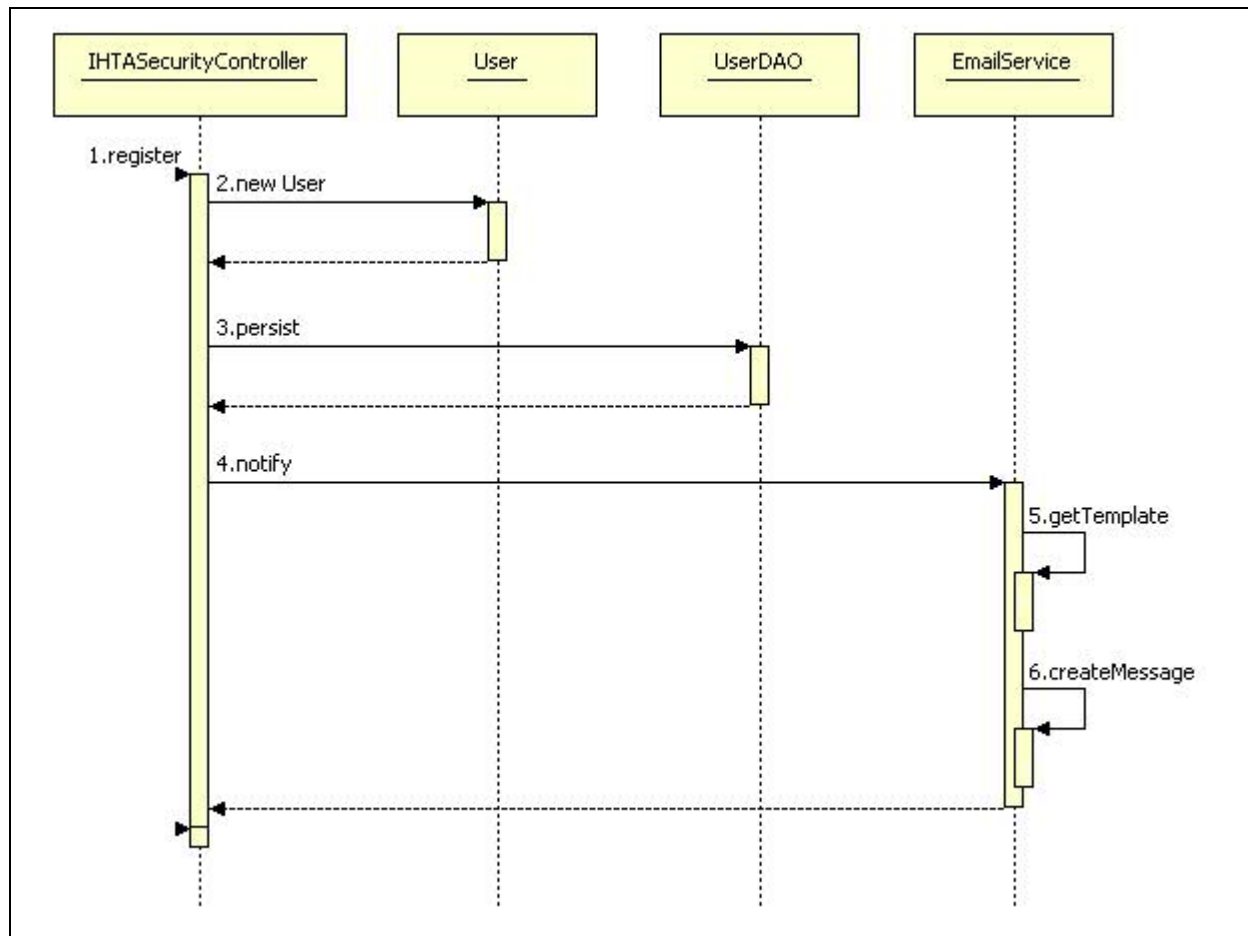


Figure 7: Registration Process Flow

## 3.4. Registration Approval

This section addresses the target policy store and process flow for registration approval.

### 3.4.1. Registration Approval Policy Store

The registration approval process for IHTA is performed by a sufficiently empowered administrator. The screens of the registration approval process capture and store IHTA database information about user roles, groups, and permissions related to specific application modules of IHTA.

### 3.4.2. Registration Approval Process Flow

Figure 8 illustrates the process flow for IHTA registration approval.

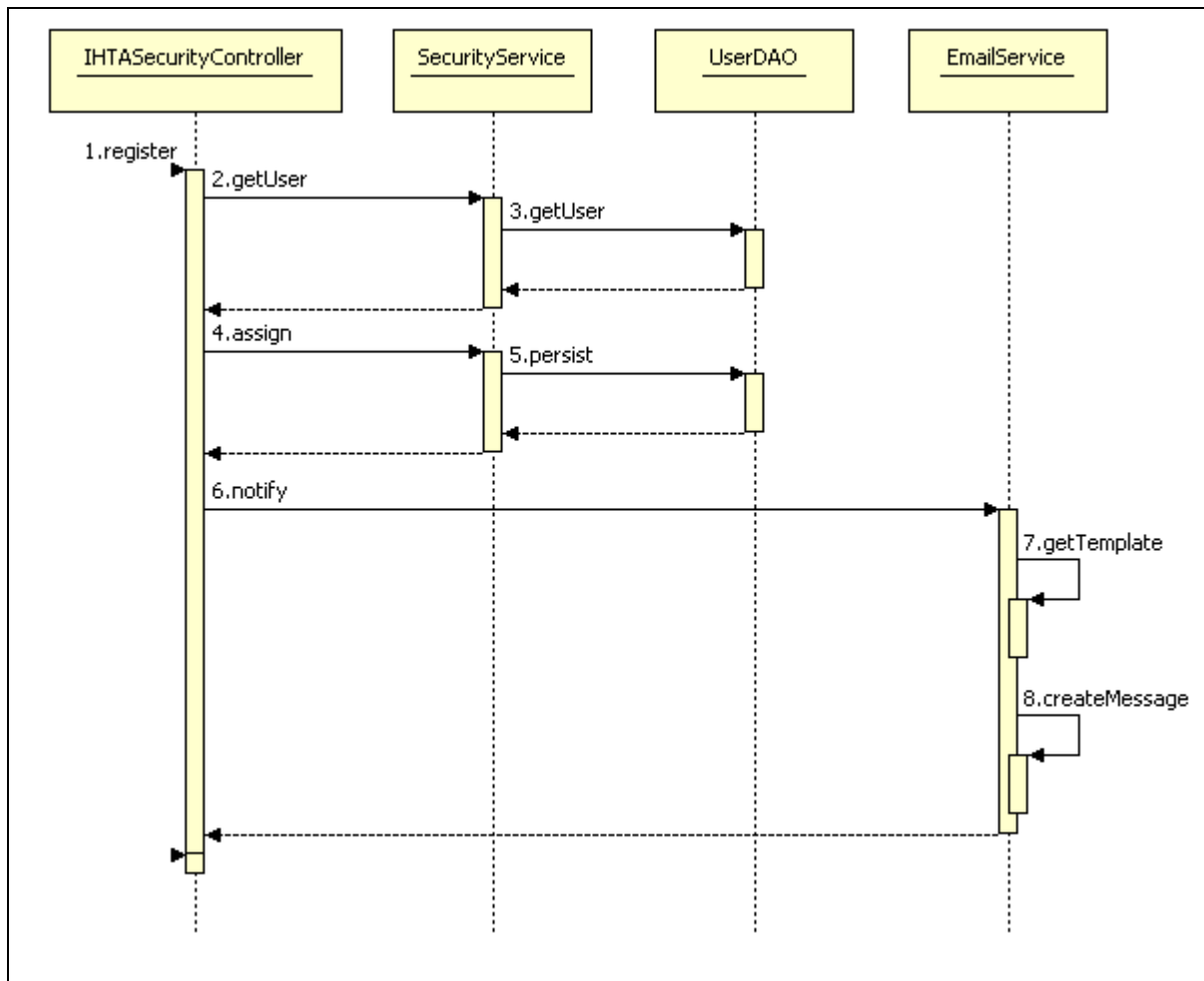
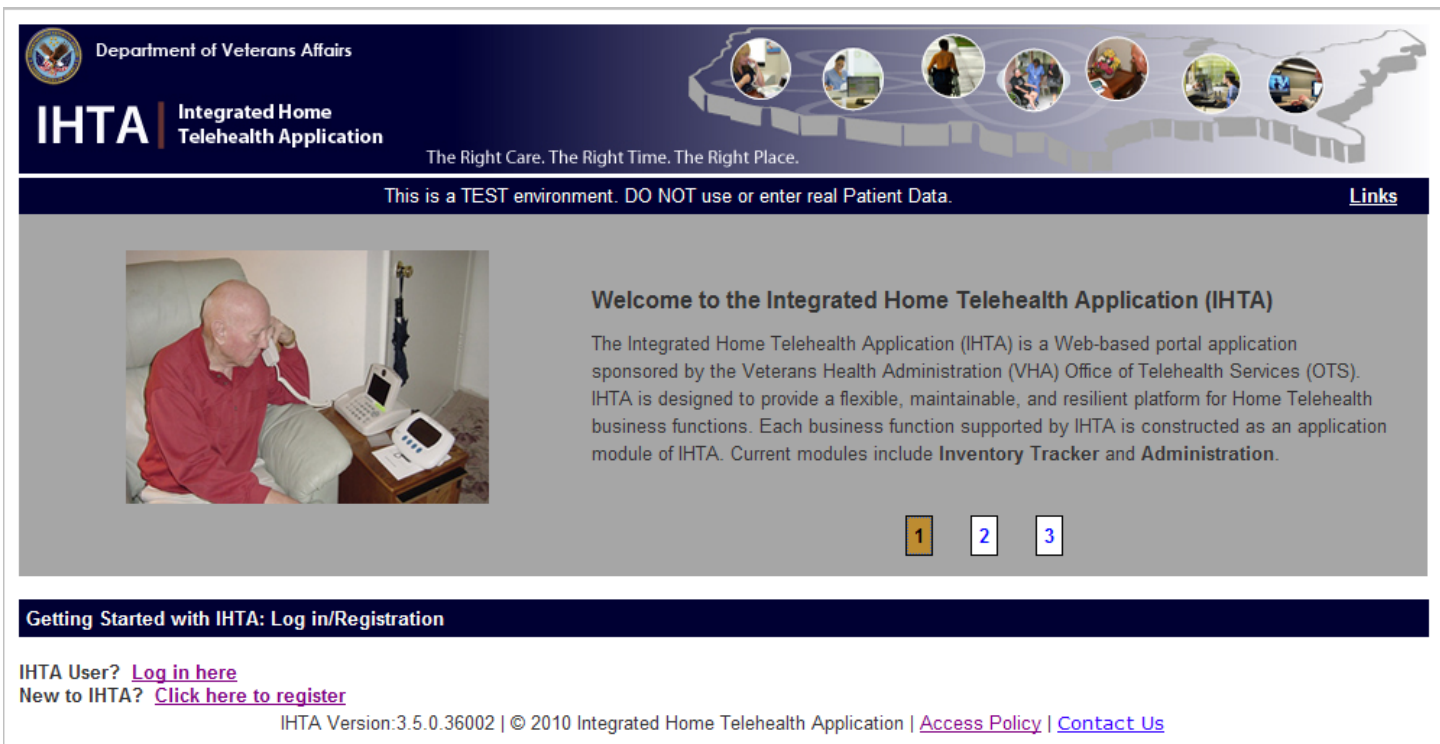


