

My HealtheVet Core Development and Enhancement Services

12.9

System Design Document



June 2014

Version 1.0

Revision History

Date	Version	Description	Author
27 May 2014	1.0	Baselined at a 1.0 for delivery with 12.9 Release	
23 May 2014	0.6	Technical Writer review	
21 May 2014	0.5	Updated various sections in relation to configurations of the various environments	
19 May 2014	0.4	Inclusions from Tech Refresh Architecture and PHR Manager Redesign	
14 Apr 2014	0.3	Technical Writer review. Update to new SDD ProPath template.	
11 Apr 2014	0.2	Added Eauth enhancement	
10 Oct 2014	0.1	Re-versioned for 12.9 release. Checked against ProPath template.	

Table of Figures

Figure 1: MHV Functionality.....	13
Figure 2: Mental Health Process Overview	15
Figure 3: Create a Goal Process.....	16
Figure 4: Projected MHV Users.....	21
Figure 5: Projected IPA Users.....	21
Figure 6: Application Context for MHV.....	25
Figure 7: MRP—My Goals Application Context Diagram	27
Figure 8: Logical System Diagram	29
Figure 9: MRP My Goals Subsystem Component View	30
Figure 10: Conceptual Data Model.....	36
Figure 11: Goal Maintenance Conceptual Data Model.....	37
Figure 12: Goal Reports Conceptual Data Model	38
Figure 13: CDCO Environment	39
Figure 14: Conceptual Networks and Environments.....	41
Figure 15: Sample Conceptual Networks and Environments.....	41
Figure 16: MHV Development Architecture (as of 05/21/2014)	42
Figure 17: MHV L&P Architecture (as of 05/21/2014)	43
Figure 18: MHV Pre-Production Architecture (as of 05/21/2014)	44
Figure 19: MHV Production Architecture (as of 05/21/2014)	45
Figure 20: Software Architecture	51
Figure 21: Portlet Applications.....	52
Figure 22: Generated portletApp Components	53
Figure 23: Content portletApps Components.....	54
Figure 24: User Account Mgmt Package	54
Figure 25: PHR portletApp Components	55
Figure 26: RxRefill portletApp Components	55
Figure 27: Portal Core Components.....	56
Figure 28: Logical Communications Segmentation	57
Figure 29: MHV Target System Design	59
Figure 30: MHV Communications Architecture.....	62
Figure 31: MHV Communication Paths.....	63
Figure 32: External Communication with MHV	64
Figure 33: Hardware Detailed Design – Terramark Culpeper, VA.....	68
Figure 34: Hardware Detailed Design - Secondary Site Terramark Miami, FL.....	69
Figure 35: MHV Site Redesign Logical Model.....	71
Figure 36: MHV Site Redesign Functional Model	73
Figure 37: Project Component Breakdown Structure	74
Figure 38: MHV Product Perspective Block Diagram.....	76
Figure 39: Detail Communication Diagram	83
Figure 40: MHV System LAN Topology.....	85
Figure 41: MHV PHRMGR Design.....	86
Figure 42: PHRMGR-R Deployment Diagram	87
Figure 43: PHRMGR-R Component Drill-Down.....	88

<i>Figure 44: PHRMGR-R Component Dependency Graph.....</i>	<i>88</i>
<i>Figure 45: Integration Services.....</i>	<i>90</i>
<i>Figure 46: Ensemble Messaging Services</i>	<i>91</i>
<i>Figure 47: External Interface Components.....</i>	<i>93</i>
<i>Figure 48: Technical Architectural Layering.....</i>	<i>94</i>
<i>Figure 49: Interface Diagram</i>	<i>95</i>

Table of Tables

<i>Dev Tools and Uses</i>	1
<i>Scope Inclusions</i>	2
<i>Scope Exclusions</i>	2
<i>Use Case Actors</i>	5
<i>Business Processes</i>	13
<i>Prescription Refill Hits over Time</i>	20
<i>Workload and Performance Requirements</i>	22
<i>Operational Requirements</i>	22
<i>Technical Requirements</i>	22
<i>Application Context Description</i>	26
<i>Application Context Description</i>	28
<i>Objects in the MRP My Goals Subsystem Component View</i>	31
<i>Application Locations</i>	31
<i>Application Users</i>	32
<i>Database Inventory</i>	38
<i>Technology Location</i>	40
<i>SysTest Hardware Overview</i>	46
<i>Preview Hardware Overview</i>	47
<i>Integration B Hardware Overview</i>	47
<i>Integration A Hardware Overview.</i>	49
<i>MHV Database Configuration</i>	65
<i>Database Schema</i>	66

Table of Contents

1. Introduction	1
1.1. Purpose of the SDD	1
1.2. Identification	2
1.3. Scope	1
1.4. Constraining Policies, Directives and Procedures	2
1.4.1. Constraints	2
1.4.2. Architecture Review Board (ARB)	3
1.4.3. Architecture Reviews of Requirements	3
1.4.4. Architecture Reviews of Design	4
1.4.5. System Design Directives, Policies, and Guidelines	4
1.5. User Characteristics	5
1.6. Relationship to Other Documents and Plans	8
1.7. Definitions, Acronyms, and Abbreviations	8
1.8. References	8
1.9. Style Guide	8
1.10. Methodology, Tools, and Techniques	8
1.10.1. MHV Programming and Design Guidelines	9
1.10.2. Proven Software Engineering Practices	9
1.10.3. Frameworks and Tools	9
1.10.4. Iterative Development Lifecycles	9
1.11. User Problem Statement	10
1.11.1. User Objectives	10
2. Background	10
2.1. Overview of the System	11
2.2. Overview of the Business Process	12
2.3. Business Benefits	16
2.4. Assumptions	17
2.4.1. Design Assumptions	17
2.4.2. Design Constraints	18
2.4.3. Design Trade-offs	18
2.5. Overview of the Significant Requirements	19
2.5.1. Overview of Significant Functional Requirements	20
2.5.2. Overview of Functional Workload/Performance Requirements	20
2.5.3. Overview of Operational Requirements	22
2.5.4. Overview of the Technical Requirements	22
2.5.5. Overview of the Security or Privacy Requirements	23

2.5.6.	Overview of System Criticality and High Availability Requirements	23
2.5.7.	Single Sign-on Requirement	24
2.5.8.	Requirement for Use of Enterprise Portals	24
2.5.9.	Special Device Requirements	24
2.6.	Legacy System Retirement	24
3.	Conceptual Design	24
3.1.	Conceptual Application Design	25
3.1.1.	Application Context	25
3.1.2.	High Level Application Design	29
3.1.3.	Application Locations	31
3.1.4.	Application Users	32
3.2.	Conceptual Data Design	35
3.2.1.	Project Conceptual Data Model	35
3.2.2.	Database Information	38
3.2.3.	User Interface Data Mapping	38
3.3.	Conceptual Infrastructure Design	39
3.3.1.	System Criticality and High Availability	39
3.3.2.	Special Technology	40
3.3.3.	Technology Locations	40
3.3.4.	Conceptual Infrastructure Diagram	40
4.	System Architecture	42
4.1.	Hardware Architecture	46
4.2.	Software Architecture	50
4.2.1.	Portlet Applications	52
4.2.2.	Generated	52
4.2.3.	Content Portlets	53
4.2.4.	User Account Management	54
4.2.5.	PHR	54
4.2.6.	RxRefill	55
4.2.7.	MHV Portal Core	55
4.3.	Network Architecture	56
4.3.1.	MHV System Network Communications	56
4.3.2.	Logical Communications Segmentation	56
4.3.3.	Backend Data Synchronization	57
4.4.	Service Oriented Architecture / ESS	57
4.5.	Enterprise Architecture	58
4.6.	Services	60
4.7.	Utilities	61
4.8.	Communications Architecture	61

5.	Data Design	65
5.1.	Database Management System (DBMS) Files	65
5.1.1.	Database Configuration	65
5.1.2.	Database Schema	65
5.1.3.	EVAULT Schema.....	66
5.1.4.	SDSADM Schema	66
5.1.5.	SMS Schema	66
5.1.6.	PHRMGR Schema	66
5.2.	Non-DBMS Files	67
5.3.	Data View.....	67
6.	Detailed Design	67
6.1.	Hardware Detailed Design.....	67
6.2.	Software Detailed Design.....	70
6.2.1.	Conceptual Design	70
6.2.2.	Specific Requirements	82
6.3.	Communications Detailed Design	82
6.4.	Network Detailed Design.....	82
6.5.	Service Oriented Architecture / ESS Detailed Design	85
6.6.	MHVAPI Façade & PHRMGR-R Design	85
6.6.1.	MHV API Façade Enterprise Application	86
6.6.2.	PHRMGR-R Enterprise Application	86
6.7.	Heracles Toolkit	88
6.8.	Integration Framework	89
6.8.1.	Integration Services.....	89
6.8.2.	Ensemble Messaging Services.....	91
6.9.	Inversion of Control.....	91
6.10.	Data Access Object (DAO)	92
6.11.	Presentation Control	92
6.12.	MPI/MVI and VistA Communication Mechanism	92
6.13.	VAPii Communication Mechanism	92
6.14.	MhvDomainApp Framework	92
6.15.	InterApp Framework.....	93
6.16.	Secure Messaging Logical View.....	93
7.	External System Interface Design.....	93
7.1.	Interface Architecture.....	94
7.2.	Interface Detailed Design	95
8.	Human-Machine Interface	96
8.1.	Interface Design Rules	96
8.2.	Inputs	96

8.3. Outputs	96
8.4. Navigation Hierarchy	96
8.4.1. Screen [x.1]	96
8.4.2. Screen [x.2]	96
8.4.3. Screen [x.3]	96
9. Security and Privacy.....	96
9.1. Security.....	96
9.2. Privacy	97
Attachment A - Approval Signatures.....	98
Additional Information.....	99
A.1. RTM	99
A.2. Packaging and Installation.....	99
A.3. Design Metrics	99
A.4. Acronym List and Glossary	99
A.5. Required Technical Documents	102

1. Introduction

The Department of Veterans Affairs (VA) My HealtheVet (MHV) program is a complex, high visibility program in the Initiative 7: New Models of Care Initiative in the VA Office of Information Technology (OIT). The program provides a web-based application for Veterans, their families, and care providers to access health resources in an online environment. It improves Veteran health care by providing easy access to health information, online resources, and facilitates patient/health care provider interactions. MHV gives the Veteran access to VA benefits, special programs, and health information and services. It also provides the Veteran web-based tools to increase their knowledge about health conditions, manage their health records, and communicate with health care providers. Veterans now have the ability to take a more proactive approach to managing their health and utilizing VA health services and benefits.

The purpose of the MHV Capabilities Enhancements Project is to stabilize the MHV portal and the environments supporting the portal. The MHV portal currently runs on non-supported hardware and software that is no longer in compliance with current Identity Management (IdM) processes, making the environments supporting the portal unstable and incapable of meeting specifications required for future growth, use, and product enhancements. In order to support the applications that will be housed in the MHV portal, and the future increase in traffic anticipated by these applications, the platform must be stabilized and re-architected. Failure to proceed with the Capabilities Enhancements Project would not only put the future capability of MHV to meet user and technical requirements at risk, but would also limit future endeavors to support Veterans' access to medical records. This would result in a failure to meet Secretarial and Presidential mandates.

The MHV system incorporated additional functionality for the My Recovery Plan (MRP). MRP is a series of applications that are developed to support the recovery of Veterans with mental health and substance use disorders. The MRP application will help Veterans monitor and track their mental health symptoms, medication adherence and side effects, and progress towards recovery goals, as well as manage their triggers using a coping plan. The MRP application will be accessed via MHV Health Portal. MRP is a component of a broader Improve Veteran Mental Health (IVMH) initiative to improve the mental health of our Veterans.

The purpose of the MHV Technical Refresh (TR) project is to define an architecture for MHV systems that is compliant with the OneVA Enterprise Technical Architecture (ETA) and to provide a technical platform that supports identified operational, capacity, and business needs. The 12.9 release of MHV is the first release of MHV systems that includes system components that have been reengineered to be consistent with the TR architectures.

1.1. Purpose of the SDD

This System Design Document describes the architecture of the MHV as well as the subsystem of MRP and the additional the Veterans Health Information Systems and Technology Architecture patching work performed in this build.