Figure 8: Registration Approval Process Flow

### 3.5. User Experience

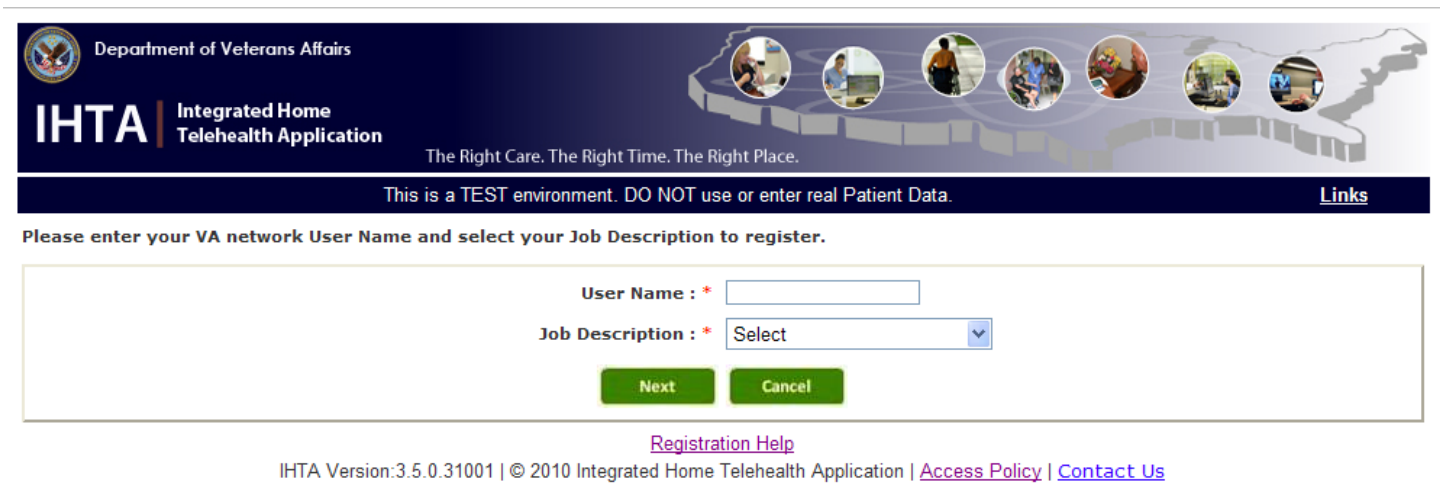
Figures 10 through 13 are comprised of storyboards for the IHTA Home Page, Registration Page, Login Page, and Portal Page (contains the application modules that are displayed depending on a user's role and permissions). The first screen the user will see is the IHTA Home Page. The IHTA Home Page contains a list of commonly used Home Telehealth websites, including, but not limited to, Patient Census, Average Daily Census, and Network and Server Status. This page also contains the IHTA Login and Register Here links (see Figure 9). To register, the user will click on the Register Here link from the IHTA Home Page. The IHTA User Registration page will display (see Figure 10). The user will only need to register once for access to IHTA. To login, the user will use the IHTA User Authentication or Login page (see Figure 11). After successful login, the IHTA Portal Page will display. The Portal Page displays the application modules that the users have access to according to their assigned role and permissions (see Figure 12).





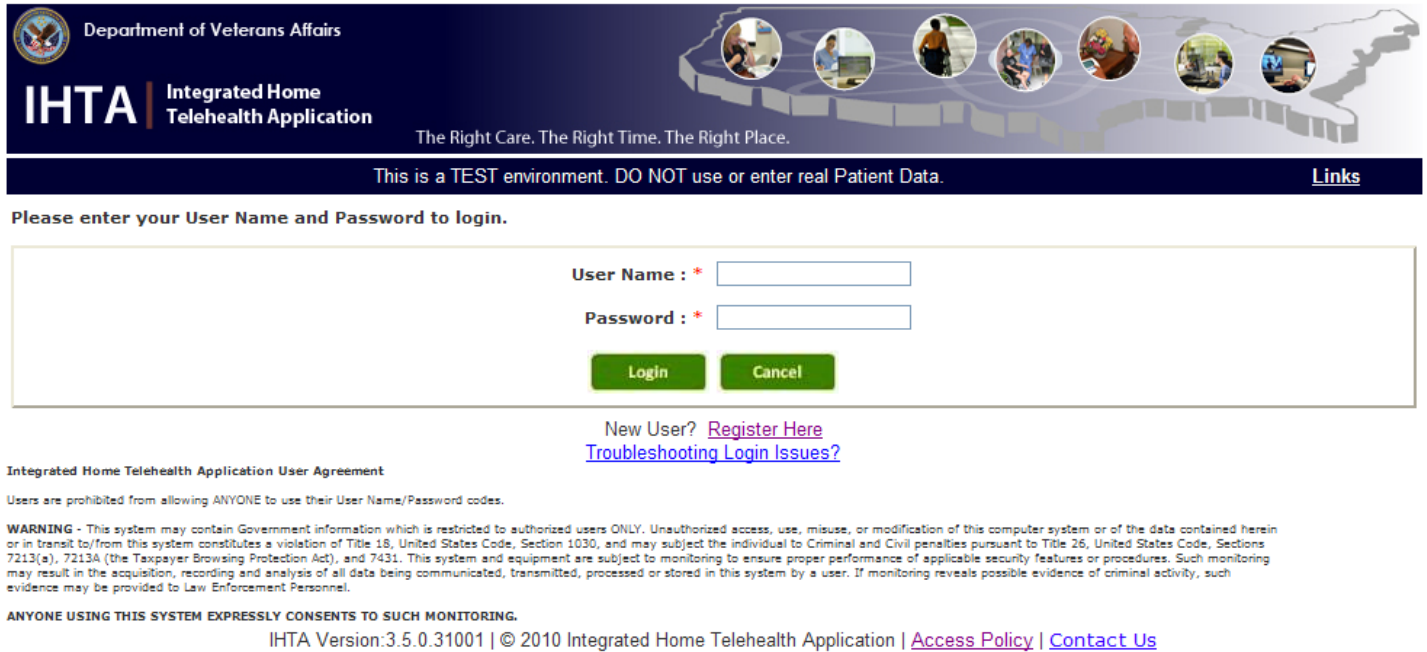
The screenshot shows the IHTA Home Page. At the top, there is a header with the Department of Veterans Affairs logo and the text "Department of Veterans Affairs". Below this, the "IHTA Integrated Home Telehealth Application" logo is displayed, along with the tagline "The Right Care. The Right Time. The Right Place." A navigation bar contains several circular icons representing different telehealth services. A dark blue banner across the top of the main content area states: "This is a TEST environment. DO NOT use or enter real Patient Data." with a "Links" button on the right. The main content area features a photograph of an elderly man in a red shirt talking on a telephone. To the right of the photo, the text reads: "Welcome to the Integrated Home Telehealth Application (IHTA). The Integrated Home Telehealth Application (IHTA) is a Web-based portal application sponsored by the Veterans Health Administration (VHA) Office of Telehealth Services (OTS). IHTA is designed to provide a flexible, maintainable, and resilient platform for Home Telehealth business functions. Each business function supported by IHTA is constructed as an application module of IHTA. Current modules include **Inventory Tracker** and **Administration**." Below this text are three numbered tabs: 1, 2, and 3. At the bottom of the page, a dark blue bar contains the text "Getting Started with IHTA: Log in/Registration". Below this bar, the text reads: "IHTA User? [Log in here](#)" and "New to IHTA? [Click here to register](#)". At the very bottom, the footer text is: "IHTA Version:3.5.0.36002 | © 2010 Integrated Home Telehealth Application | [Access Policy](#) | [Contact Us](#)".

Figure 9: IHTA Home Page



The screenshot shows the IHTA User Registration Page. It features the same header and navigation bar as Figure 9. The dark blue banner at the top of the main content area contains the text: "This is a TEST environment. DO NOT use or enter real Patient Data." with a "Links" button on the right. Below the banner, the text reads: "Please enter your VA network User Name and select your Job Description to register." The registration form consists of two fields: "User Name : \*" with a text input box, and "Job Description : \*" with a dropdown menu showing "Select". Below these fields are two green buttons: "Next" and "Cancel". At the bottom of the page, the footer text is: "IHTA Version:3.5.0.31001 | © 2010 Integrated Home Telehealth Application | [Access Policy](#) | [Contact Us](#)". A link for "Registration Help" is also present above the footer.

Figure 10: IHTA User Registration Page



The IHTA Login Page features the Department of Veterans Affairs logo and the IHTA Integrated Home Telehealth Application title. It includes a banner with a map of the United States and several circular images showing telehealth sessions. A dark blue bar contains the text "This is a TEST environment. DO NOT use or enter real Patient Data." and a "Links" button. Below this, a login form prompts the user to enter their User Name and Password, with "Login" and "Cancel" buttons. A "New User?" link points to "Register Here" and "Troubleshooting Login Issues?". A "User Agreement" section follows, including a "WARNING" about government information and a statement that users consent to monitoring. The footer shows the IHTA version (3.5.0.31001) and copyright (© 2010) along with links to "Access Policy" and "Contact Us".

Department of Veterans Affairs

**IHTA** Integrated Home Telehealth Application

The Right Care. The Right Time. The Right Place.

This is a TEST environment. DO NOT use or enter real Patient Data. [Links](#)

Please enter your User Name and Password to login.

User Name : \*

Password : \*

[Login](#) [Cancel](#)

New User? [Register Here](#)  
[Troubleshooting Login Issues?](#)

Integrated Home Telehealth Application User Agreement

Users are prohibited from allowing ANYONE to use their User Name/Password codes.

**WARNING** - This system may contain Government information which is restricted to authorized users ONLY. Unauthorized access, use, misuse, or modification of this computer system or of the data contained herein or in transit to/from this system constitutes a violation of Title 18, United States Code, Section 1030, and may subject the individual to Criminal and Civil penalties pursuant to Title 26, United States Code, Sections 7213(a), 7213A (the Taxpayer Browsing Protection Act), and 7431. This system and equipment are subject to monitoring to ensure proper performance of applicable security features or procedures. Such monitoring may result in the acquisition, recording and analysis of all data being communicated, transmitted, processed or stored in this system by a user. If monitoring reveals possible evidence of criminal activity, such evidence may be provided to Law Enforcement Personnel.

ANYONE USING THIS SYSTEM EXPRESSLY CONSENTS TO SUCH MONITORING.

IHTA Version:3.5.0.31001 | © 2010 Integrated Home Telehealth Application | [Access Policy](#) | [Contact Us](#)

Figure 11: IHTA Login Page

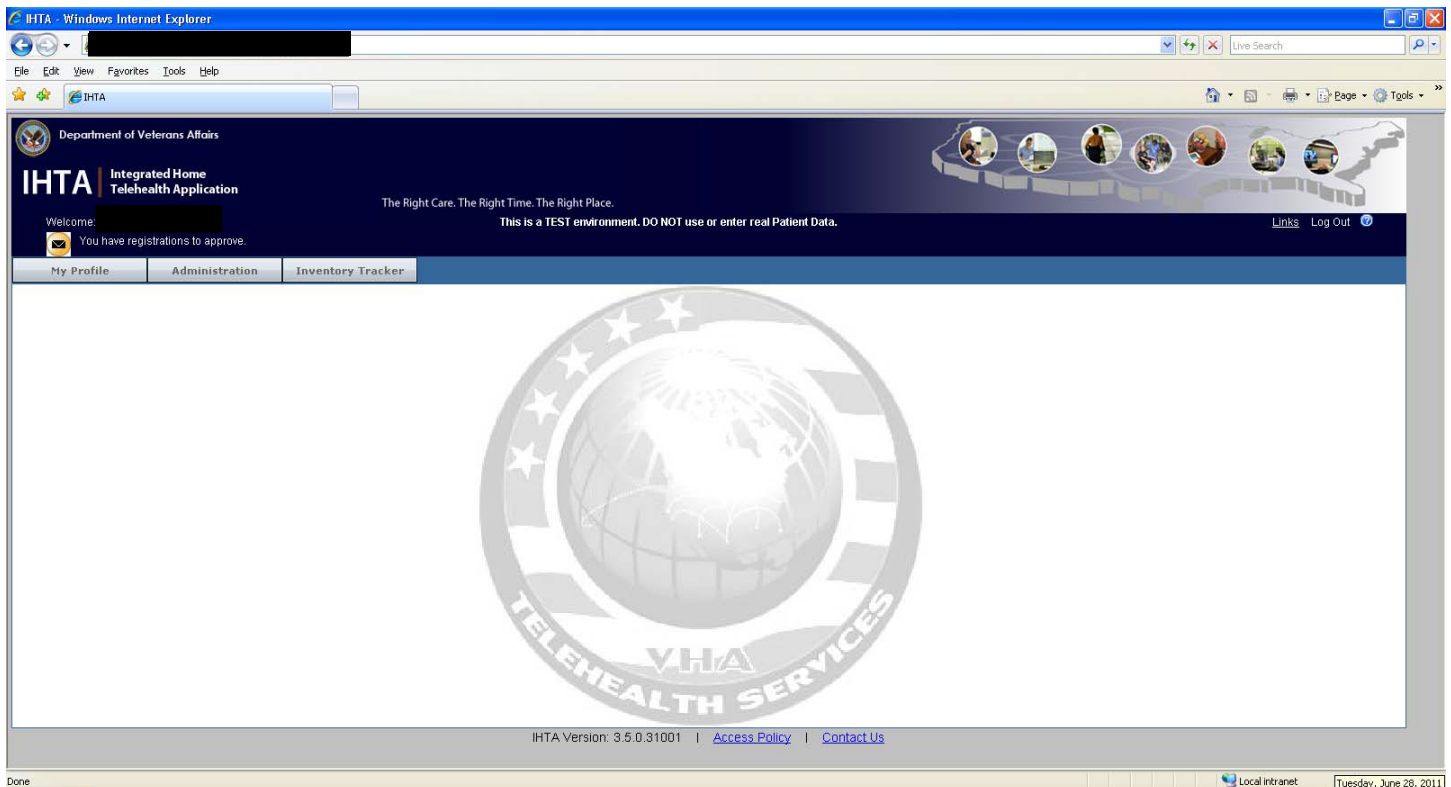


Figure 12: IHTA Portal Page



## 3.7. Access to IHTA via MHV

Only Veterans who are enrolled in the HT program and who have received an In-Person Authentication (IPA) account will be allowed to access IHTA from MHV. The following subsections outline the steps for accessing IHTA through MHV.