This SDD is based on requirements already gathered for the MHV and MRP My Goals subsystem, and will render these requirements into a high level architecture that will then become detailed requirements defined during the project's life cycle. This document focuses on

the design of key functional components required to successfully demonstrate MHV as well as the MRP My Goals subsystem features and functionalities.

This SDD serves the following primary purposes:

- Translates requirement specifications into a document from which developers can create the actual system
- Identifies the top-level system architecture
- Identifies hardware, software, communication, and interface components within the MHV and MHV's MRP Infrastructure

1.2. Identification

Many tools are used only for code generation, testing, design, and documentation; they are categorized as “Dev only”. Tools categorized as “Deployed” must be packaged into the archive files deployed to the application server.

A substantial fraction of MHV tools do not demonstrate compliance with the latest version of the approved tools list. The following list details the several reasons for this:

- The approved tool list is not current
- The approved tool list does not represent the same granularity as the listed tools. This is indicated by those tools with a compliance value of “No category”
- Many of the tools on the list function as dependencies of other tools on the list and the existing framework (Atlas Framework). Thus, the many instances of XML parsers and binding frameworks. This is beyond the control of the MHV team.
- Many of the tools are fully encapsulated by MHV portal applications and are neither exposed nor impose additional requirements on the hosting environment

The MHV team is working closely with Technical Reference Model (TRM), the enterprise architects to resolve these tool issues.

Table 1: Dev Tools and Uses

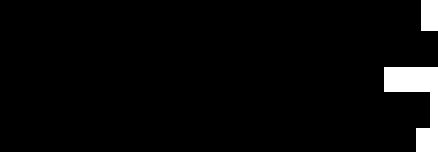
Tool	Used For	Ver.	Type	License #	License Administered By	In TRM?	TRM Research Notes
Ant	Building software	1.6.5	Development	Apache Software License	Apache - http://ant.apache.org/	Yes	
Ant Apache RegEx	Regular Expressions	1.6.5	Development	Apache Software License	Apache - http://ant.apache.org/	Yes	
Apache	Webserver	2.2.3	Development	Apache License	http://www.apache.org/licenses/LICENSE-2.0	Yes	
ANTLR	Language recognition tool	2.7.2	Development	ANTLR Public License	http://www.antlr.org/	Yes	
AOP Alliance	AOP api	1.0	Development	GNU Lesser General Public License	http://www.gnu.org/licenses/lgpl.html	Yes	
ASM Framework	Java bytecode manipulation framework	3.0	Development	ASM Public License	http://asm.ow2.org/	Yes	
Aspectj	aspect-oriented extension	1.5.3	Development	Eclipse Public License - v 1.0	http://www.eclipse.org/legal/epl-v10.html	No	Prohibited
AspectWerkz	AOP api	0.8.1	Development	GNU Lesser General Public License	http://www.gnu.org/licenses/lgpl.html	No	Unapproved
Atlas	Atlas Weblogic MDA	1.2	Development	CollabNet/Tigris.org Apache-style license	http://atlas.tigris.org/	Yes	
Avalon framework	Spring integration	4.1.3	Development	Apache Software License, Version 1.1	http://apache.xmlcity.org/avalon/framework/v4.1.3/	No	Not Listed
Axis	Web Services API	1.4/1.5.1	Development	The Apache Software License, Version 2.0	http://ws.apache.org/axis	No	Not Listed
Backport-util-concurrent	Java API	3.0	Development	Public License	http://creativecommons.org/licenses/publicdomain/	No	Not Listed
Beehive (Apache)	Annotation based tags and controls	1.0.2/1.1/1.2	Development	CollabNet/Tigris.org Apache-style license	http://atlas.tigris.org/	No	Prohibited

Tool	Used For	Ver.	Type	License #	License Administered By	In TRM?	TRM Research Notes
Batik	Java-based toolkit for applications or applets that want to use images in the Scalable Vector Graphics (SVG)	1.6	Development	Apache License, Version 2.0	http://www.apache.org/licenses/	No	Not Listed
Bean Shell (bsh)	Bean Scripting	2.0/2.0.1	Development	Dual Licensing: Sun Public License / Gnu Lesser Public License	http://www.beanshell.org/license.html	No	Not Listed
CE Wolf	Chart enabling web object framework	0.12.0/1.0	Development	LGPL license	http://cewolf.sourceforge.net/new/	Yes	
CGlib	Code generation	2.1.3	Development	Apache License, Version 2.0	http://www.apache.org/licenses/	No	Prohibited
Classworlds	complex manipulation of Java's ClassLoaders	1.1-alpha2	Development	The Codehaus Project License v 1.1.1.1	http://classworlds.codehaus.org/license.html	No	Not Listed
Commons-beanutils	Easy-to-use wrappers around the Java reflection and introspection APIs.	1.7.0	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Commons-chain	"Chain of Responsibility" pattern implementation.	1.0	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Commons-cli	Command Line arguments parser.	1.0	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Commons-codec	General encoding/decoding algorithms (for example phonetic, base64, URL).	1.3	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Commons-collections	Extends or augments the Java Collections Framework.	3.2	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Commons-dbcp	Database connection pooling services.	1.2.1	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Commons-digester	XML-to-Java-object mapping utility.	1.8	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	

Tool	Used For	Ver.	Type	License #	License Administered By	In TRM?	TRM Research Notes
Commons-discovery	Tools for locating resources by mapping service/reference names to resource names.	0.2	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Commons-el	Interpreter for the Expression Language defined by the JSP 2.0 specification.	1.0	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Commons-email	Library for sending e-mail from Java.	1.0	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Commons-fileupload	File upload capability for your servlets and web applications.	1.0	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Commons-httpclient	Framework for working with the client-side of the HTTP protocol.	3.0	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Commons-io	Collection of I/O utilities.	1.2	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Commons-lang	Provides extra functionality for classes in java.lang.	2.1	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Commons-logging	Wrapper around a variety of logging API implementations.	1.1	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Commons-pool	Generic object pooling component.	1.2	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Commons-validator	Framework to define validators and validation rules in an xml file.	1.3.0	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Concurrent	Utility classes for concurrent Java programming	1.3.4	Development	TECHNOLOGY LICENSE FROM SUN MICROSYSTEMS, INC	http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/sun-u.c.license.pdf	No	Not Listed
Crimson	Java XML parser	1.1.3	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	No	Not Listed

Tool	Used For	Ver.	Type	License #	License Administered By	In TRM?	TRM Research Notes
Display Tag	Html table tag	1.1	Development	http://displaytag.sourceforge.net	http://www.displaytag.org/1.2/	No	Prohibited
DOM4j	XML framework	1.6.1	Development	BSD style license	http://dom4j.sourceforge.net/license.html	No	Not Listed
Doxia	content generation framework	1.0alpha5	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	No	Not Listed
Dumbster	fake SMTP server designed for unit and system testing	1.6	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	No	Not Listed
Eclipse	IDE	3.2	Development	Eclipse Public License - v 1.0	http://www.eclipse.org/org/documents/epl-v10.php	Yes	
Ehcache	Caching framework	1.1	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Ensemble	Middleware Messaging, BPM, ESB	2010.2.6	Development	Obtainable through CDCO	Obtainable through CDCO	Yes	
Freemarker	Generate text output based on templates	2.3.8	Development	Visigoth Software Society	http://freemarker.org/docs/app_licens_e.html	No	Not Listed
ganymed	implements the SSH-2 protocol in pure Java	build210	Development	BSD style license	http://www.ganymed.ethz.ch/ssh2/	No	Not Listed
HAPI	Java HL7 Transformations	0.4.3	Development	GNU General Public License	http://sourceforge.net/directory/licenses/gpl/	No	Not Listed
Hibernate	Object/relational mapping	3.1.1	Development	GNU Lesser General Public License	http://www.gnu.org/licenses/lgpl.html	Yes	
HSQLDb	Embedded and server mode database	1.8.0.1	Development	BSD License	http://hsqldb.org/web/hsqLicense.html	No	Not Listed
Ittext	Java PDF Utility	1.3.1	Development	GNU Affero General Public License version 3.	http://info.itextpdf.com/licensing	Yes	
jakarta-regexp	Java Regular Expression package	1.4	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	No	Not Listed
Jasper Reports	Web based reporting	1.2.1	Development	GNU Lesser General Public License	http://jasperreports.sourceforge.net/license.html	Yes	
Java	Programming language	1.5.0	Development	Proprietary	http://www.oracle.com/technetwork/java/javase/terms/license/index.html	Yes	

Tool	Used For	Ver.	Type	License #	License Administered By	In TRM?	TRM Research Notes
Jaxen	Xpath processor	1.1 beta-9	Development	The Werken Company	http://jaxen.codehaus.org/	No	Not Listed
Jaxme	Java/XML binding	0.3	Development	BSD License/Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	No	Not Listed
Jboss Cache	Caching framework	1.2.2	Development	GNU Lesser General Public License	http://www.gnu.org/licenses/old-licenses/lgpl-2.1.txt	No	Not Listed
Jdom	XML framework	1.0	Development	Apache-style open source license	http://jdom.org/docs/faq.html#a0030	No	Not Listed
Jfree	Web page Charts	1.0.1	Development	GNU Lesser General Public License	http://www.gnu.org/licenses/lgpl.html	Yes	
Jmock	create mock objects to support unit testing of java code	1.0.1	Development	jMock Project License	http://jmock.org/license.html	Yes	
Jgroups	A Toolkit for Reliable Multicast Communication.	2.2.7	Development	GNU Public License 2.1 (LGPL).	http://www.jgroups.org/license.html	No	Not Listed
JMS (Sun)	Java messaging	1.0.2	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Junit & Junit-addons	Unit testing	3.8.2 / 1.4	Development	Common Public License Version 1.0	http://opensource.org/licenses/eclipse-1.0.txt	Yes	
Log4j	Process logging	1.2.14	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Logkit	Process logging	1.0.1	Development	GNU General Public License, version 2	http://olex.openlogic.com/licenses/avalon-logkit-apache-v1_1-license	No	Not Listed
msv	Dynamic Configuration Toolkit	#####	Development	Apache Software License, version 1.1	http://dom4j.sourceforge.net/dependencies.html	No	Not Listed
OGNL	binding and expression language	2.6.5	Development	BSD License	http://www.opensource.org/licenses/bsd-license.php	No	Not Listed
Oro	text-processing utility	2.0.8	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	No	Prohibited

Tool	Used For	Ver.	Type	License #	License Administered By	In TRM?	TRM Research Notes
PJA Toolkit	Java library for drawing graphics	2.5	Development	GNU General Public License	http://www.gnu.org/copyleft/gpl.txt	No	Not Listed
Plexus	IoC container and components	1.0.3	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
poi	reading-writing Microsoft Office file formats using pure Java	2.5.1-final-20040804	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Pull Parser	incremental XML parser	2.0	Development	LGPL license	http://www.extreme.indiana.edu/xgws/xsoap/xpp/download/PullParser2/LICENSE.txt	Yes	
Qdox	Java annotations parser	1.5	Development	Common Public License Version 1.0	http://qdox.codehaus.org/usage.html	No	Not Listed
Quartz	Job scheduling utility	1.5.0-RC1	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
ServiceMix WSIF	Web Services Invocation Framework	2.0	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	No	Not Listed
Slide (Jakarta)	content management framework	2.0	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	No	Not Listed
Spring	Dependency injection and inversion of control	2.0	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Stax	Iterative, event-based processing of XML documents	1.0.1	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	No	Not Listed
Stax Utils	Streaming api for XML	##### ##	Development	GNU LGPL License		No	Not Listed
Struts	MVC Framework	1.2.7 and 2	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Subversion	CM Repository Control	1.6.17	Development	Subversion License	http://svn.apache.org/repos/asf/subversion/tags/1.6.0/www/license-1.html	No	Prohibited
Surefire Reports	Maven plug-in: test results report into html format	1.4	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	No	Not Listed

Tool	Used For	Ver.	Type	License #	License Administered By	In TRM?	TRM Research Notes
Tag libs	create custom actions and encapsulate functionality in web applications	1.0.6	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	No	Prohibited
Velocity	Templating utility	1.4	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
WebLogic Platform	WL Application Server, Portal Server, Workshop IDE	10.3.2 and 10.3.5	Development	Obtainable through CDCO	Obtainable through CDCO	Yes	
Woodstox	StAX-compliant (JSR-173) Open Source XML-processor	3.0.1	Development	BSD License	http://www.opensource.org/licenses/bsd-license.php	Yes	
Wsd4J	Creation, representation, manipulation of WSDL documents	1.5.2	Development	IBM Common Public License	http://rpmfind.net/linux/RPM/centos/5.9/x86_64/CentOS/wsd4j-1.5.2-4jpp.1.x86_64.html	No	Not Listed
Xalan	XSLT processor	2.6.0	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Xerces	XML processing utilities	2.0.2	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
Xfire	java SOAP framework	1.2.2	Development	Envoi Solutions	http://xfire.codehaus.org/License	No	Not Listed
XML APIs	XML utilities	2.0.2	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	No	Not Listed
XMLbeans	XML binding framework	2.3.0	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	Yes	
XML Resolver	Resolves public identifiers to local resources.	1.1	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	No	Not Listed
XMLSchema (Apache)	manipulate and generate XML schema representations	1.1	Development	GNU Lesser General Public License LGPL v3	http://www.gnu.org/licenses/lgpl-3.0.txt	Yes	
Xpp3	XML pull parser	1.1.3.3	Development	Apache License, Version 2.0	http://www.apache.org/licenses/LICENSE-2.0.txt	No	Not Listed
XSDLib	XSD utility	##### ##	Development	Apache Software License, version 1.1	http://dom4j.sourceforge.net/dependencies.html	No	Not Listed

1.3. Scope

The MHV Web-based application creates a new, online environment where Veterans, their advocates, and health care providers can come together to optimize Veterans' health care. Web technology combines essential personal health record information, enhanced by online health resources, to enable and encourage patient/clinician collaboration.

This document includes the design of the PHR Manager Redesign (PHRMGR-R) MHV application component. This component was developed as the reference implementation (RI) of the MHV Technical Refresh architecture. The 12.9 release of MHV initiates the process for upgrading the MHV application platform and elements of the technical stack. Both of these activities are required in order to implement the PHRMGR-R and validate the architecture and that systems are functioning and performing as expected.

MHV will provide health education information and health self-assessment tools. Veterans will be able to reorder medications, view appointments, and review their health records online. This computerized patient record system (CPRS) builds upon VistA to operate patient/data-centric, use the best modern technology, move to an enterprise-wide approach, standardize health data and communications, substantially enhance the health systems supporting Veterans' care, and secure health systems and Veterans' health information.

MHV included functionality for the MRP project is to provide requirements for an interactive set of web-based tools that will support Veterans with mental health or Substance Abuse Disorder (SUD) conditions in their recovery efforts. These MHV and MRP tools will apply evidence-based practices, defined by VA clinical and administrative staff, and enable the Veteran to track important aspects of their self-care and professional care as well as perform a variety of recovery oriented exercises. These tools are complemented by a progress tracking function that assesses and tracks Veteran recovery progress over a period of time. The MRP My Goals subsystem, will specifically allow a Veteran to set individualized, personally relevant recovery goals, and monitor his or her progress towards achieving these goals.

The scope of the MRP My Goals SDD includes developing a system design for the MRP My Goals subsystem functional components. In other words, the MRP My Goals subsystem shall enable the Veteran to:

- Identify larger life goals
- Break goals down into manageable components
- Track progress towards those goals
- Help patients identify and overcome barriers to goal attainment
- Identify strengths and goal rewards
- Generate goal reports

The functional requirements themselves are comprised of a requirements traceability matrix, system use case model, a list and descriptions of MRP system actors, and the MRP system use cases. Wireframes will also be developed and vetted with the stakeholders. Test cases/test scripts will consequently be derived from the system use cases. These and other project artifacts such as the master test plan and product demos will be developed using VA OIT ProPath system

development lifecycle processes. The following tables list MRP My Goals subsystem scope inclusions and exclusions.

Table 2: Scope Inclusions

Includes
Support MRP My Goals subsystem functional and non-functional requirements
Support interface(s) to My Health eVet portal

Table 3: Scope Exclusions

Excludes
Development of interfaces to VA clinical systems such as VistA and CPRS
Full integration with a Patient Education Management System (PEMS) that would allow users to take instructive and interactive courses related to SUD, post-traumatic stress disorder (PTSD), Depression, Schizophrenia, and various other mental health topics.

1.4. Constraining Policies, Directives and Procedures

While the development methodology for the development of software for the MHV program evolves, the following architecture processes and standard form the basis for architectural discussion and reviews, the development of architecture mechanisms, the approval of architecture changes, and the architectural standards and guidelines.

The MHV project will be managed using the policies and procedures in the IT Program Management Guidelines. Please refer to the following site: [http://\[REDACTED\]](http://[REDACTED])

The MHV project will utilize the ProPath SDLC guidelines from the Engineering Process Group (EPG) and located at the following site: [h \[REDACTED\]](http://[REDACTED])

ProPath is the front-end tool to a Process Asset Library containing information regarding standard processes. It is a one-stop shop providing critical links to the formal approved processes, artifacts, and templates to assist the MRP project team in facilitating their daily work.

The governance of system development is provided through the Project Management Accountability System (PMAS). The link to the PMAS site is located at: [\[REDACTED\]](http://[REDACTED])

Moreover, implemented by the Assistant Secretary for IT, PMAS is a VA-wide initiative to better empower project managers and teams to meet their mission: delivering world-class IT products that meet business needs on time and within budget.

1.4.1. Constraints

The following constraints will greatly affect the architecture of the MHV Portal framework.

- No users will be allowed to access and/or manipulate another's personal data unless delegated to do so
- All communications between applications and users must be over a secured channel

- Accessibility of resources will be strictly based on the role of the user currently logged into the system
- The system must be able to store users' personal information securely and safely
- The system will support physically challenged users
- The system must communicate with other VA and non-VA systems to provide consolidated services to Veterans
- The design and implementation platform chosen for this application is the Oracle WebLogic Platform. The WebLogic Platform comprises the WebLogic Server, WebLogic Workshop Framework, and WebLogic Portal. WebLogic components other than WebLogic Server restrict the MHV Portal from becoming vendor neutral in design and implementation.
- The system must communicate with other external systems such as the Health Data Repository (HDR), VistA, Master Patient/Veteran Index (MPI/MVI) and Veterans Authorization Preferences Interface Improvements (VAPii). Future requirements and guidelines will include communications with forthcoming external systems and services, such as Person Lookup, Pharmacy Reengineering, Security Services, and Enrollment System Redesign (ESR).
- Hardware interfaces—The MHV application and all its associated functionality will rely exclusively on the hosted environments at the Austin Information Technology Center (AITC)
- Compliance with the VA Enterprise Architecture TRM
- Unless specifically noted, all styles applied throughout MHV features will employ the styles outlined by the MHV IT Program Office Design Team
- The MHV application will utilize labeling and other natural language components in line with VA strategic plan initiatives and existing accessibility regulations
- The MHV application will be fully integrated with portlet design for deployment of applications. MHV's targeted user population is adverse to the display of popup messages. To minimize the number of popup message, successful results messages are minimized. However, error messages are utilized where appropriate (e.g., for lack of data in a required field).

1.4.2. Architecture Review Board (ARB)

The MHV ARB is a board of SMEs chartered by the MHV Program Manager to meet regularly to discuss and react to changes proposed to the MHV architecture. As the approving authority for all architectural issues, the MHV ARB grants or denies each change.

1.4.3. Architecture Reviews of Requirements

As part of the evolving development methodology, an architecture review of requirements for new features or projects exists in the form of a checklist to assess the impact of those requirements on the existing MHV architecture. Where an impact to the architecture is found as a

gap or potential change, the reviewer will document the finding and forward a suggested mechanism or change request to the MHV ARB. The review checklist becomes an artifact of the development process that benefits the architects and developers during tool selection, design, or construction.

1.4.4. Architecture Reviews of Design

As part of the evolving development methodology, an architecture design review form allows a reviewer to document how a design compares to the current MHV architecture baseline.

1.4.5. System Design Directives, Policies, and Guidelines

The MRP system design is also guided by the following directives, policies, and guidelines:

1. 44 U.S.C. § 3541, “Federal Information Security Management Act (FISMA) of 2002”
2. FIPS Pub 201, “Personal Identity Verification of Federal Employees and Contractors,” March 2006
3. 10 U.S.C. § 2224, “Defense Information Assurance Program”
4. Software Engineering Institute, Software Acquisition Capability Maturity Modeling (SA CMM) Level 2 procedures and processes
5. 5 U.S.C. § 552a, as amended, “The Privacy Act of 1974”
6. 42 U.S.C. § 2000d “Title VI of the Civil Rights Act of 1964”
7. Department of Veterans Affairs Directive 0710, “Personnel Suitability and Security Program,” September 10, 2004
8. VA Directive 6102, “Internet/Intranet Services,” July 15, 2008
9. 36 C.F.R. Part 1194 “Electronic and Information Technology Accessibility Standards,” July 1, 2003
10. OMB Circular A-130, “Management of Federal Information Resources,” November 28, 2000
11. 32 C.F.R. Part 199, “Civilian Health and Medical Program of the Uniformed Services (CHAMPUS)”
12. An Introductory Resource Guide for Implementing the Health Insurance Portability and Accountability Act (HIPAA) Security Rule, March 2005
13. Sections 504 and 508 of the Rehabilitation Act (29 U.S.C. § 794d), as amended by the Workforce Investment Act of 1998 (P.L. 105-220), August 7, 1998
14. Homeland Security Presidential Directive (12) (HSPD-12)
15. VA Directive 6500, “Information Security Program,” August 4, 2006
16. VA Handbook 6500, “Information Security Program,” September 18, 2007
17. VA Handbook 6500.6, “Contract Security,” March 12, 2010
18. NBS SP500-153, “Guide to Auditing for Controls and Security: A System Development Life-Cycle Approach,” April 1988

19. Program Management Accountability System (PMAS) portal (reference Vendor Library at [http://\[REDACTED\]](http://[REDACTED]))
20. ProPath Process Methodology (reference Vendor Library at [http://\[REDACTED\]](http://[REDACTED])) NOTE: In the event of a conflict, ProPath takes precedence over other processes or methodologies.
21. TRM (reference Vendor Library at [http://\[REDACTED\]](http://[REDACTED]))
22. National Institute Standards and Technology (NIST) Special Publications

1.5. User Characteristics

This section describes each actor identified in the MHV Use Case Model.

Table 4: Use Case Actors

MHV Use Case Actor	Description of Actions
DIRECT	Refers to either the DOD Direct system or the VA Direct system, which has services used to send CCD documents to participating health care providers securely.
HDR	HDR refers to the national instance of Health Data Repository. MHV will provide read access to the HDR through its middleware (RPC) messaging framework - MDWS.
HRA	Refers to the Health eLiving Assessment application that can be linked to from within the MHV portal application.
In-Person Authentication (IPA) User	Conduct in-person authentication procedures, and provide Veterans online access to their VA personal health information.
Master Veteran Index (MVI)	Made up of two systems, PSIM; the Java-based Health eVet Enterprise hosted portion and MVI; the Cache' based Vista portion. Combined, the service is the authoritative source for Identity Management functions for persons throughout the VA. For details, please check MVI_Service_Description.docx. Connection timeout is set to 45 seconds and request timeout is set to 25 seconds. The communication is via VAAFI integration for all the calls going to MVI.
MHV Administrator	VA staff responsible for managing and maintaining the MHV system.
MHV Help Desk	VA staffs who supports users of MHV by responding to comments and questions submitted to the help desk.
MHV Registered User	All users who have established a personal account in the MHV system. Registered users are given access to system functionality that is unavailable to unregistered users.