### 3.7.1. Enroll in Home Telehealth Program

The Care Coordinator will enroll the Veteran in the HT program in the Veterans Health Information Systems and Technology Architecture (VistA). VistA will distribute enrollment information to all necessary VA IT systems, including IHTA. Only Veterans currently enrolled in the HT program will be able to access IHTA via MHV.

### 3.7.2. In-Person Authentication (IPA)

IPA is the process of verifying a Veteran's identity by a qualified VA staff member at a VA facility. MHV matches a Veteran's first name, last name, social security number, and date of birth as registered in his/her MHV account with the same information in his/her's official VA record. This match allows MHV to request updates to a Veteran's personal health record from a VA facility. Once the Veteran has completed the IPA process, MHV disables those fields to ensure MHV data will always match the Veteran's official VA record. Once a Veteran's official VA record is changed, a Veteran's MHV account information will automatically reflect that change.

During the face-to-face validation, the Veteran is registered for MHV and provided a unique username and password. The Veteran will use this username and password for his/her initial login to MHV.

### 3.7.3. Login to MHV

1. Veterans access the Login Page by clicking on "Go to My HealtheVet – Enter Here" on the Home Page (Figure 13) or directly accessing the Login Page using the below link. Click on the links below to access the MHV Home Page and Login Page:
  - MHV Home Page: <http://www.myhealth.va.gov/>
  - MHV Login Page: <https://www.myhealth.va.gov/mhv-portal-web/anonymous.portal?nfpb=true&nfto=false&pageLabel=mhvHome>



Figure 13: My HealtheVet Home Page



The screenshot shows the My HealtheVet homepage with a navigation bar at the top containing links: HOME, PERSONAL INFORMATION, PHARMACY, RESEARCH HEALTH, GET CARE, TRACK HEALTH, and MHV COMMUNITY. Below the navigation bar are links for LEARN ABOUT, WHAT'S NEW?, and COMING SOON. The main content area is divided into three sections. The left section, titled "In the Spotlight", features a photo of a doctor and a patient, with the headline "Secure Messaging: Coming to a VA Medical Center Near You" and the date "November 2010". Below this is a paragraph: "VA often asks Veterans what they would like to see added on My HealtheVet. One of the top requests is being able to talk to". The middle section contains a list of links: Download My Data, Prescription Refill, Emergency Contacts, Providers & Physicians, Vitals & Readings, Military Health History, Medical Library, and VA Honors Veterans. The right section, titled "Member Login", contains fields for User ID and Password, a Login button, and links for Forgot User ID?, Forgot Password?, First time My HealtheVet user?, and Register today! with a REGISTER button. A red bracket on the right side of the page indicates a zoomed-in view of the login form, which is shown in a separate window on the right. This zoomed-in view shows the "Member Login" title, the User ID and Password fields, the Login button, and the links for Forgot User ID?, Forgot Password?, First time My HealtheVet user?, and Register today! with the REGISTER button.

2. Figure 14), Veterans will enter their VA User ID and Password under “Member Login” and click **Login**.

This screenshot is identical to the one above, showing the My HealtheVet homepage with the same navigation bar, main content sections, and a callout showing a zoomed-in view of the login form. The zoomed-in view shows the "Member Login" title, the User ID and Password fields, the Login button, and the links for Forgot User ID?, Forgot Password?, First time My HealtheVet user?, and Register today! with the REGISTER button.

Figure 14: My HealtheVet Login Page





### 3.7.4. Access IHTA via MHV

1. Once logged in to MHV, Veterans will access IHTA by clicking on the **Questions and Vitals** tab from the MHV Portal Page (Figure 15).
2. The IHTA Portal Page will display (Figure 16).

Note that the figures below are provided for illustration purposes only. The detailed technical requirements and the user interface of the MHV/IHTA integration will be defined and developed according to the specifications outlined by OTS and the MHV Team.

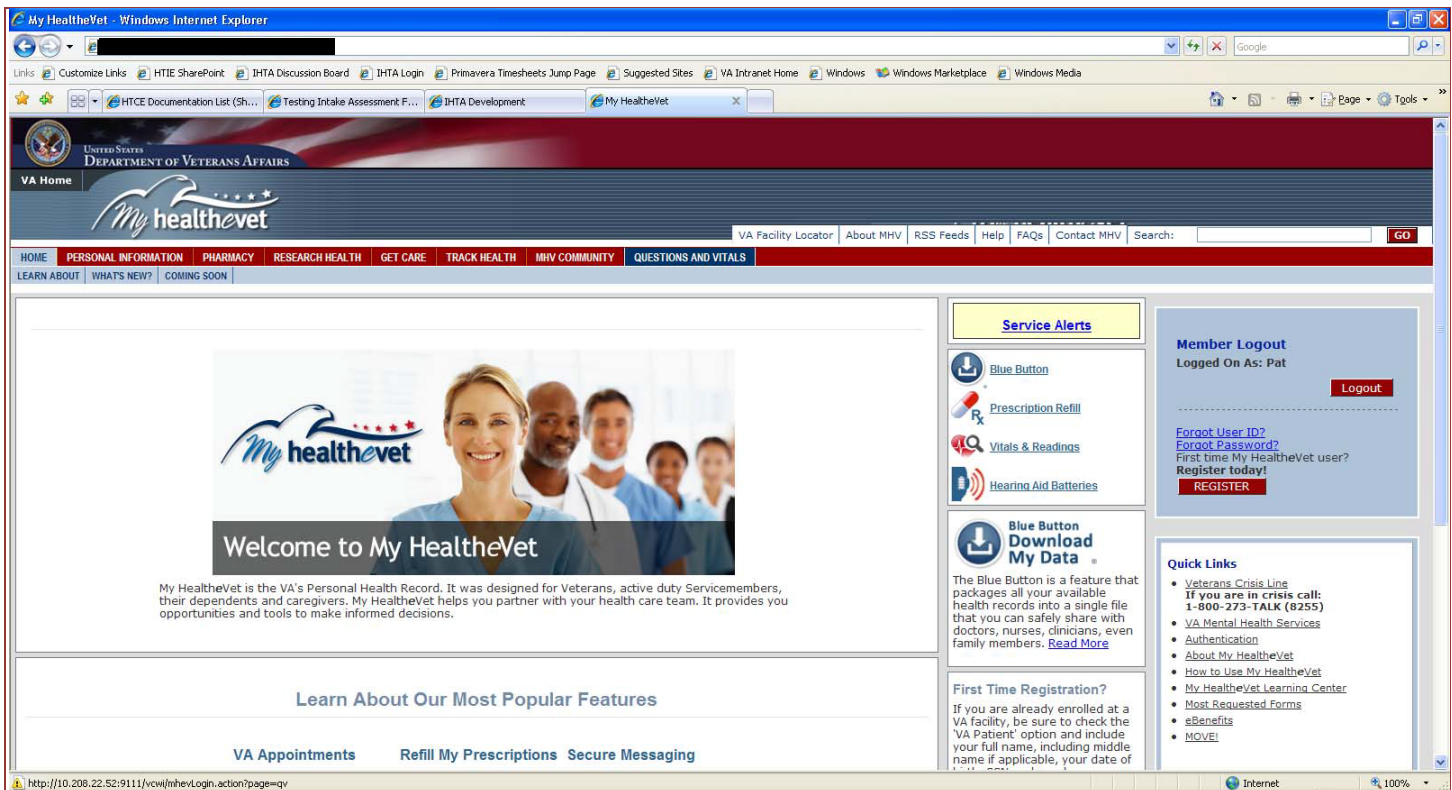


Figure 15: IHTA Access from My HealtheVet Portal Page

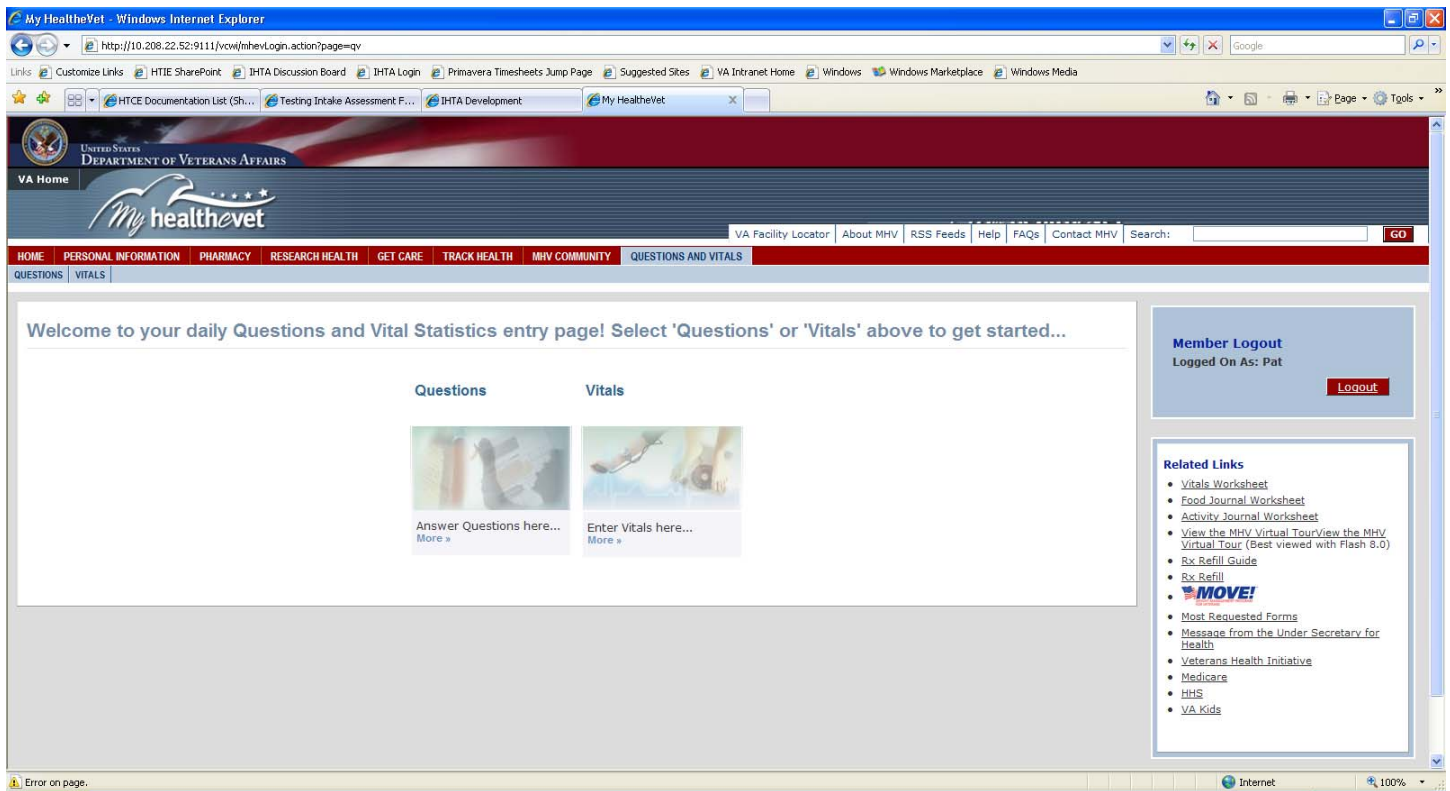


Figure 16: Veteran's Common Web Interface Module (IHTA/MHV Integration)



## 4. Process View

This section focuses on various processing constructs within the IHTA architecture.