MHV Use Case Actor	Description of Actions
MHV Unregistered User	Users who have not logged into the MHV system. Unregistered users can access all publicly available content, such as the health information library, yet they do not have access to the system's core functionality, such as viewing and managing health records, making entries into health logs, designating and acting as a Delegate, refilling prescriptions, and viewing appointments. If an Unregistered user has not created an account, the User can register at any time. Once this account has been created, the Unregistered User can login at any time.
MPI/MVI	MPI/MVI is a central index of unique MHV users. MHV users Health Administration's implementation includes assigning each patient an Integration Control Number (ICN) and a Coordinating Master of Record (CMOR) site. The ICN assignment enables the sharing of patient data between operationally diverse systems. Each record, or index entry, in the MPI/MVI contains a small amount of patient data used to identify individual entries. MPI/MVI data is maintained in a centralized, dynamic database located at the AITC that is available to meet multiple information needs across many systems.
Non-VA Provider	Commercial or private-sector physicians, nurses and other medical specialists who use MHV as their primary online method to view a Veteran's Self-Entered Information and Personal Health Record (PHR). Access to information and functionality will vary from that of the VA Provider. Non-VA Providers can act as a Delegate.
Release of Information (ROI) Clerk	VAMC staff member who authorizes the release of information, thereby making the information accessible to a Veteran within their MHV account.
VA Medical Center (VAMC) Administrative Staff	VAMC staff members who operate in an administrative capacity within the VA medical system and MHV.
VA Provider	VA physicians, nurses and other medical specialists who use MHV to access a Veteran's Self-Entered Information and communicate with Veteran's via MHV Messaging. A VA Provider can act as a Delegate.
VA Staff	VA staff that evaluate and measure the effectiveness of MHV in improving the health of Veterans and the quality of provided services.
Veteran	Any person whom has previously served in and been discharged from, other than dishonorably, the United States military, including the Army, Air Force, Navy, Marines and Coast Guard. A Veteran can act as both a Delegating User (for him/herself) and as a Delegate (for other MHV Registered Users).

MHV Use Case Actor	Description of Actions
Veterans Authorization Preferences Interface Improvements (VAPii)	VAPii will allow a veteran to provide authorization for release of health information and other pertinent information between all entities where treatment has been provided or needed. The VAPii application is an effort by the VA to develop electronic forms, in an interview style, that will provide service members, veterans, their families, and advocates with access to information, content, and functionality relevant to the user and the user's benefits from both agencies. Together the VAP/PPP and VAPii along with MVI, the NHIN Gateway Adapter, and Vista Imaging, will create an enterprise-wide electronic solution capable of supporting consumer authorization requirements, consent directives, and organizational policies on privacy and security relative to Release of Information (ROI).
Veterans Service Organization (VSO) Representative	VSO representatives help Veterans understand their VA benefits and can act in their behalf. They function as Advocates and can act as Delegates.
VistA	A rich automated environment that supports day-to-day operations at local VA health care facilities. VistA is built on a client-server architecture, which ties together workstations and personal computers with graphical user interfaces at VA facilities, as well as software developed by local medical facility staff. VistA also includes the links that allow commercial off-the-shelf software and products to be used with existing and future technologies.

Access to the majority of the MRP tools will only be granted to Veterans that are in care or have recently been in care by VA providers and/or completed an episode of VA Mental Health or SUD treatment. It is expected that MRP will facilitate a high level of interaction between the patient and the provider. Additionally, it is envisioned that MRP will be utilized by Veterans' families to be educated about mental health problems, to track progress in treatment, and to provide information potentially useful to treatment and will produce ongoing, quantifiable, reproducible data that can be shared with care providers to provide a more detailed picture of the Veteran's experience between appointments.

The MRP My Goals subsystem will be available to all users that have registered in MHV. It will specifically be designed to support a user community of Veterans both young and old, each with varying levels of computer proficiency. The MRP will be designed for all Veterans with behavioral or mental health concerns, regardless of whether or not they meet Diagnostic and Statistical Manual of Mental Disorders (DSM IV) criteria for a psychiatric disorder.

1.6. Relationship to Other Documents and Plans

This document is closely related to the Requirements Specification Document (RSD), but provides more detail on the MHV application and each environment. The MRP system is described in the following documents and thus has a relationship to these documents.

- My Recovery Plan—Project Management Plan
- My Recovery Plan—Configuration Management Plan
- My Recovery Plan—Requirements Specification Document
- My Recovery Plan—Software Quality Assurance Plan
- My Recovery Plan—Acceptance Criteria Plan
- My Recovery Plan—Communications Plan
- My Recovery Plan—Risk Management Plan

1.7. Definitions, Acronyms, and Abbreviations

See an acronym list [here](#).

1.8. References

See a document list [here](#).

1.9. Style Guide

Graphical and Technical Specification, Veteran's Health Administration (VHA)—MHV User Centered Design—Version 2.0, January 5, 2006

1.10. Methodology, Tools, and Techniques

The MHV program follows mandated VA guidelines and process for product development. Listed below is additional information about programming and design guidelines, engineering practices, frameworks and tools as well as the development cycle.

The Iterative Development Lifecycle (IDL) and ProPath processes defined for the VA OIT will be used for the MRP project. The timeframe for this project will require several iterations. Each iteration will include requirements, design, development, and test cycles.

MHV operates in an agile iterative (sprint) development lifecycle of approximately three weeks per sprint. This means that at any time, there will be functionalities that are in different stages of development from high level requirements to partial implementation. A work plan is implemented that reflects best practices in software development, database administration and user experience development that are particularly well-suited to the MHV core development technologies and will best serve the MHV users. Completed code will be promoted into the production environment and as well as “fixes” required addressing submitted work items.

The software tools or techniques required for performing design documents tasks, (e.g. software for managing work items that can be made during the development phase), are as follows:

IBM—Rational Tools

- IBM Rational Requirements Composer will be used to manage the requirements of this project.
- IBM Rational Team Concert will be used for epics, stories, and defect management.
- IBM Rational Quality Manager will be used for managing the testing cycles.
- A mix of Subversion and Rational Team Concert (CCM) will be used to provide complete software configuration management (SCM) with version control and global development support. Source Control for the Redesign efforts will be managed within CCM. The future plan is to migrate solely to CCM.

1.10.1. **MHV Programming and Design Guidelines**

The MHV Programming and Design Guidelines documents provides a series of rules, guidelines and standards for the various platforms, frameworks, components and tools used in the MHV program. It provides more succinct guidance on specific programming and design steps than documented here in the baseline architecture document.

1.10.2. **Proven Software Engineering Practices**

To support a rapidly growing user base from an estimated total population of around 27 million Veterans, the MHV portal architecture's goals are:

- To have the architecture be scalable, standard, simple, robust, maintainable, modifiable, secure, and auditable
- To have inter-application communications and APIs consistent, coherent, and clear
- To encourage system encapsulation through various layers to minimize tight-coupling, minimize risk and to allow for parallel development
- To encourage high cohesion to reduce the impact of changes in functionality to components

1.10.3. **Frameworks and Tools**

The architecture must be designed to employ frameworks, tools, and cross-cutting architectural mechanisms to enable developers to more efficiently develop and test business logic.

Note: MHV 12.9 includes MHV components that have been reengineered to be consistent and compliant with the MHV TR architecture. This includes upgraded technical stack (e.g. Java 7, WebLogic Server 12c), and updated / upgraded dependencies (e.g. Spring Framework, Apache CXF).

1.10.4. **Iterative Development Lifecycles**

One of the goals of the MHV program is to continually decrease the time it takes to deliver new functions to the end user community. To evolve away from the “big-bang” release paradigm and therefore to support a more iterative development lifecycle, the following architectural best practices will be encouraged across the team:

- Mandate Test Driven Development as a way to rapidly produce feedback of the development effort
- Employ Continuous Builds and Integration as a practice to frequently test and integrate work for the purpose of detecting integration errors as quickly as possible

1.11. User Problem Statement

For the MRP system with Veterans who are currently in, or have recently completed, an episode of VA Mental Health or SUD treatment require alternative tools of managing their mental health and well-being. These tools ideally would actively engage the Veteran in identifying and tracking progress toward their goals and yet maintain confidentiality, security, convenience, ease of use, and 24 hour/7-day access.

1.11.1. User Objectives

The Veterans' requirements for the new system are focused on collaborative ways to be more actively engaged or involved in one's own mental health care, with the ultimate objective of improving one's health and quality of life. MHV will allow Veterans to engage in a program of guided self-change, which will draw upon empirically supported psychosocial treatments (e.g. Cognitive-Behavioral Therapy).

Specifically, the MRP My Goals subsystem shall enable the Veteran/user to:

- Self-identify larger life goals
- Break goals down into manageable components
- Track progress towards those goals
- Identify and overcome barriers to goal attainment
- Identify strengths and goal rewards
- Generate goal reports

2. Background

The MHV Capabilities Enhancements Project will provide stability to the MHV platform and support future growth and enhancements required by the business. In order to accomplish this task, the MHV project team will collaborate with the AITC to plan the re-architecting of the currently non-supported platform. Tasks will include:

- Creating pre-production and development environments that mirror the production environment, ensuring that both use supported hardware and software and are compliant with IdM processes
- Providing applicable compliant technical environments required to enable the portal's anticipated growth and development while providing the highest level of application support
- Implementing automated testing tools that are currently not in place

- Updating and stabilizing the platform, which will further enable the planned future growth and enhancement of the portal

These environments and tools will decrease the time needed to test new portal features and applications and lower the risk of not meeting performance requirements in production.

The main objective of the MRP subsystem project is to create an interactive set of web-based tools that will allow Veterans who have behavioral or mental health concerns to track important aspects of their self-care and professional care. MRP will:

- Be useful to all Veterans with behavioral or mental health concerns, regardless of whether or not they meet DSM-IV criteria for a psychiatric disorder
- Provide the Veteran with hope of improved health and quality of life through collaborative involvement in his or her own mental health care
- Be described on the open MHV site, but access will require a special entry key to be set up by the provider
- Follow a Concurrent Recovery Monitoring model, in which regular outcomes monitoring is incorporated as a routine part of clinical care

The MRP My Goals subsystem is one of these tools, and the high level business functions of the My Goals subsystem are as follows:

- Goal Creation
- Goal Management
- Goal Reporting

The MRP application will be fully integrated with the MHV portal, with the MRP My Goals subsystem made available within the MHV Track Health module. The MRP My Goals subsystem design will be aligned with the overall MHV architecture, and its look and feel will adhere to MHV and MRP web style guidelines. Design efforts will also ensure that the target population's accessibility needs are identified and incorporated into MRP My Goals subsystem designs, per Section 508 accessibility guidelines.

2.1. Overview of the System

The functions of the MHV Portal can be grouped into the following general categories:

- Services for the User: These are portlet applications that provide functionality to the user
- Personal Health Record (PHR) Extraction Services for the Users: Services for establishing a local cache of the user's PHR.
- Information for the User: Health and other related information important to the user
- Application Server Administration: Administrator functions to support application configurations and user administration
- System Administration: These are functions that support the operations of the MHV Portal

MRP Current mental healthcare business processes are the result of five primary factors:

- The legacy of mental healthcare processes over the past three decades
- The movement of care for certain mental disorders from specialty care to primary care
- A shift in paradigms from “treatment” to “recovery
- The firm grounding of recovery efforts in evidence-based treatment practices
- The increasing use and integration of information technology to advance mental healthcare

These fundamental changes to mental healthcare processes have emerged over the past decade and are embodied by the VHA FY 2003–2008 Strategic Plan and the Department of Veterans Affairs VHA Handbook 1160.01 Uniform Mental Health Services in VA Medical Centers and Clinics.

There is a movement of treatment from specialty care to primary care, and it is most evident for Veterans with depression. Initial care for SUD is also shifting increasingly to primary care facilities. A similar, but less pronounced, shift is also underway for treating PTSD. Specialty care for these disorders now focuses more on chronic and other more difficult cases.

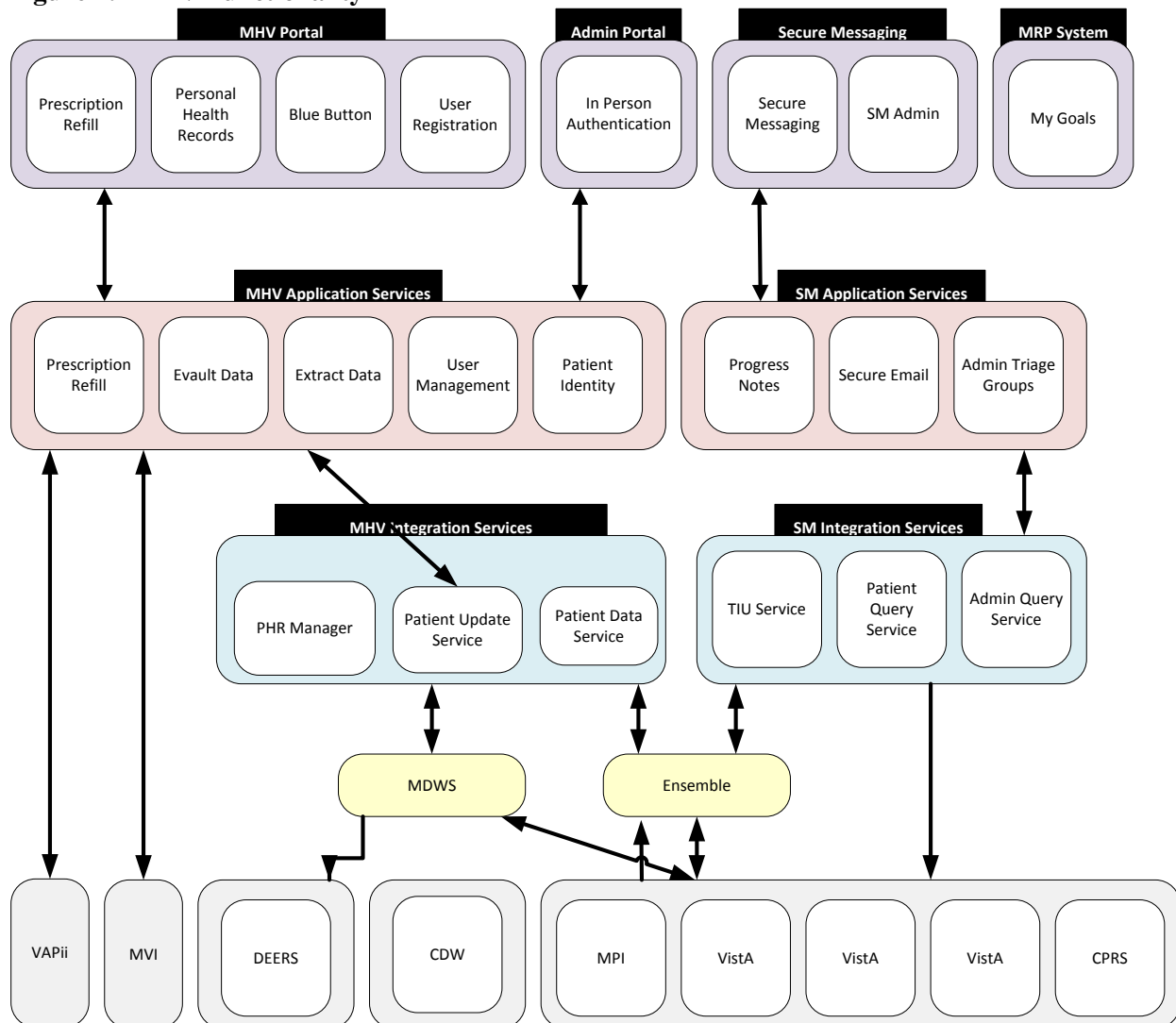
The shift in mental healthcare paradigms from “treatment” to “recovery” is profound but emerging. Under the treatment paradigm, which has been clearly defined by the VA’s mental healthcare leaders, mental health providers diagnose a patient, develop a treatment plan for that patient, negotiate revisions to that plan if the patient balks, and then implement the plan with a focus on treatment. Under the recovery paradigm, both the provider and the patient focus on the results of the treatment, rather than the treatment itself.

The MRP My Goals subsystem design addresses this recovery paradigm by allowing the Veteran to take greater “ownership” over his/her recovery, allowing him/her to identify goals and track his/her progress towards achieving those goals. The Veteran with behavioral or mental health concerns will be the primary user of the MRP My Goals subsystem, as well as the provider who will be able to better understand what is important to the Veteran and ensure that treatment plans complement and support the Veteran’s goals.

2.2. Overview of the Business Process

The following figure displays the MHV functionality.

Figure 1: MHV Functionality



The following table describes the above diagram and provides details about it.

Table 5: Business Processes

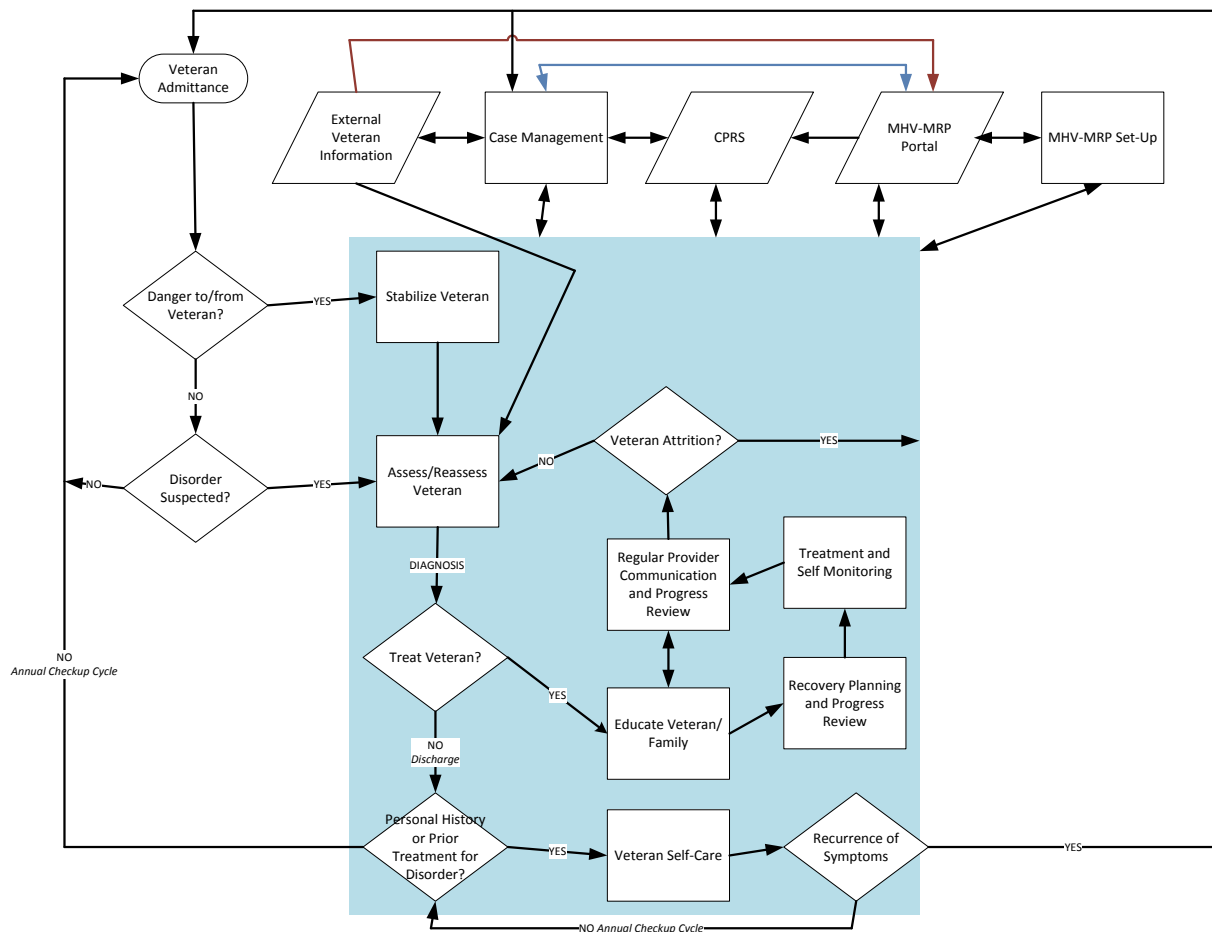
Business Process ID	Business Process Name	Type	Owner	Description
1.	User Registration			To register a patient in My HealtheVet application
2.	User Manage Profile			To manage the user profile.

Business Process ID	Business Process Name	Type	Owner	Description
3.	User Management—In Person Authentication			To Authenticate a patient in order to have that user access to his personal health record.
4.	Rx Refill Management			To retrieve prescription refill detail information for a patient from VISTA.
5.	PHR Data Extract			To Extract personal health record of a pertinent from external systems.
6.	Manage VA Staff			Administrative functionality to manage VA Staff.
7.	Manage SM/MHV Users			Administrative functionality to manage MHV user's access. SM/MHV is a controlled environment
8.	Perform Self Entered Information			To add/modify/delete self-health information.
9.	Blue button			To retrieve patient data with a single click and to provide the functionality necessary for a patient to share their CCD with a participating health care provider.
10.	Eauth			The goal of this project is to implement an interface with the VA E-Authentication infrastructure where My HealtheVet can accept a level 2 credential from a Credential Services Provider (CSP) and use that credential to authenticate and authorize the associated My HealtheVet user. When a new CSP is available, MHV will automatically accept the credentials from the new CSP. MHV validates the credentials from the CSP and also validates against MVI to make sure that MVI repository contains one and only patient record for the given credentials.
11.	MRP			See below.

Through the MHV and MRP, both the patient and the provider collaborate to develop the recovery plan; both focus on the results reported by the patient and observed by the provider; based on these results, the patient participates more actively in managing and revising the recovery plan to fit needs; and the provider helps the patient implement his/her own recovery plan. Stronger patient “ownership” over the recovery process will improve the quality of mental healthcare and reduce the large proportion of Veterans withdrawing from care. The major opportunities for MHV-MRP to support the recovery paradigm are shown in blue in the diagram in the following figure.

The below figure shows the Mental Health Process Overview.

Figure 2: Mental Health Process Overview



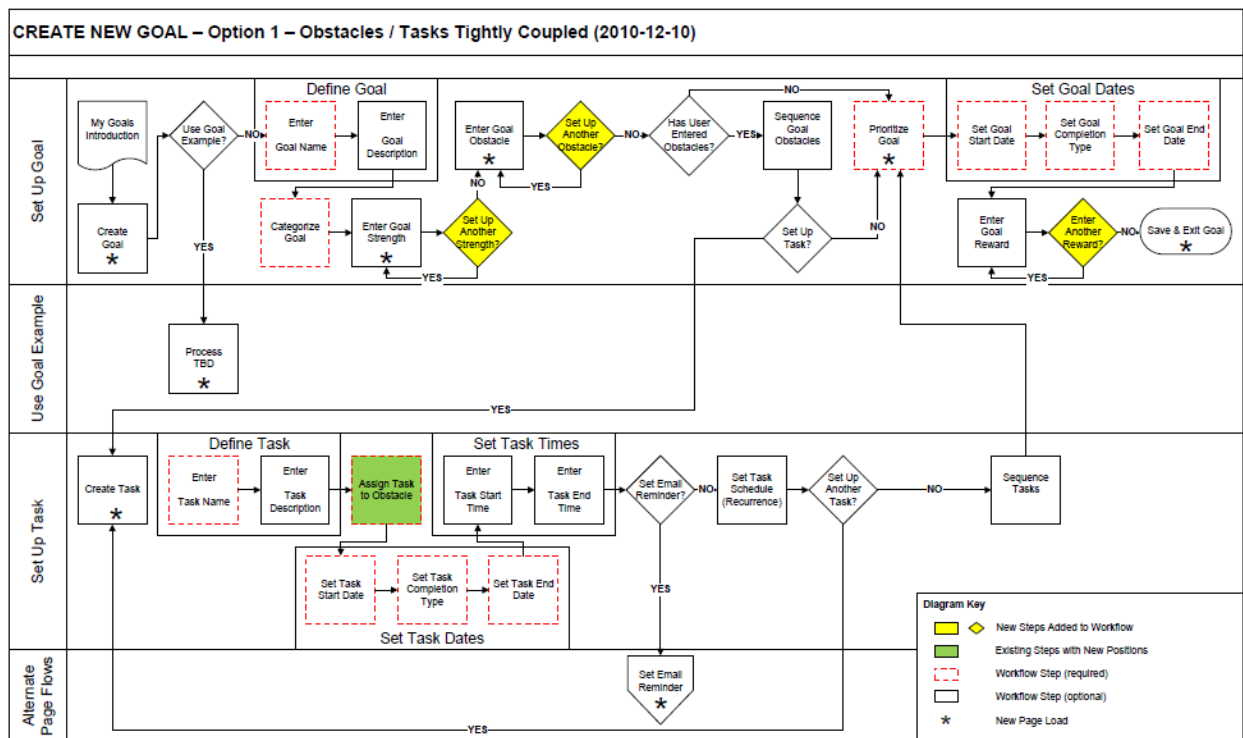
The My Goals section is a key component of MRP that will provide Veterans with the ability to Set, monitor, and track goals and supporting tasks that are individually important to their recovery.