### 4.1. Enterprise Context

Figure 17 depicts enterprise systems that IHTA will interface with. The details of the enterprise services and applications are summarized in Table 1.

**NOTE:** The CCWI and the VCWI will be delivered in subsequent releases.

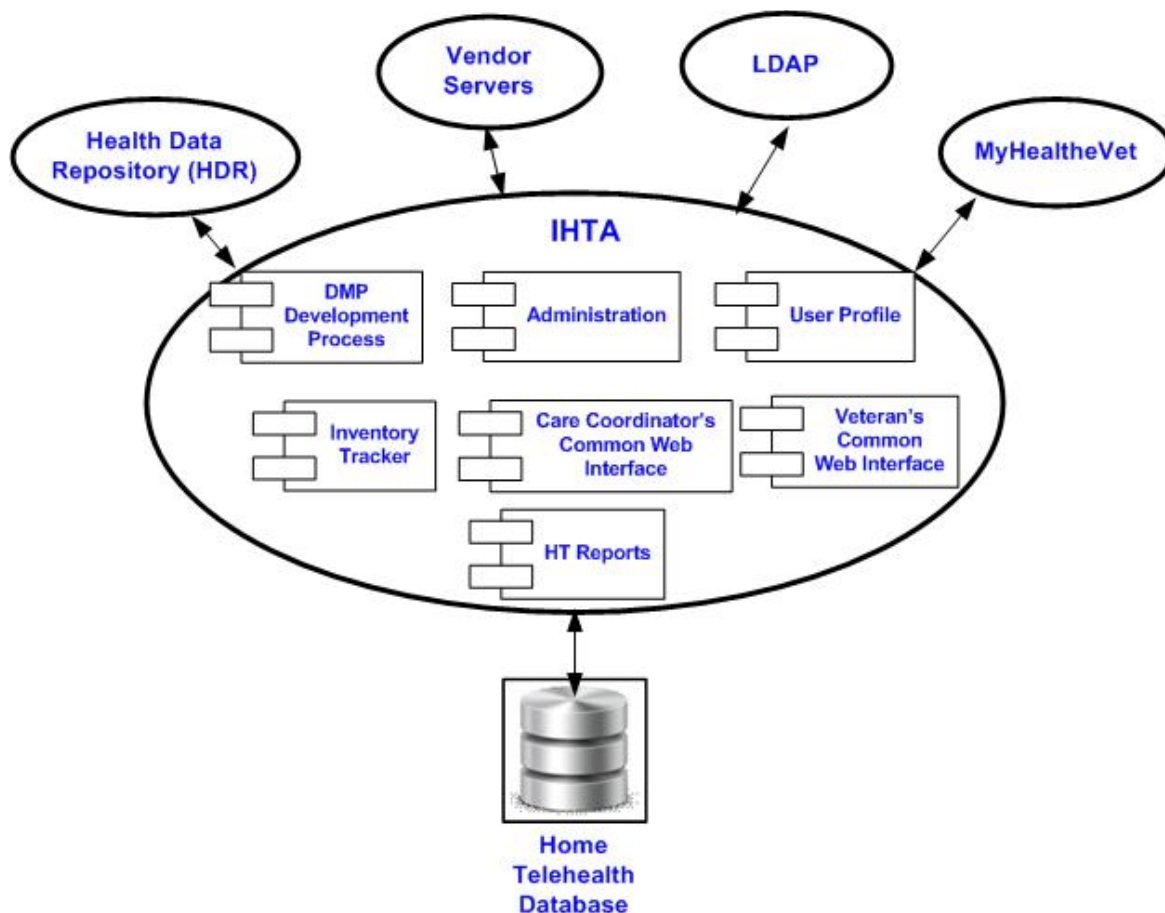


Figure 17: Enterprise Context

Table 1: Enterprise Service and Application Summary

Service	Category	Integration Technology	Notes
HDR	Enterprise	HL7 messaging	None
Vendor Servers	Enterprise	Indirect integration through HT database	None
Active Directory	Enterprise	Spring LDAP	Authentication and Authorization
MHV	Enterprise	Portal Server	Host HTCE as a Java Portlet



Service	Category	Integration Technology	Notes
HT Database	Internal	Java Persistence Application Programming Interface (API)	Database for all of HT

## 4.2. Enterprise System Integration

This section describes the enterprise systems that will integrate with IHTA.

### 4.2.1. Vendor Server

The Inventory Tracker module of IHTA contains a logical grouping of business processes for tracking patient device inventory. Vendor Servers send an HL7 message containing device inventory information formatted in XML format to the HT HL7 Census Server for processing. The HT HL7 Census Server extracts inventory information from the OBX segment of the HL7 message, persists the data to the HT database, and sends an acknowledge HL7 message to the Vendor Servers. When IHTA users access the Inventory Tracker module and query for a device inventory report, IHTA uses the Java Persistent API (JPA) to query the HT database for inventory data that was previously inserted into the HT database by the HT HL7 Census Server. Figure 18 depicts the process flow of IHTA integration with vendor servers and HT HL7 Census Server.

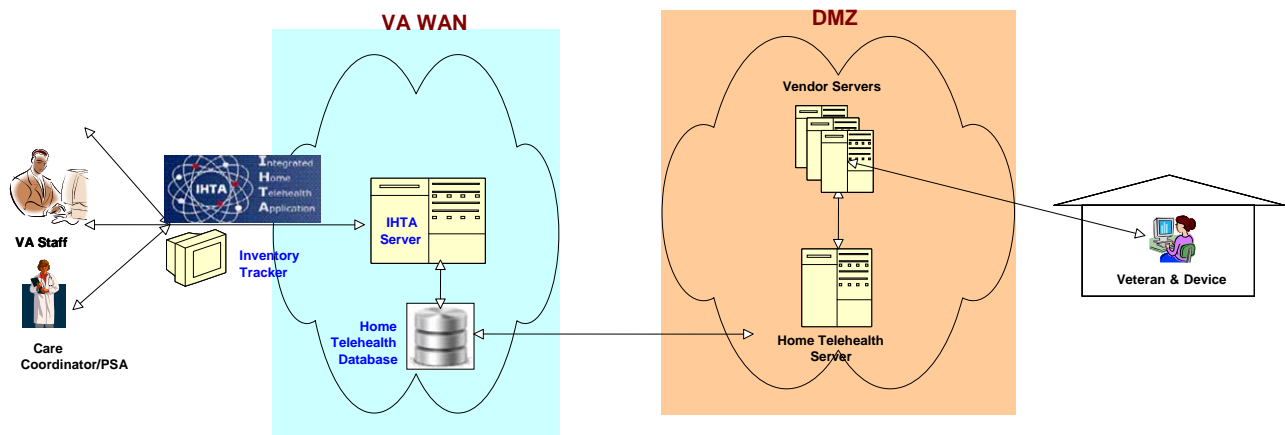


Figure 18: Vendor Server - Inventory Tracker

### 4.2.2. Home Telehealth Database and LDAP

IHTA interfaces with the HT Database using the JPA to perform database-related operations. All database-related operations are centralized in the IHTA persistent classes. IHTA leverages the CISS LDAP Security service to authenticate VA users. Figure 19 depicts the interaction between IHTA, the HT database, and the VA Enterprise LDAP.



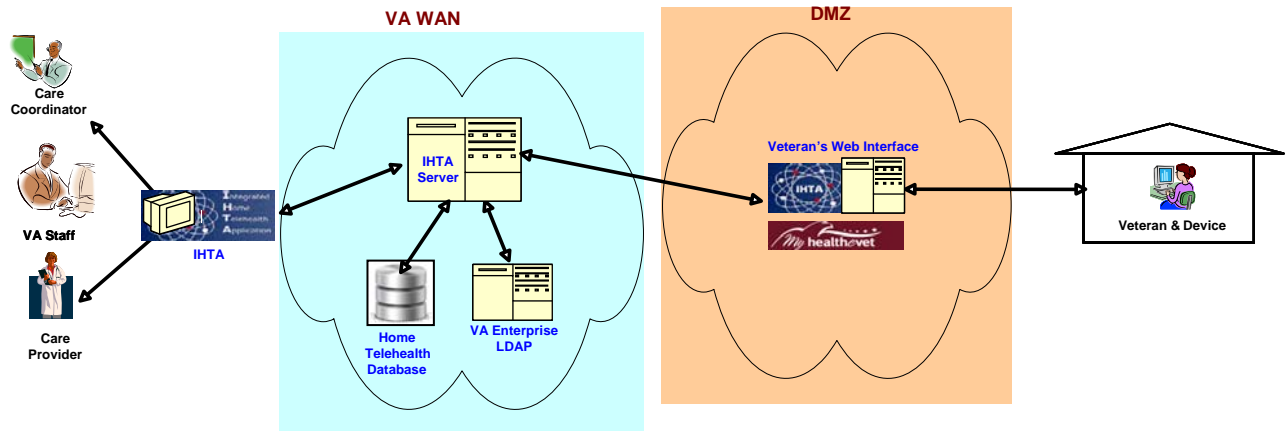


Figure 19: Home Telehealth Database and the VA Enterprise LDAP

### 4.2.3. My HealtheVet

The interaction between IHTA and MHV (the VCWI module) will be facilitated in the External Services component of IHTA. IHTA external services are implemented using Web Service technology. IHTA Web Services are defined in the Web Service Definition Language (WSDL) file and are hosted on the IHTA Portal Server. Veterans will access IHTA through MHV. To gain access to IHTA through MHV, a Veteran must be enrolled in the HT Program and have an IPA account. (Part of the registration process for MHV is to invoke an IHTA Web service to request access to IHTA.) When a Veteran's registration for access to MHV is in-person authenticated and approved, they will be eligible for access to IHTA if they are approved and enrolled in an appropriate HT program. When a Veteran logs in to MHV, MHV will invoke an IHTA Web service to authenticate a Veteran for access to the Veteran's Interface module of IHTA. Further details on the Veteran's Interface module will be provided as business needs are defined. Figure 20 depicts the interaction between MHV and IHTA.

**NOTE:** The VCWI will be delivered in subsequent releases pending a customer decision to move forward.

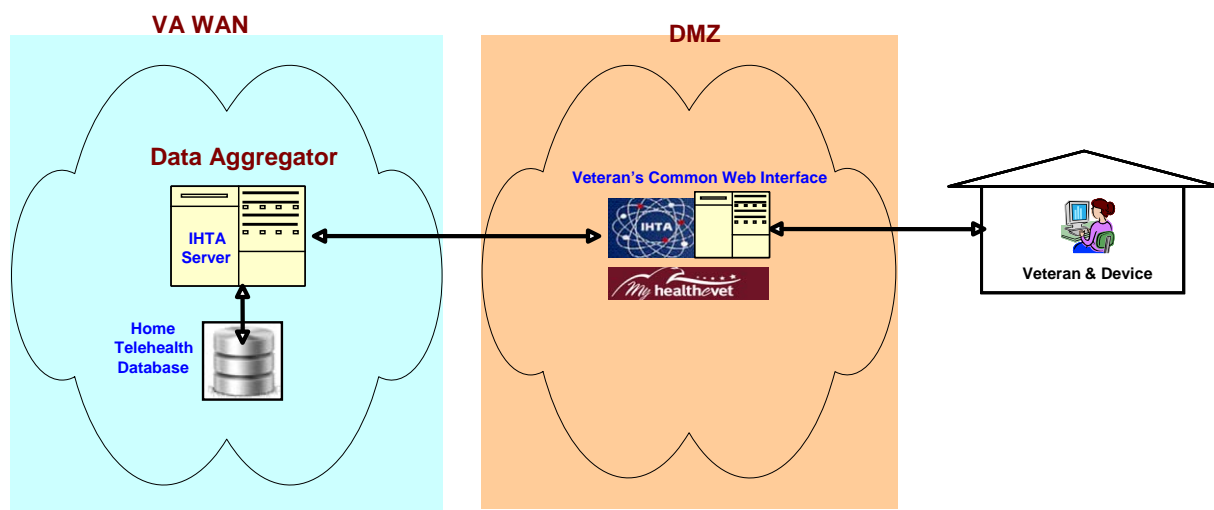


Figure 20: My HealtheVet and IHTA



#### 4.2.4. Common Data Aggregator

The Common Data Aggregator (CDA) component of the IHTA architecture will:

- Provide a well-defined, centralized, system-to-system interface for the IHTA CCWI, VCWI, Vendor Servers, and HDR to query HT data from and to upload HT data to the HT database.
- Leverage existing proven technologies, such as messaging and Web Services to provide a platform and system independent implementation.
- Minimize the impact of changes to external systems, such as Vendor Servers, HDR, and MHV, to IHTA.