- MRP user will navigate to the Goals section through the dashboard page or by clicking the left navigation element “Goals.” Once there, they will set up/view/edit/delete/prioritize goals, track and view goal reports, and add/edit/delete/view personal and VA success stories using the control panel on the right.

- Quick Links/Related Links will be provided wherever appropriate within the Goals section
- MRP users will click on the Track Goals links to enter and track goal-related information. Additionally, they can view the entered and tracked goal data by clicking the Previous and Next Week links displayed in the Track screen.
- Graphs or Text-Only reports will be provided for the MRP users to view the progress of tasks and goals

The Create a Goal process is detailed in the workflow below.

Figure 3: Create a Goal Process



The above figure represents the Create a Goal Process.

2.3. Business Benefits

As a HealthVet-VistA application being developed under the VA's evolving service oriented architecture, the MHV Portal architecture will perform the following to ensure enterprise requirements are met:

- Engaging the emerging HealthVet-VistA application teams, as they are created, to define needs and features
- Engaging legacy application teams to ensure that MHV will, at a minimum, support current functionality
- Engaging the enterprise architecture and management teams to proactively define requirements

- Engaging the VHIM team, who are authoritative for defining the data content model, during model and message content development
- Engage the VHIT ARB and CCB to address compliance issues with the TRM and extended tool list
- To implement current functionality and to accommodate future requirements, use existing and planned systems and services such as VistA, MPI/MVI, Person Lookup, Pharmacy Reengineering, ESR, Security Services, and so on
- Integrating MRP features into MHV, and allowing Veterans to participate in goal-setting and active use of evidence-based problem management information
- Producing ongoing, quantifiable, reproducible data that can be shared with care providers to provide a more detailed picture of Veterans' experiences between appointments
- Providing a tool that also can be used to educate Veterans' families about mental health problems, track treatment progress, and provide potentially useful treatment information

The business benefits of developing the MRP My Goals subsystem are as follows:

- Empowers Veterans with tools to take an active, collaborative role in their treatment
- Encourages Veterans to take responsibility for their own recovery
- Allows Veterans to monitor and track their progress toward self-identified goals
- Helps Veterans identify their strengths and overcome barriers to goal attainment
- Encourages health care team providers to make sure the treatment process complements and supports the Veterans' goals
- Helps health care teams better understand what is important to the Veterans (in terms of goals)
- Helps Veterans set goals and look beyond their illness
- Helps the therapeutic process

2.4. Assumptions

Assumptions are factors internal to the environment that expand alternative choices for solving any issues identified in the business process review and influence the system design. Constraints also impact system design in that they refer to limitations on the conditions under which the system is developed.

2.4.1. Design Assumptions

The following assumptions were gathered during interviews with SMEs (SMEs) and vetted with the MHV IT Program Office or related SMEs for feasibility in the VA setting:

- The MHV site is currently being redesigned by contractors employed by the MHV Information Technology (IT) Program Office. A Heuristic Evaluation Report will be employed in this redesign to meet the needs of cognitively impaired individuals using the site. MRP will follow the MHV site design.

- MRP/MVI will reside within the proposed redesign; however, it will be granted left navigation area space specific to MRP and My Goals, fundamental features supporting MRP/MVI
- Prior to implementation of any interactive courseware, a VA-approved Learning Management System (LMS) will be deployed within the architecture and meet the fundamental requirements of MRP outlined
- All features and functionality of MRP must be reviewed and approved by the Clinical Advisory Board (CAB) prior to implementation
- The MHV Development team slates the features and functionality outlined in the use cases and requirements for development in Fiscal Year 2013
- Features and functionality of MRP can be rolled out in a phased release schedule as long as they follow the required dependency guidelines for said requirements
- The MHV Site Map will be updated to reflect the features and functionality of MRP components

2.4.2. Design Constraints

The following constraints impacted the range of MRP design choices:

- The 12.9 release of MHV includes components that have been reengineered and are consistent with the MHV Tech Refresh architecture
- All MRP functionality will be designed to operate within the MHV existing portal framework, using applicable software development languages, and architecture
- MRP will support Veterans with cognitive disabilities
- MRP shall comply with MHV technical requirements with regard to development and implementation and release constraints
- MRP shall comply with MHV data requirements with regard to data types, entry constraints, length and character constraints, as well as database design and content

2.4.3. Design Trade-offs

The following design elements were selected as critical (key design trade-offs) to meeting the objective of support MHV program efforts for the coming decade.

- Open Source versus Commercial Off-the-Shelf (COTS) versus Custom Built Solutions – The design team has determined that basing the new MHV Site Redesign architecture on open source systems, components, and frameworks that have strong community support and that are actively developed will provide the lowest cost, lowest risk, and will support shorter delivery cycles when compared with either COTS-based or custom built solutions. It is understood that in some cases support agreements may be required in order to receive the level of support required to operate open source solutions within the VA production environment. These types of agreements may cause the program to incur additional costs but care will be taken when selecting technologies to ensure that these costs are understood by program management up front and that the support costs are

reasonable. An exception to the rule of preferring open source to COTS or custom built solutions is in ensuring alignment with the standard technologies that VA utilizes throughout the enterprise. It is assumed that in cases where licenses have already been procured and infrastructure has been put into place for operating and maintaining COTS products, it would be lower cost to the agency to select an agency standard technology as opposed to selecting a technology that is open source and that has zero licensing costs. Finally, custom built solutions are considered a higher risk, higher overall cost, and expected to take more time to implement than either open source or COTS products. Thus custom built solutions shall only be utilized if no viable open source or COTS product is available to fulfill a particular requirement. Custom built solutions shall be used to implement application logic that is specific to MHV processes.

- **Web and Web-Service Standards versus Proprietary Techniques and Technologies** – The design team has determined that a standards based approach for integration shall be employed where standards exist. It is expected that through implementation of standards based technologies MHV systems will support an increased level of interoperability. Standards make the web better for users, developers, and operations personnel. Additionally, when systems are built on standards, the risk of projects failing as well as high maintenance costs is reduced. Tables that include the web, web-service, and VA specific standards to be observed by the architecture are included in Appendix B below.
- **Horizontal versus Vertical Application Scaling** – The design team has determined that the MHV Site Redesign architecture shall be designed to run on clusters of commodity servers in order to support horizontal application scaling. This decision is based on our understanding of the Federal Government’s “Cloud First” policy to utilize cloud based infrastructure to reduce IT costs across the government as well as our understanding that MHV management plans on deploying MHV applications to a private cloud at multiple Terramark data centers. Additionally, vertical scaling would require the acquisition of larger capacity systems to handle increasing load on the systems. It would also require complex analysis and estimates in order to adequately size the system being procured to ensure that the system would have adequate capacity to meet increasing system demands. Additionally, horizontal scaling typically doesn’t fully support the concept of system elasticity, in that you don’t really grow and shrink system capacity as the system loads change over time. The procured big-iron system has the capacity that it has, with no growing or shrinking as system demands change. Designing an application to scale horizontally does have downstream impacts in application and database design. For instance, if an application is stateful, care must be taken to ensure that state can be communicated across the cluster. Additionally, any database solution and data design implemented must be capable of taking advantage of new system resources. This is typically implemented through a combination of database server clustering, data sharding, and data replication.

2.5. Overview of the Significant Requirements

This section serves to inform non-project personnel reading this document as to the basis for the MHV/MRP design.

2.5.1. Overview of Significant Functional Requirements

For the MRP My Goals subsystem, the functional requirements are captured in the system use cases. This use case model is composed of the following use cases:

- MRP-01 View My Goals Introduction
- MRP-02 Create My Goal
- MRP-03 Create My Goal Using Example
- MRP-04 Update My Goal
- MRP-05 View My Completed Goals
- MRP-06 Track My Goals
- MRP-07 Generate My Goals Report

Each use case corresponds to a major function of the MRP My Goals subsystem and will consist of a set of actors, a basic flow of events, as well as alternate and exception flows, and business rules. Each use case will correspond to a subset of My Goals wireframes. Subsequent design, development, and testing activities will trace back to these primary use cases and wireframes.

2.5.2. Overview of Functional Workload/Performance Requirements

Below is historical activity of the production activity of the MHV Portal in terms of “hits” to the national portal. A hit presents a user requesting a page on the site. The target performance is to support 24K transactions (clicks), or 8 per second, and 50 hits per second.

The table below shows hits across just Rx Refill functionality as well as MHV as a whole. Current performance in terms of hits is just below the target value.

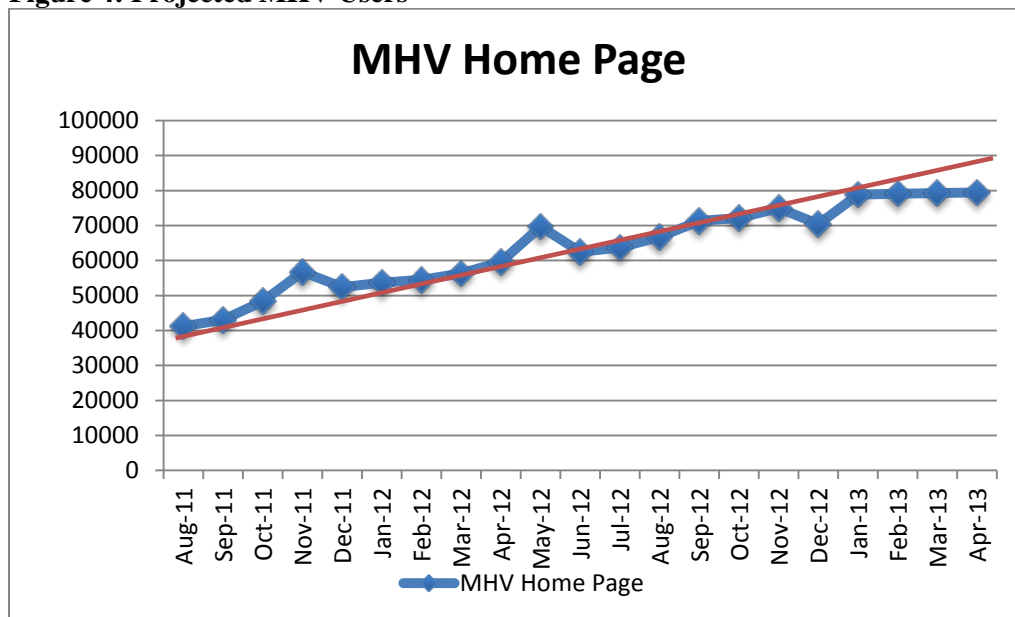
Table 6: Prescription Refill Hits over Time

Page	No Of Hits on Aug 2011	No Of Hits on Nov 2013	Percent Growth
MHV Rx refills Page	20236	24560	21.3 %
MHV Refill Prescription Page	22228	38494	73.1%

The MHV portal will be used to access the MRP My Goals application. As indicated from the tables below, there is a projected significant increase in MHV and Registered Users—IPA usage from FY12 through FY17.

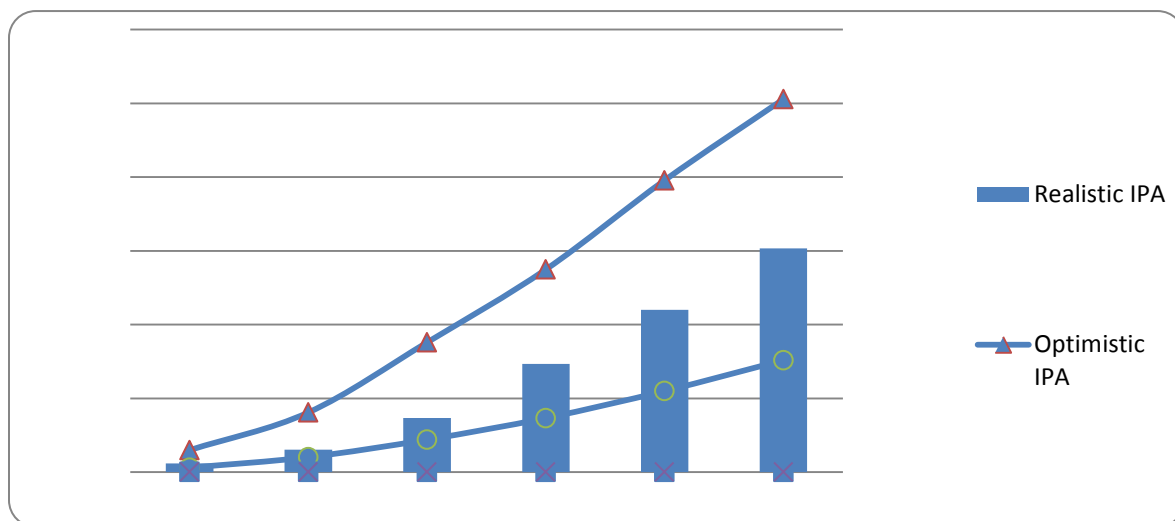
The below figure represents the projected MHV users over the next several fiscal years.