Figure 21 illustrates typical business cases of the CDA.

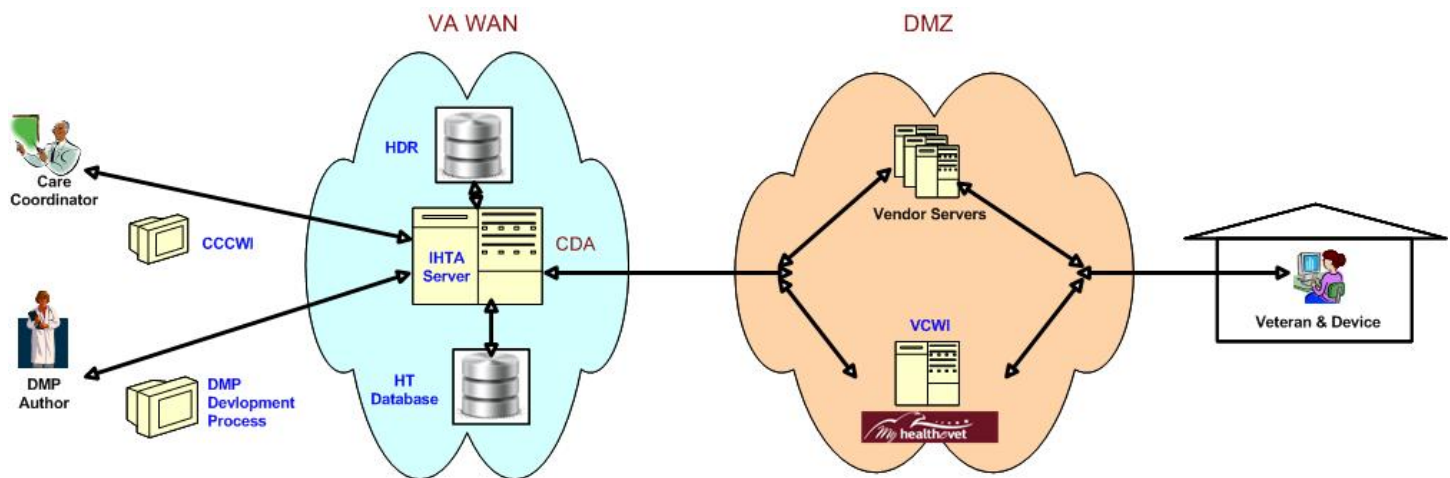


Figure 21: Common Data Aggregator

DMP Authors will use the DMP Development Tool to create and to update DMPs, which will be uploaded to the HT database. The Vendor Servers will interface with the CDA to upload vital sign, survey, and DMP data entered by Veterans to the HT database. Care Coordinators will use the CCCWI to designate DMPs for specific Veterans. These DMPs will be pushed to the Vendor Servers and IVR, and displayed on the VCWI, via the CDA. Veteran vital sign, survey, and DMP response data will also be collected from the HDR. Future architecture will also allow veterans to gain access to HT data through MHV and the VCWI module, which interfaces with the CDA. Vital sign, survey, and DMP data entered by the Veteran through the VCWI will be uploaded to the CDA for consolidation with data entered from the other Veteran communication channels.

#### 4.3. Portlet Integration

IHTA will be implemented as a Web-based “Integrated Portal”. To be compliant with other MHV applications, IHTA will be implemented in Java using both standard and enterprise features. MHV applications will integrate to IHTA using the Java Portlet API (to include industry standard protocols, such as Java Specification Request (JSR)-168, JSR-286, and Web Services for Remote Portlets (WSRP) 2.0). These standards dictate “java portal and portlet” behavior. MHV applications that do not integrate to these standards (e.g., legacy applications in a different language, etc.) may still be able to be integrated into IHTA, in a looser fashion via a URL.



At the current time, the IHTA architecture calls for IHTA to be implemented and deployed as a portlet on its own dedicated portal server. If the business requirements call for IHTA to be deployed onto the MHV portal system, IHTA will be packaged as a portlet and deployed onto the MHV portal server.

## **4.4. Service Oriented Architecture**

When deemed necessary, and based on business requirements, IHTA will expose its own business services to the VA Enterprise as Service Oriented Architecture (SOA) services. These services will be implemented using industry standard technology as either 1) a remote procedure call using Stateless Session Enterprise Java Beans (EJB) or 2) a Web service implemented in the Representational State Transfer (REST) style.

## **4.5. Event-Driven**

### **4.5.1. External**

If IHTA business requirements call for consumption of various external events, such as HL7 message formats, IHTA will implement a messaging module utilizing Java Messaging Service (JMS) queues to handle specific external events. The loose coupling between IHTA and external systems preserves the principle of “isolation” in the IHTA framework’s I3-Architecture.

### **4.5.2. Internal**

In addition, internal events will be processed asynchronously to increase performance. All internal events will be logged, will exhibit retry characteristics, be race-condition exempt, and be distributed across the target logical cluster. The Spring JMS framework is used to publish internal JMS messages. EJBs as Message Driven Beans will be used to publish and subscribe to JMS messages. The publishing of JMS messages to the target JMS consumers will be configured in a Spring context file with a fallback strategy in a separate cluster of JMS servers in cases where the target cluster is down.

## **4.6. Batch Processing**

When necessary, IHTA will use Java Batch services that encapsulate the implementation of a Spring Batch Library. IHTA batch services will support one-time and recurring batch processing of business tasks that are called for in the IHTA business requirements.

## **4.7. Business Intelligence**

Complex business processing requires that various business intelligence features be accommodated in the architecture. These business intelligence features can be categorized into two distinct areas: 1) Rules and 2) Workflow.

### **4.7.1. Rules Processing**

The IHTA architecture leverages the Java Rule libraries that offer interface support for the incorporation of a JSE 94-compliant Business Rules Management System (BRMS). The BRMS implementation, to be leveraged by IHTA, are JBoss Drools<sup>®</sup> Rule Engine and IBM ILOG<sup>®</sup> Rule Engine. When necessary, business logic will be extracted from Java services and represented as business rules deployed on BRMS. Internal Java components of IHTA will interface with BRMS through a centralized Java interface.



## 4.7.2. Workflow Processing

The IHTA framework offers interface support for the incorporation of a Business Process Management (BPM) capability. The BPM implementation supported in the IHTA architecture is the JBoss jBPM® Workflow Engine (using the Java Process Definition Language [jPDL] process language).

## 4.8. Logging

The IHTA framework provides logging support through the logging utility for the Java (Log4J) library and the Spring Aspect-Oriented Programming (AOP) interceptor encapsulated in a Java Logging service. IHTA will configure its own Log4J logging file and will use the Java Logging Service for all logging purposes.

## 4.9. Exception Management

The IHTA framework provides exception management support through a Java Exception Management service that can be configured to: 1) log, 2) log and persist locally, and/or 3) forward the exception to the Common Service Enhanced Event Logging System (EELS) in future releases. The persistence of an exception is critical as it allows end users to report an exception ID to the Help Desk. Warranty support can then look up the technical details of the exception in the database. The IHTA architecture uses the Exception Management service support provided in its framework.

## 4.10. Session Management

The IHTA architecture handles session management through a Java service defined in a context file of the Spring library. This Spring-managed service is the singular component that knows the context (user role) and abstracts the underlying storage mechanism stored in either an Http Session, Flex Session (if applicable), or a database.

# 5. Deployment View

This section focuses on the application topology and load balancing, along with hardware/software solutions and data center configurations.

## 5.1. Application Topology

IHTA will be deployed onto its own “domain”, which is built to scale horizontally as throughput increases. The domain and other widely supported concepts across the major Java Enterprise Edition (JEE) application/portal servers are described in Table 2.

**Table 2: Application Topology**

Concept	Description
Domain	Consists of one or more servers that work together
Admin Server	The administrative controller of the domain
Server Instances	Server processes that handle application requests
Cluster	Groups server instances together to provide a distributed mechanism and fail-over for handling application requests

The IHTA domain will be configured as a Weblogic Portal domain for the Weblogic Portal Server. Figure 22 illustrates, at a high level, major components configured in the IHTA domain.

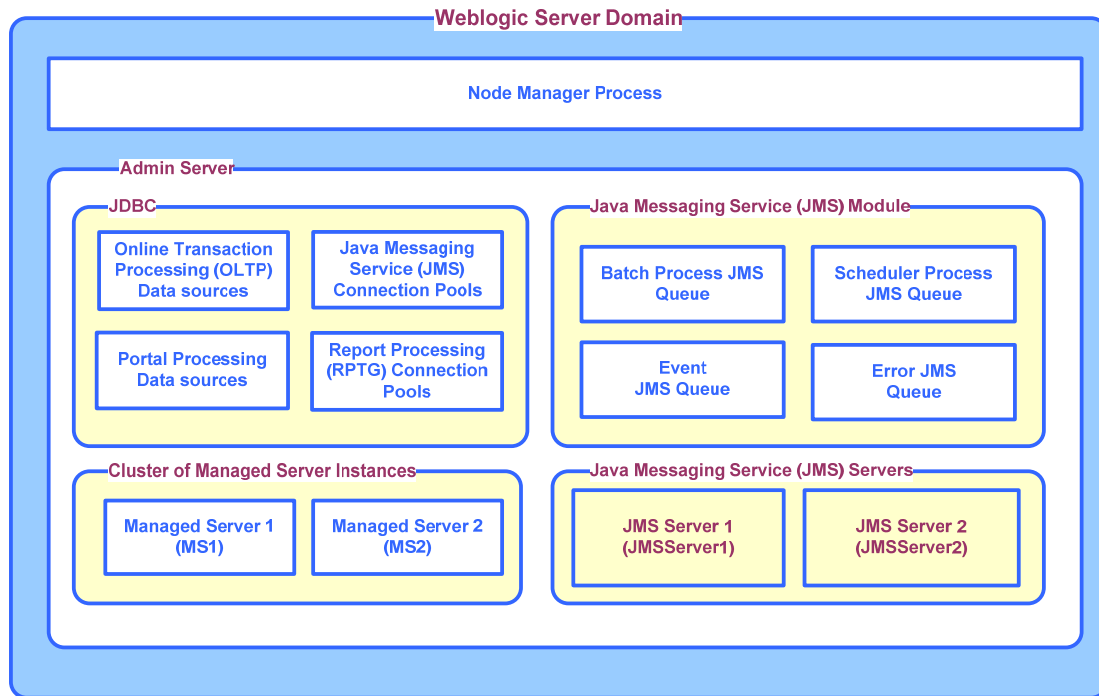


Figure 22: Application Topology

## 5.2. Physical Topology

The physical topology of the IHTA environments will evolve over time. Figure 23 illustrate a high-level overview of the IHTA environments.

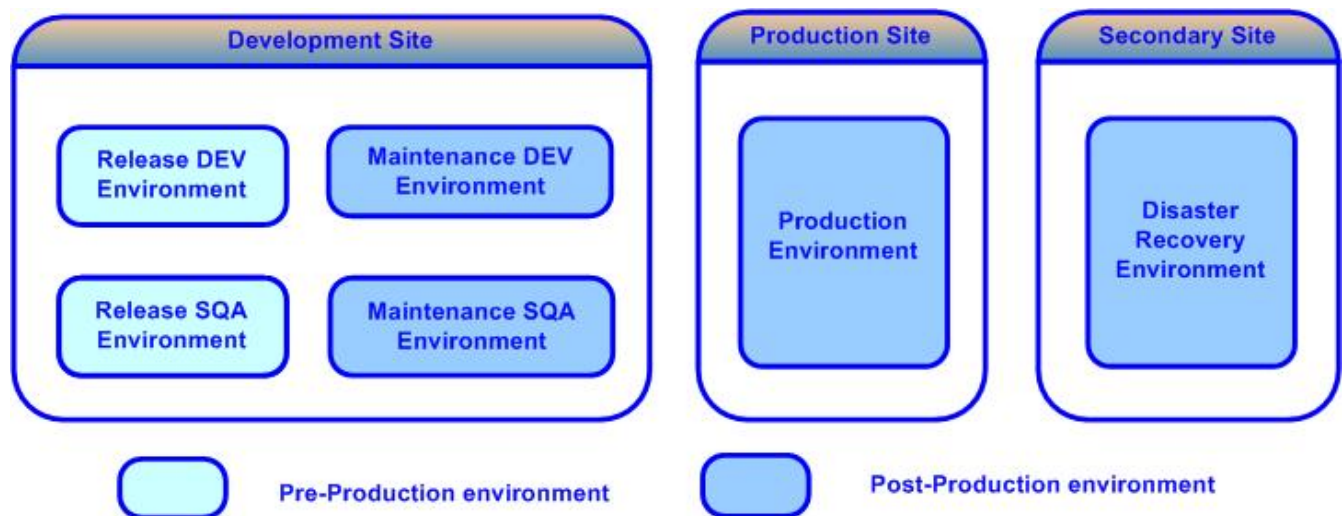


Figure 23: Environment Overview



To support both future development and production maintenance, IHTA environments are defined as the following:

- Release Development Environment: Support development of ongoing IHTA releases.
- Release SQA Environment: Support testing of ongoing IHTA releases.
- Maintenance Development Environment: Support maintenance and defect fixes for IHTA production.
- Maintenance SQA Environment: Support maintenance and testing of defect fixes for IHTA production.
- Production Environment: Host IHTA production version.
- Disaster Recovery Environment: Back-up environment for IHTA production environment.