Figure 4: Projected MHV Users



The figure below represents the Projected IPA Users over time.

Figure 5: Projected IPA Users



MHV statistics as of April 2014:

Over 39 Million visits

- 2.7 Million Registered Users (2.4 million Veterans)
- About 1.5 million VA Patients with Premium Accounts
- More than 156,000 self-enter data
- Over 15.4 Million VA prescription refills since August 2005

The MRP My Goals subsystem design is governed by MRP performance requirements listed in the following table.

Table 7: Workload and Performance Requirements

ID	Requirement
PERF01	Response Time—The system shall respond for all transaction requests with an average response time of 3 seconds
PERF02	Capacity—The system shall support 1,000 simultaneous users in the pilot implementation, with up to 300,000 simultaneous users in national roll-out.
PERF03	Throughput—The system shall support 900,000 transactions per second.

2.5.3. Overview of Operational Requirements

The MHV design is governed by operational requirements listed in the following table.

Table 8: Operational Requirements

ID	Requirement
OPS01	MHV Availability—MHV availability depends upon AITC availability.
OPS06	Resource Utilization—The resources utilized by the user population (memory, disk, Communications, etc.).
OPS07	Usage Growth—The system shall support a minimum of 1,000 users in the first year of operation under the pilot implementation. Subsequent national roll-out shall support a user population of up to 300,000 individuals.
OPS11	Maximum Transactions per Page—The system shall limit the number of transactions on any single page to 900.

2.5.4. Overview of the Technical Requirements

The MHV/MRP subsystem shall comply with MHV technical requirements with regard to development and implementation and release constraints. The MHV/MRP subsystem shall also comply with MHV data requirements with regard to data types, entry constraints, length and character constraints, as well as database design and content.

Other relevant MRP-specific technical requirements are listed in the following table.

Table 9: Technical Requirements

ID	Requirement
INT03	Accessibility, General—MHV will meet all Priority 1 and 2 checkpoints of the Section 504 and 508 Web Content Accessibility Guidelines.

ID	Requirement
INT04	Accessibility for Cognitively Impaired Veterans—MHV/MRP will meet all guidelines discussed in the Heuristic Evaluation Report and in the disorder specific patient profiles developed by the Patriot UI/UX team. The information provided in MRP should be at a level that moderately impaired poly-trauma and/or traumatic brain injury patients can access and understand the system.
INT05	Windows Browser Compatibility, Microsoft Internet Explorer—MRP shall properly render and be fully functional in both Microsoft Windows and Apple Macintosh (where applicable respectively): <ul style="list-style-type: none"> • Internet Explorer browser, versions 5.5—7.0 • Mozilla’s Firefox browser, versions 1.5.0.1 and higher • Netscape’s Navigator browser, versions 7.2 and higher • Freedom Scientific’s JAWS for Windows browser, versions 5.0 and higher • GW Micro’s Window-Eyes browser, versions 4.5 and higher • Apple’s Safari browser, versions 1.2 and higher
INT13	MRP Help Files—All MRP screens will provide screen-level help.
INT14	Help Text for Control Elements—MRP will provide help text for the most common presentation and control elements.

The MRP My Goals project is in compliance with the OED Software Engineering TRM and Enterprise Architecture.

2.5.5. Overview of the Security or Privacy Requirements

The MRP My Goals project is in compliance with IPRM Certification Program Office. Specifically, the project:

- Is in compliance with the VA Certification and Accreditation (C&A) process
- Has properly documented within the PMAS artifacts how compliance with the C&A process will be maintained

2.5.6. Overview of System Criticality and High Availability Requirements

The approach that will be taken to provide the required level of availability and disaster recovery is that the MHV will take advantage of the disaster recovery mechanisms currently in place at the AITC. The approach will include a workload distribution scheme to support high availability for a group of geographically dispersed users.

The MHV project is in compliance with Enterprise Infrastructure Engineering as it pertains to System Criticality and High Availability Requirements. Specifically:

- The project in compliance with IT Infrastructure Standards
- The project’s infrastructure been ratified through the Technical Analysis Review-Technical Analysis Summary (TAR-TAS) process

2.5.7. Single Sign-on Requirement

N/A

2.5.8. Requirement for Use of Enterprise Portals

N/A

2.5.9. Special Device Requirements

The MHV design shall comply shall comply with all Federal Accessibility regulations—
Currently falling under Section 508 of the U.S. Rehabilitation Act (29 U.S.C. § 794d). Full
details on requirements for compliance with Section 508 can be found at www.508online.com.
In relation to device requirements, the MHV design shall support the requirement that when
pages utilize scripting languages to display content, or to create interface elements, the
information provided by the script shall be identified with functional text that can be read by
assistive technology. § 1194.22 (1)

2.6. Legacy System Retirement

Not applicable.

3. Conceptual Design

This MHV system can be categorized as data driven application supplied by CDW, VistA, and in
the case of MRP CPRS. It is also considered to be a Self-Entered Information or SEI within the
MHV Portal application. All information to be stored within eVault is provided by the user; there
are no integration points with outside systems. The primary design goal is to follow the existing
design patterns for creating new SEI modules by utilizing the Atlas framework for the generation
and maintenance of the persistence and service tier. The user interface components will follow
similar design patterns with the exception of any unique functionality introduced in this module.

Where possible the domain objects have been generalized into common objects for reusability.

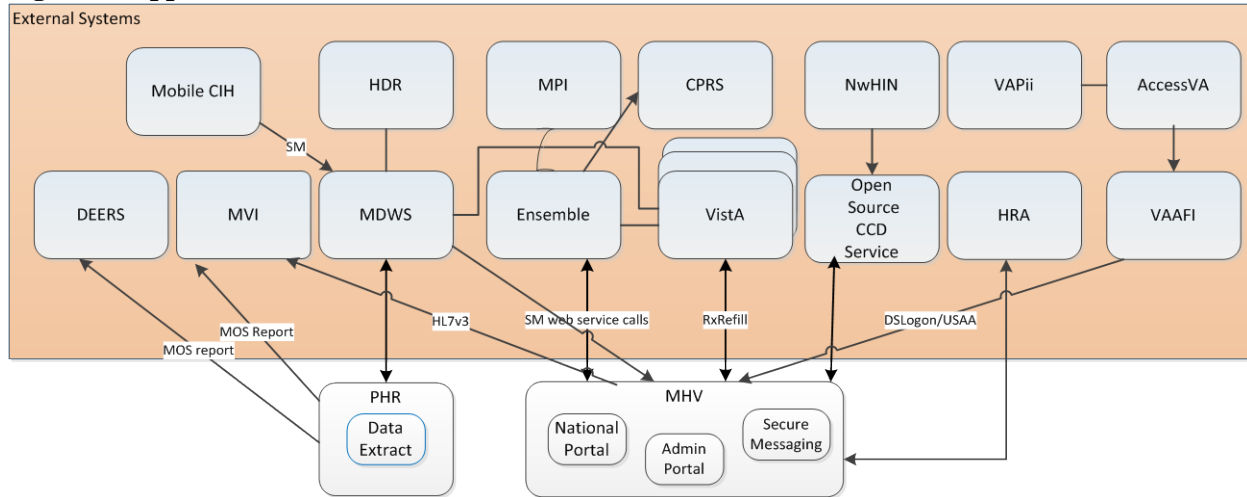
3.1. Conceptual Application Design

The following sections cover the conceptual design of the MHV Release 12.9.

3.1.1. Application Context

The following figure details the application context for MHV:

Figure 6: Application Context for MHV



Application Context Description is listed in the following table.

Table 10: Application Context Description

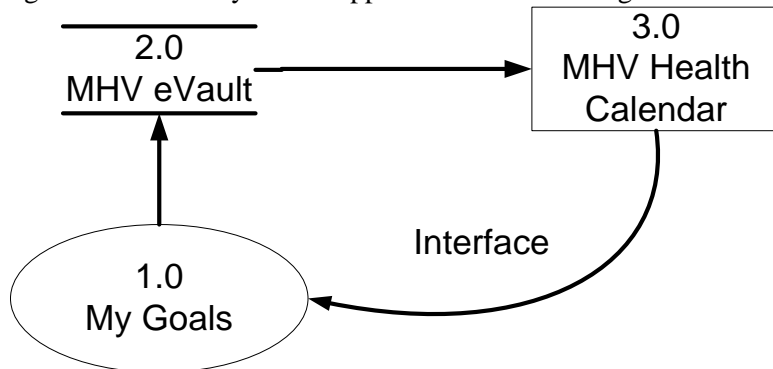
	Objects				
ID	Name	Description	Interface Name		Interface System
					[Systems with which this one interfaces]
	Interfaces External to OI&T				
ID	Interface Name	Related Object	Input Messages	Output Messages	External Party
	Master Veteran Index (MVI)	Military Service Information	MHV user_profile_id, see PHR Log	Runtime values, no new records, see PHR Log	
	Interfaces Internal to OI&T				
ID	Interface Name	Related Object	Input Messages	Output Messages	Other CBP Party
	HDR	N/A	Request for patient data using id	PHR Data records	
	Medical Domain Web Service (MDWS)	N/A	Request for patient data stored in VistA	PHR Data records and non-PHR	
	Ensemble	N/A	Patient Id and various parameters VistA Clinics requests VistA TIU Titles request VistA Provider requests	Status or Data records	
	MPI/MVI	Ensemble	Patient ID and various parameters	Status or Data records	
	VistA	N/A	RxRefill requests	Data records or status	
	VistA	Ensemble	SM requests	Data records or status	
	Open Source Continuity of Care Document (CCD) Service	N/A	Request for C32 document retrieval status using patient ICN and C32 xml document using ICN and date/timestamp.	C32 Document retrieval status C32 xml document	

ID	Objects				
	Name	Description		Interface Name	Interface System
	CCD Direct		Request a C32 document be delivered securely through an interface with the VA Direct service	Confirmation of successful delivery of C32 xml document	
	HRA HealtheLiving Assessment		Authenticate and establish a session for transfer of a user to the HealtheLiving application	Session information and SEI data	
	VAPii	eRAR form	Request the submitted eRAR form status. Request the submitted eRAR form. Access and sign eRAR form	Status Form Web Interface	

Note: the following table refers to all other objects in the diagram except the system to which this design applies, represented by a single object (typically in the center of the diagram).

The following diagram shows the context within which the applications exist.

Figure 7: MRP—My Goals Application Context Diagram



The following table describes the information in the Application Context Diagram. Note: the following table refers to all other objects in the diagram except the system to which this design applies, represented by a single object (typically in the center of the diagram).