The following figures illustrate a high-level configuration for the Web server, application server, and database server in each IHTA environment.

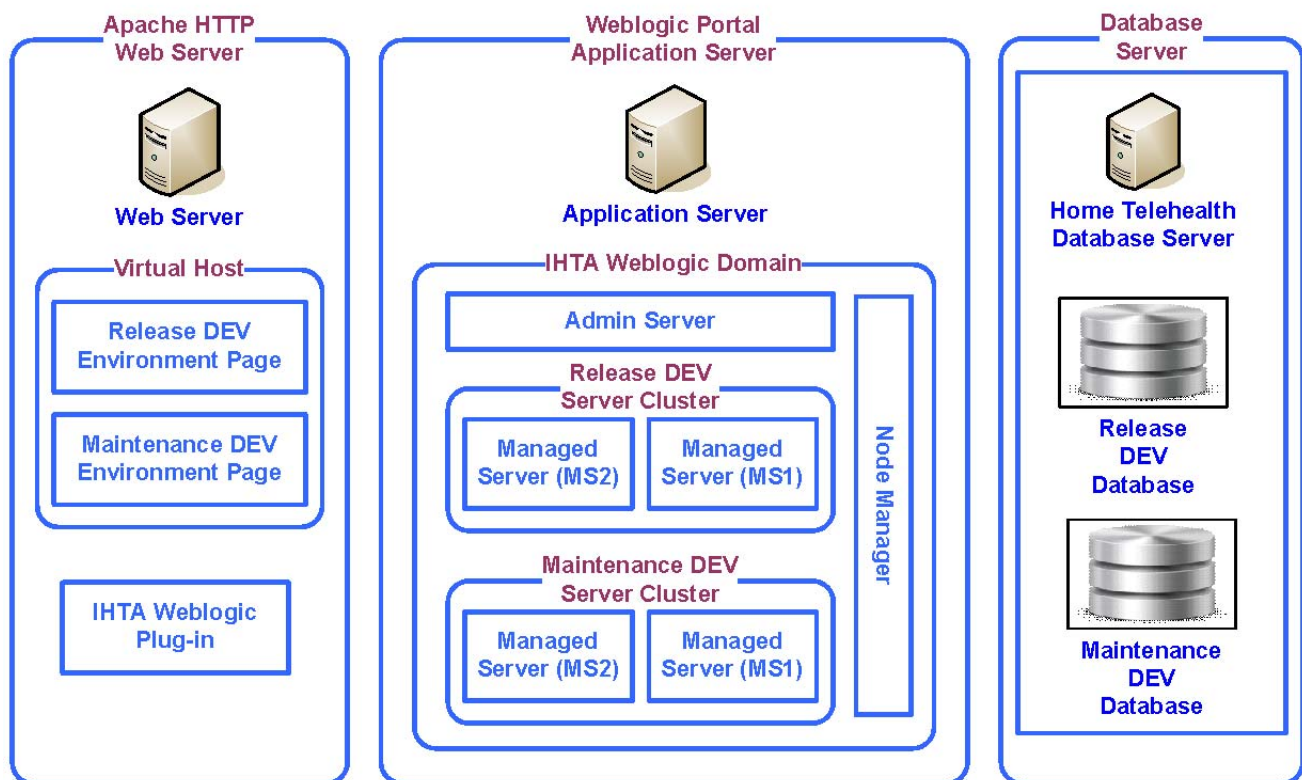


Figure 24: Development Environment



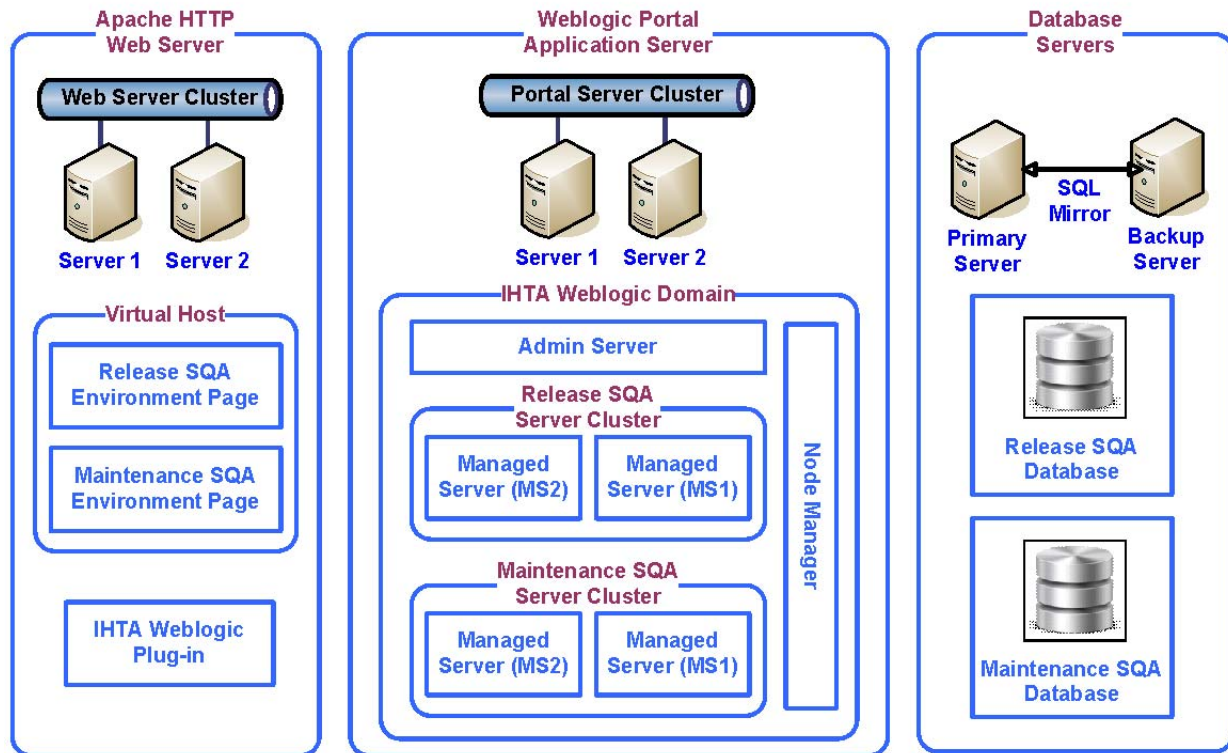


Figure 25: SQA Environment

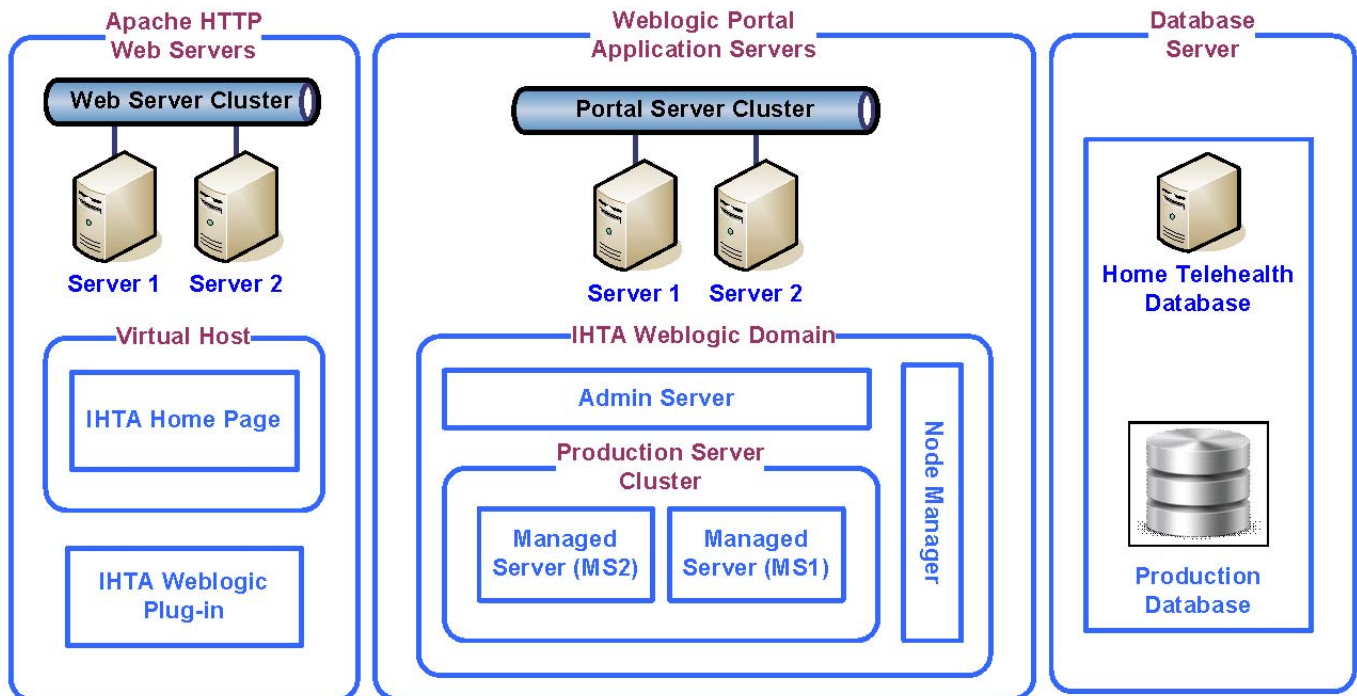
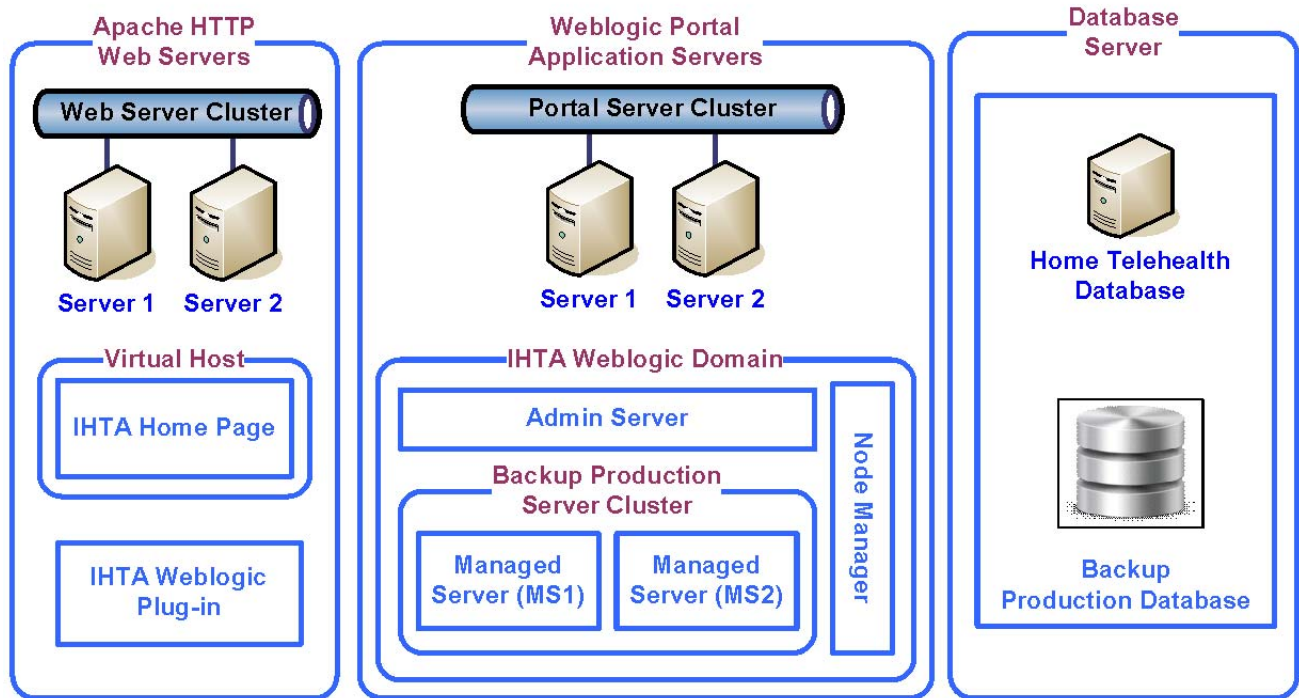


Figure 26: Production Environment



**Figure 27: Disaster Recovery Environment**

While IHTA development and SQA environments will be hosted at the Hines Data Center, the IHTA production deployment will be at Martinsburg as the primary data center and Hines as the back-up data center. The following figures depict IHTA environments and the high-level cluster configuration of each environment.



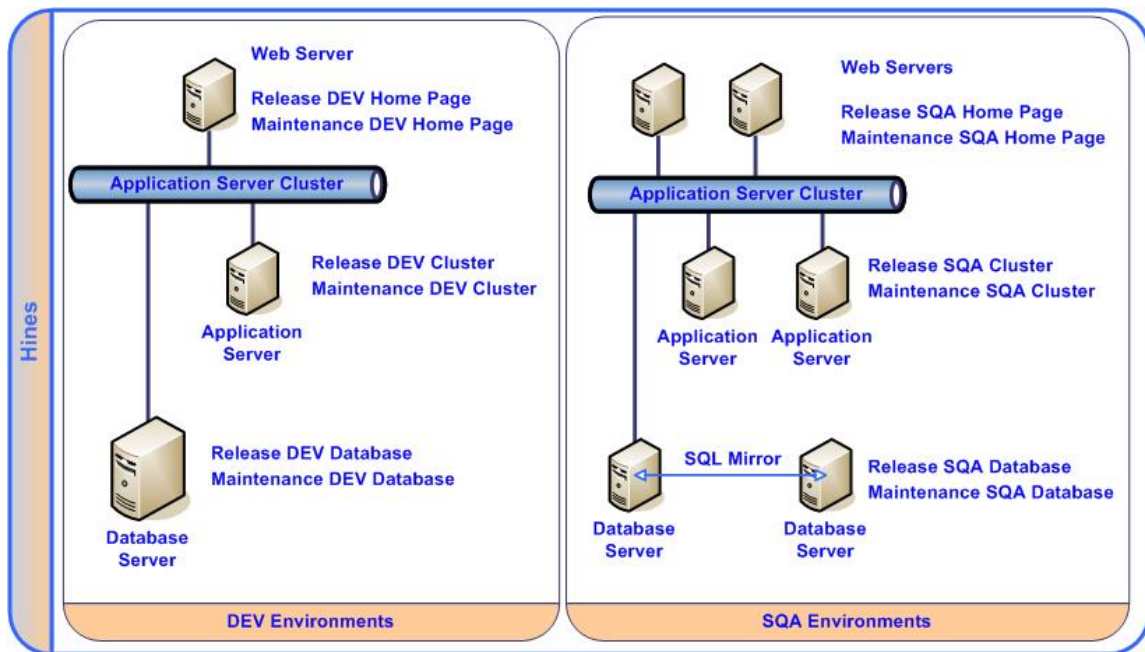


Figure 28: Development Data Center

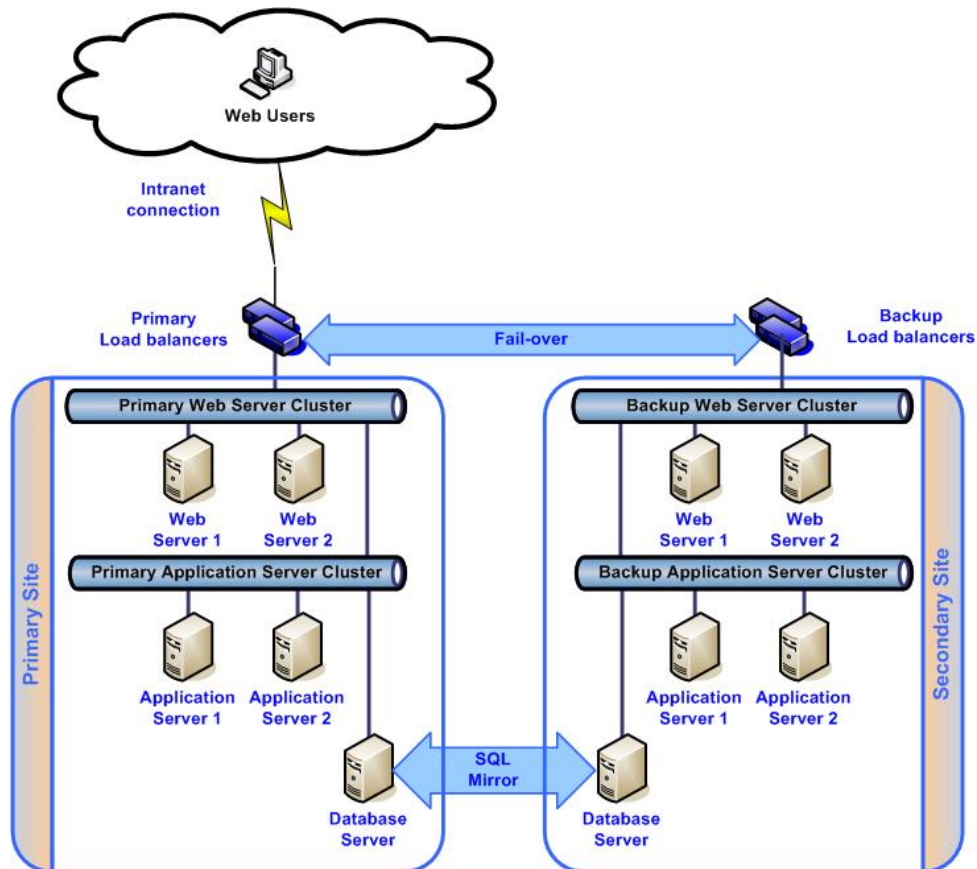


Figure 29: Production Data Centers



## 6. Implementation View

This section contains details about the architectural design and the technology stack envisioned for IHTA.

### 6.1. Application Layering

Application layering generalizes the various functional layers in the architecture (see Figure 30). For IHTA, its HTML-rendered content implements the standard Struts2 Web framework, injected with Spring components called business services. IHTA uses the Flex library to render its content and HTTP requests are tunneled through a servlet (BlazeDS) connected to a Spring controller. The Spring controller will then interact with a Spring business service, rules engine, workflow engine, and JPA persistent component.

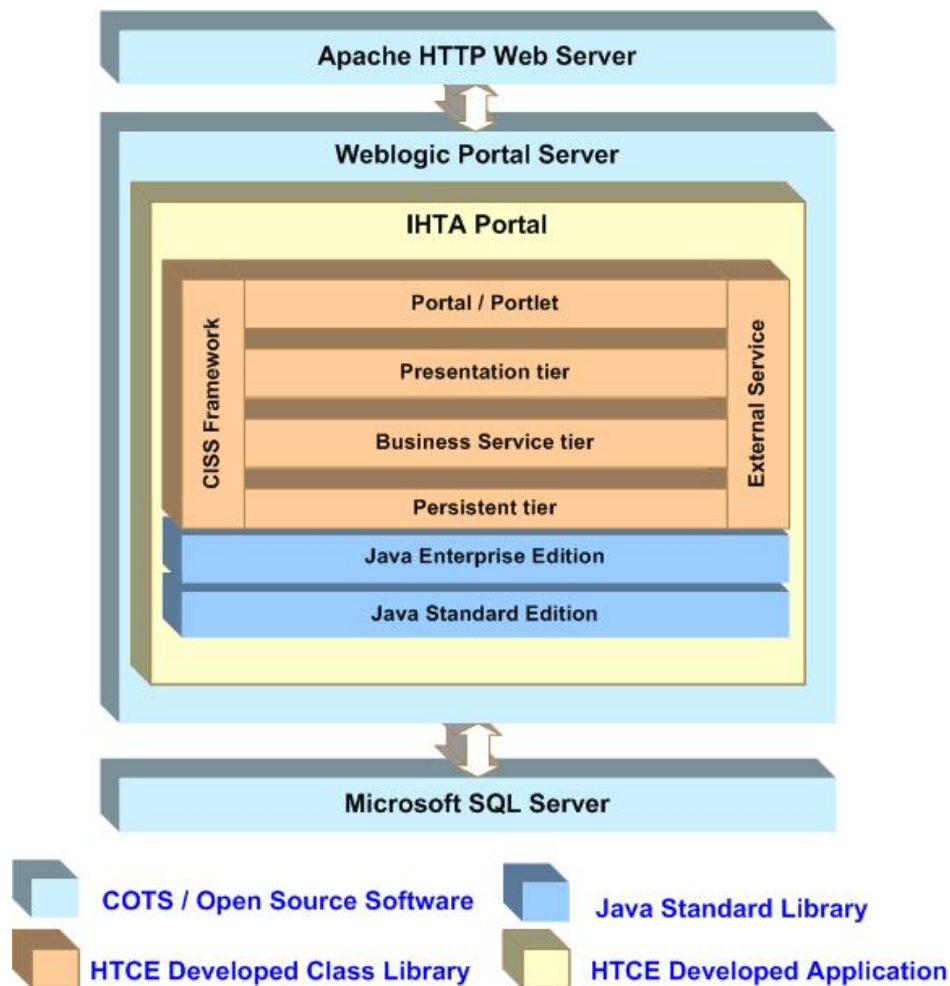


Figure 30: IHTA Architectural Layers



## 6.2. Technology Stack

Figure 31 identifies and groups core IHTA technologies. These technology choices are also covered in Section 6.6.

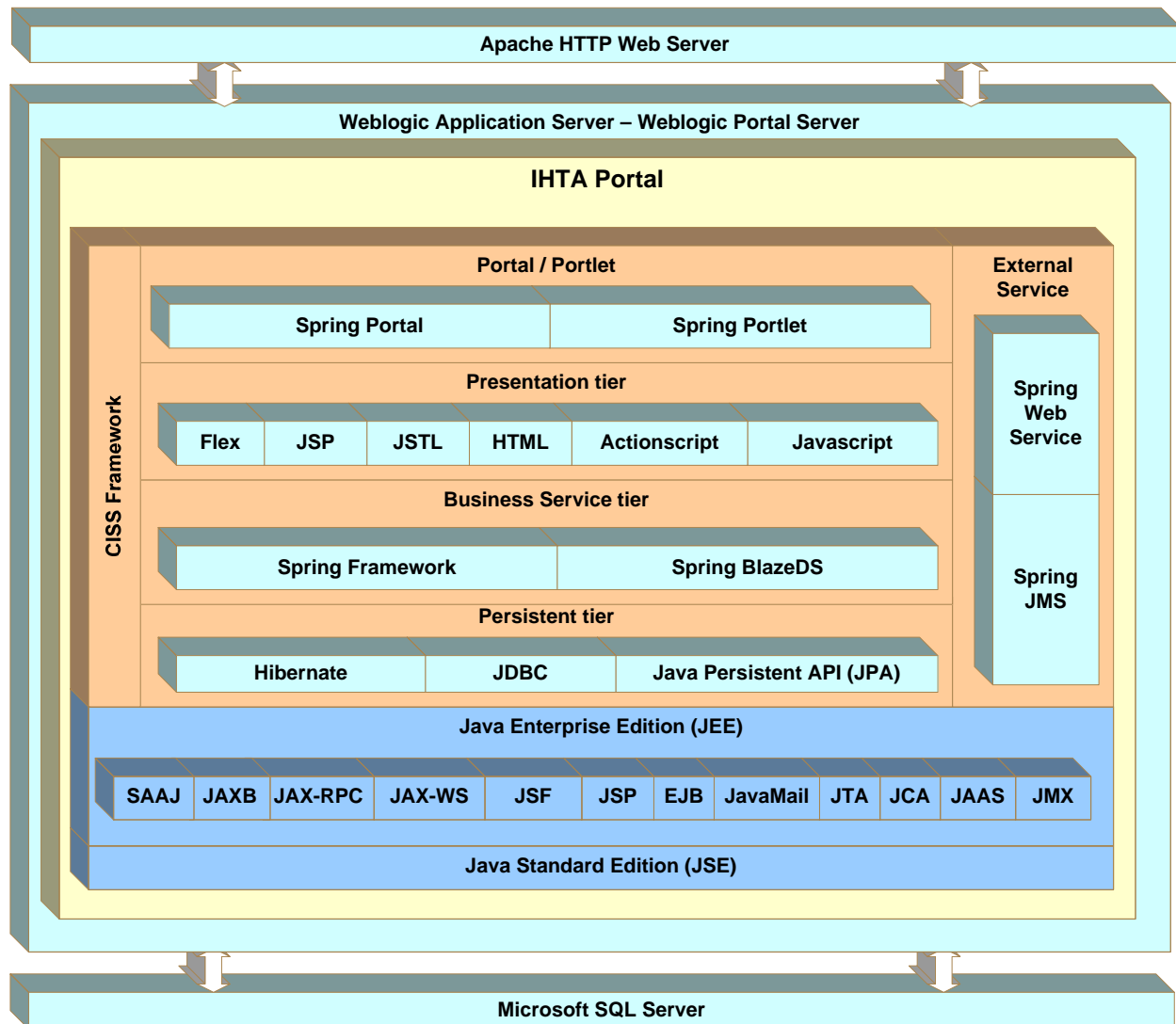


Figure 31: IHTA Technology Stack

## 6.3. Application Components

IHTA modules represent a logical grouping of Java classes and components that are implemented to perform the same or similar business functions. IHTA module codebase uses the IHTA common codebase to ensure a consistent UI, well-defined business entities through domain classes, and centralized business logic defined in business services. The following figure depicts IHTA modules and components. Note that, until business requirements are further defined, Wound Care and IVR are considered as possible sub-categories of the Care Coordinator's Interface module. They are not currently envisioned as major IHTA modules.

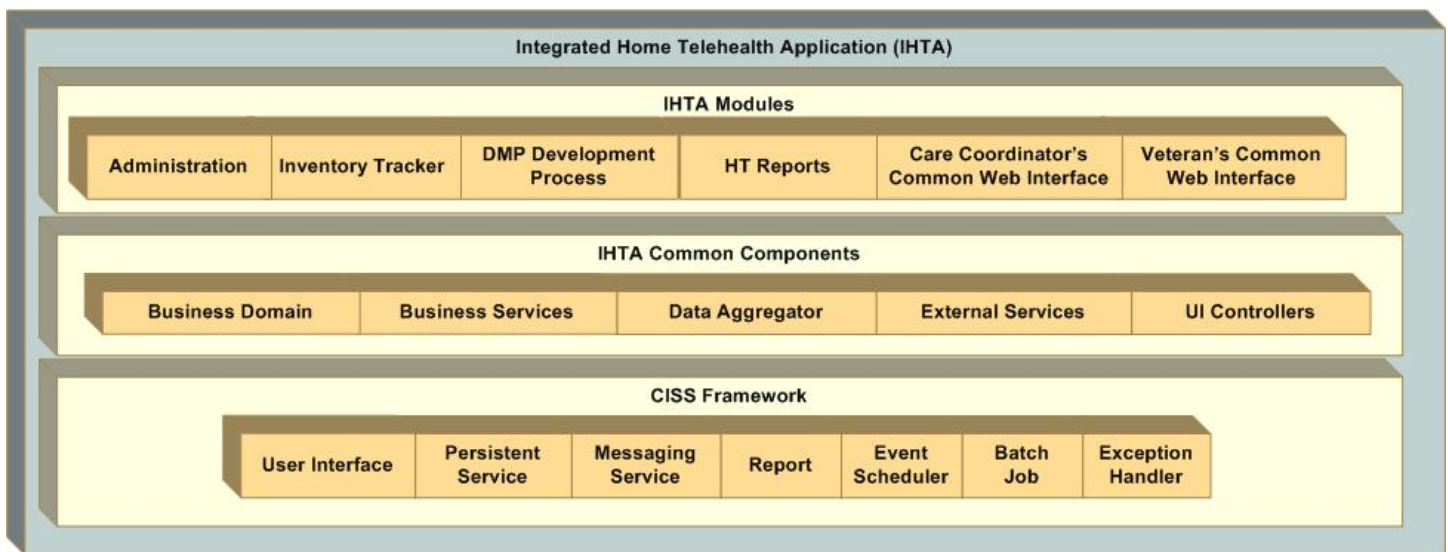


Figure 32: IHTA Application Components

**NOTE:** The Care Coordinator's Common Web Interface and the Veteran's Common Web Interface will be delivered in subsequent releases.

## 6.4. Application Context

IHTA leverages the Spring framework to manage the configuration of services, injection of dependencies, and enforcement of the DRY principle (Don't Repeat Yourself). IHTA provides an extendable Spring Application Context that can be defined as follows:

- **IHTAApplicationContext.xml:** A main application context that contains Spring context files that are specific to each functional component, such as common, persistence, service, and data access object (DAO).
- **IHTACommon.xml:** A context file containing definitions of all common components in IHTA.
- **IHTAPersistent.xml:** A context file containing definitions of Spring managed services that are responsible for data persistence.
- **IHTAService.xml:** A context file containing definitions of business services in IHTA.
- **IHTADAO.xml:** A context file containing definitions of DAO classes.

## 6.5. Test Scaffolding

The IHTA architecture provides an abstraction around TestNG and JUnit classes in its framework. Test components are injected via annotations (the only place where annotations are used).



## 6.6. Development Software

Table 3 lists the development software and tools for IHTA.

**Table 3: IHTA Development Software and Tools**

Category	Library/Tool	Estimated Cost	Waiver
<b>Continuous Integration (CI)</b>			
CI server	Build Forge	None (Enterprise License)	N/A
Automated Testing	TestNG	None (Open Source)	N/A
Build	Maven	None (Open Source)	N/A
Content Management System for Team Collaboration	SharePoint	None (Enterprise License)	N/A
Agile Scrum Support	VersionOne	Yes - \$2k	N/A
<b>Integrated Development Environment (IDE)</b>			
IDE	Eclipse 3.3 with FlexBuilder Professional	Yes – \$749 per UI developer	N/A
Database Access	Toad	Yes – \$650 per tester/developer	N/A
Configuration Management Plug-in	Rational ClearCase Remote Client	None (Enterprise License)	N/A
<b>Application Server</b>			
Portal/JEE Server	BEA Weblogic Portal 10.2	None (Enterprise License)	N/A
Operating System	Microsoft Windows 2003 Server (for Windows specific processes)	None (Enterprise License)	N/A
Operating System	RedHat Linux	Yes (TBD)	N/A
<b>Web Server</b>			
Web Server	Apache Http Server	None (Open Source)	N/A
<b>Application Monitoring</b>			
Health	BMC Patrol/JProbe	Yes (TBD)	N/A
Log Viewing	Depends on data center support	Yes (TBD)	N/A
Logging	Log4j	None (open source)	N/A
<b>Database</b>			
Database Engine	Microsoft SQL Server 2008 (Oracle was too costly)	None (Enterprise License)	N/A
Database Monitoring	Erwin/TOAD	Yes – \$650 per tester/developer	N/A
Database Modeling	Rational/Data Architect	None (Enterprise License)	N/A
<b>Configuration Management</b>			
Issue Tracking	Rational ClearQuest	None (Enterprise License)	N/A
Source Code Control	Rational ClearCase	None (Enterprise License)	N/A
<b>Requirements Management</b>			
Sprint Content	VersionOne	Yes (\$2K)	N/A



## 7. Data View

IHTA will require a database to support its persistence needs. This database will contain logically separate datasets serving the portal, customization, and reference data. For performance reasons, the IHTA domain will be configured with two, multi-data sources. One is for online transaction processing and the other is for reporting. A multi-data source is a logical data source component that contains a collection of physical data sources to the underlying databases for fail-over and load balancing purposes. From the IHTA level, logically separate schemas exist for the following domain areas: IHTA portal data, IHTA application data, and HT data. Further details about the different components of the IHTA data view are defined in the *HTCE Database Design Document* located in the HTCE TSPR.

## 8. Quality

IHTA Java classes make use of base classes implemented in the CISS framework to ensure reusability and cohesiveness. As business requirements call for additional functionalities that are not currently supported by technical components provided by base classes in the CISS framework, IHTA Java classes and technical components can extend from the CISS framework base classes to provide add-on functionalities and extensibility where applicable.

IHTA Software Architecture comprises commonly known and well-tested Java class libraries, such as Spring (<http://www.springframework.org>), Hibernate (<http://www.hibernate.org>), and Adobe Flex (<http://www.flex.org/>). In addition, by adhering to the Java Enterprise Edition (JEE) Architecture and specifications (<http://java.sun.com/javase/technologies/>), design and implementation of IHTA Java classes are fully portable to a compliant JEE container with minimal impact.

The IHTA Hardware Architecture will achieve extensibility and reliability through its implementation of load balancing and fail-over in each of its architectural tiers: Web server, application server, and database. IHTA will implement load balancing in its Web server tier through use of a primary and a backup hardware load balancer, which will only be activated in the event of a catastrophic failure of the primary load balancer. Each load balancer will be configured to forward incoming HTTP requests to a logical grouping of HTCE Web server instances called a “cluster”. A cluster of HTCE Web server instances can be easily scaled horizontally by adding additional Web server instances into the cluster for extensibility. Reliability in the HTCE Web server tier will be implemented through the fail-over capability of a load balancer. A load balancer will automatically redirect incoming requests to a malfunctioning Web server instance in a cluster to the next operational Web server instance in that cluster. Extensibility and reliability will also be achieved through the implementation of clusters in the application server and database server tier. Additional server instances can be added to the cluster for extensibility, and incoming requests to a failed server instance in a cluster are automatically redirected to the remaining server instances in the cluster for processing using a round-robin algorithm.

IHTA software will be packaged as an EAR file, which adheres to JEE standard for packaging Web-based Java applications. In addition, the IHTA server configuration will be consolidated in a domain that uses JEE specifications, such as Java Database Connectivity (JDBC) data source and connection. This technique allows for IHTA software to be deployed on any JEE compliant application server and IHTA server configurations can easily be migrated over to a suitable JEE compliant application server.





## 9. Architectural Mechanism

As the requirements for system load increase, additional servers will be added to the IHTA Web server, application, and database clusters. Availability is achieved through data replication implemented using SQL mirroring to synchronize data between the Production site and the DR site. In the event of a catastrophic failure, requests coming in to the primary load balancers at the primary site in Martinsburg will be automatically failed-over to the backup load balancers at the secondary site at Hines. Failures of individual servers in the application server cluster will be managed by the Weblogic Application Server, which will fail over incoming requests to the remaining server instances in a cluster. Fail-over capability at the Web server tier will be achieved by the configuration at the load balancer and in conjunction with the Weblogic plug-in installed on the Web server. Database server fail-over is documented in the *HTCE Database Design Document*.



## Attachment A - Approval Signatures

**Title:** Home Telehealth Capability Enhancements (HTCE)

**Application:** Integrated Home Telehealth Application (IHTA)

**Version:** 6.0

NOTE: A customer decision on whether or not to move forward with the development of the Veteran's Common Web Interface module of IHTA will affect the content of this document. Consequently, approval signatures will be obtained in a subsequent release.

---

Signed: \_\_\_\_\_ Date: \_\_\_\_\_  
[REDACTED] Integrated Project Team (IPT) Chair/IT Program Manager

---

Signed: \_\_\_\_\_ Date: \_\_\_\_\_  
[REDACTED] OTS Program Analyst (Business Sponsor)

---

Signed: \_\_\_\_\_ Date: \_\_\_\_\_  
[REDACTED] HTCE Project Manager

---

Signed: \_\_\_\_\_ Date: \_\_\_\_\_  
< Enterprise Architecture >

---

Signed: \_\_\_\_\_ Date: \_\_\_\_\_  
[REDACTED], Service Delivery and Engineering Representative