Table 11: Application Context Description

Objects					
ID	Name	Description	Interface Name	Interface System	
2.0	MHV eVault	MHV eVault is the MHV's data store for each user that constrains their Personal Health Records. Self-Entered Goal Data is stored in the MHV eVault.	N/A	N/A	
3.0	MHV Health Calendar	This is the Health Calendar function that is available in the MHV portal. Health Calendar can be used to view the all the goal tasks that are due. It displays the information in a Daily/Weekly/Monthly and list views.	Task Details	My Goals—Tasks	

Interfaces External to OI&T					
ID	Interface Name	Related Object	Input Messages	Output Messages	External Party

Interfaces Internal to OI&T					
ID	Interface Name	Related Object	Input Messages	Output Messages	Other CBP Party

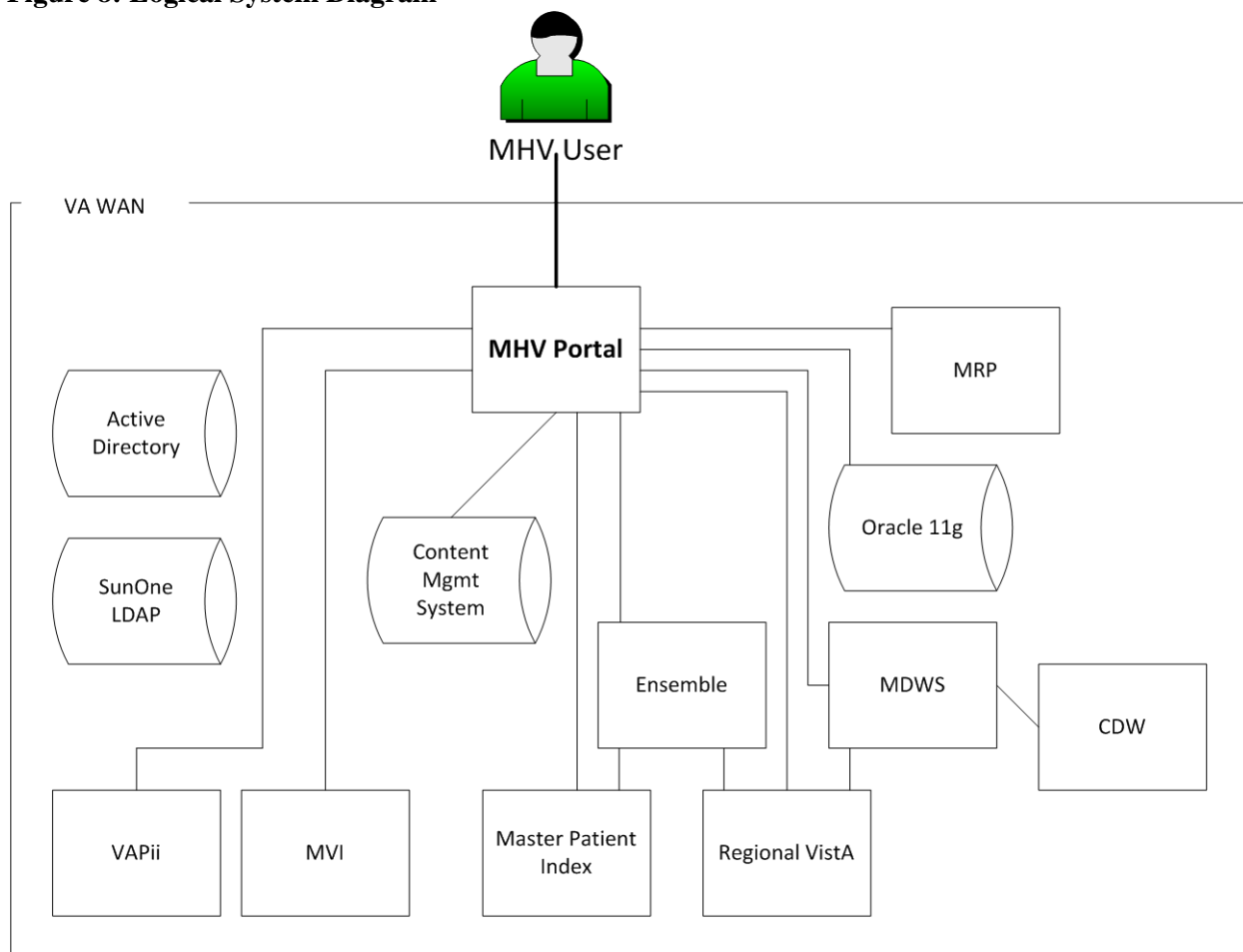
Externally Shared Data Stores					
ID	Name	Data Stored	Owner	Access	

3.1.2. High Level Application Design

The MHV Portal comprises multiple systems in a system of systems architecture. MHV will follow service tier design patterns consistent with the current production Architecture which is based on Atlas. MHV introduces no new internal or external dependencies with this subsystem; User Management provides the key dependencies, Web logic Portal, and Content management systems.

The main elements of this system can be seen in the logical system view in the following diagram.

Figure 8: Logical System Diagram

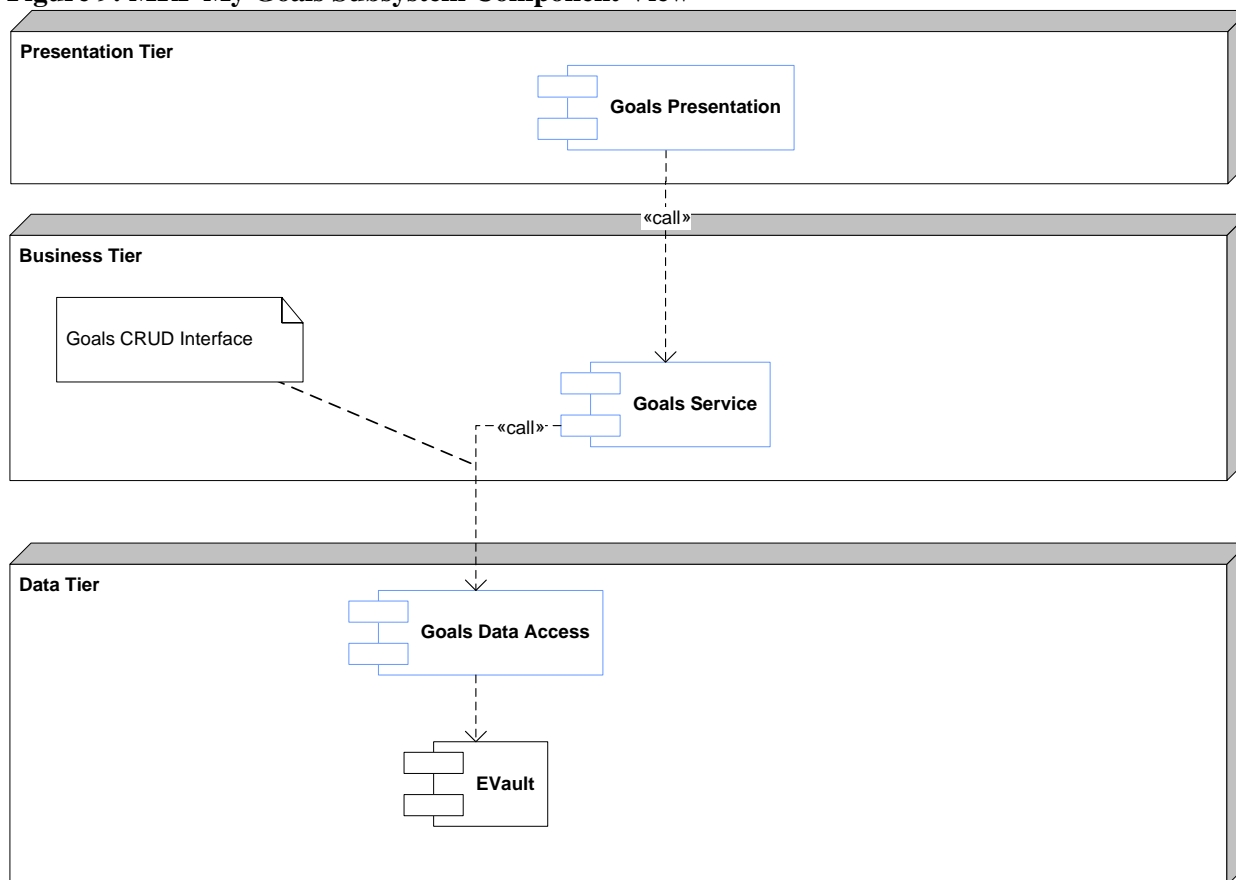


Three external execution environments exist that MVH Portal uses for certain functionality: The VistA systems that exist at the regional VISNs, the MPI/MVI and VAPii. These support the usage of medical information for Veteran prescriptions and personal health records and the ability to upgrade the user account online. Connection to these systems is socket to socket in the current implementation. Once the MHV provides the MHV Delivery Services message broker, the team will replace the socket connections.

The following figure displays the MRP My Goals Subsystem high level application design is depicted in the form of a component view in the following figure.

The core presentation infrastructure is shown in the following image.

Figure 9: MRP My Goals Subsystem Component View



The core presentation infrastructure is designed around the summary and detail views for Goals and Tasks, there are also a specialized view for managing Task Actions. Within the Goals and Tasks views these there are two generalizations, Goal Extensions and Task Extensions. Goal Extensions include Goal Rewards, Goal Strengths and Goal Obstacles, likewise the Task Rewards and Task Strengths. These generalizations provide a common framework and pattern for maintaining their associated domain objects.

Within MHV this presentation tier is unique in several ways; instead of printer friendly pages the user will have the ability to view and print reports. Also, there are many confirmation steps and messages that result in choices; each type of interaction is shown in the page flows at the detail level.

Field Level Access Control

‘My Goals’ top level page in the MHV portal application will be restricted to ‘My Goals’ field testers. This restriction is enabled in order for the field testers to test the application once it goes to field testing. The field test group for My Goals is named ‘My Goals’ in ACCESS_DOMAIN table and also added to ACCESS_FEATURE table in the database. The java UI code for this functionality lives in PHR code base (where we have all field level access control code). Starting from Sprint 4, we would need to change the order of the build, where we build mhv_phr first and

then mhv_mrp as MRP would be dependent upon PHR code base for field level access control functionality. Please refer to section 2.4.1 for database scripts. The scripts are checked into subversion:



Objects in the MRP My Goals Subsystem Component View are listed in the below table.

Table 12: Objects in the MRP My Goals Subsystem Component View

Name	Tier	Description
Goals Presentation	Presentation	Presentation layer for the user.
Goals CRUD Interface	Business	Create, Read, Update and Delete business rules are applied.
Goals Service	Business	Business rules are applied.
Goals Data Access	Data	Data retrieval from the database.
EVault	Data	Data store for the user's My Goals data.

3.1.3. Application Locations

The following table lists the application locations for MHV:

Table 13: Application Locations

Application Component	Description	Location at Which Component is Run	Type
MHV Portal		AITC	Presentation Logic
SM Application		AITC	Presentation
LDAP		AITC	Data Logic
System Database		AITC	Data Logic
PHR Manager	Application used for extracting and loading PHR data for patient into eVault.	AITC	Integration Tier
MHV API Façade	Application used for aggregating MHV services (APIs) and providing centralized authentication, authorization, auditing, and error handling.	AITC	Interface
Ensemble		AITC	Interface
MDWS		AITC	Interface
HDR		AITC	Interface
Goals Presentation	Presentation layer for the user	Available via Web browser.	Presentation Logic

Application Component	Description	Location at Which Component is Run	Type
Goals CRUD Interface and Goals Service	Business Tier of the application	AITC	Business Logic
Goals Data Access and EVault	Data tier of the application	AITC	Data Logic
Open Source CCD Service	Middleware for retrieval of CCD document from NWHIN	AITC	Interface
CCD Direct Service	Secure message delivery service for delivery of C32 xml documents	AITC	Interface

3.1.4. Application Users

The following table details the application users:

Table 14: Application Users

MHV Users	
Actor	Description of Action
IPA User	Conduct in-person authentication procedures, and provide Veterans online access to their VA personal health information.
MHV Registered User	All users who have established a personal account in the MHV system. Registered users are given access to system functionality that is unavailable to unregistered users.
MHV Unregistered User	Users who have not logged into the MHV system. Unregistered users can access all publicly available content, such as the health information library, yet they do not have access to the system's core functionality, such as viewing and managing health records, making entries into health logs, designating and acting as a Delegate, refilling prescriptions, and viewing appointments. If an Unregistered user has not created an account, the User can register at any time. Once this account has been created, the Unregistered User can login at any time.
Veteran	Any person whom has previously served in and been discharged from, other than dishonorably, the United States military, including the Army, Air Force, Navy, Marines and Coast Guard. A Veteran can act as both a Delegating User (for him/herself) and as a Delegate (for other MHV Registered Users).
VA Provider	VA physicians, nurses and other medical specialists who use MHV to access a Veteran's Self-Entered Information and communicate with Veteran's via MHV Messaging. A VA Provider can act as a Delegate.

Actor	Description of Action
Non-VA Provider	Commercial or private-sector physicians, nurses and other medical specialists who use MHV as their primary online method to view a Veteran's Self-Entered Information and Personal Health Record. Access to information and functionality will vary from that of the VA Provider. Non-VA Providers can act as a Delegate.
VAMC Administrative Staff	VAMC staff members who operate in an administrative capacity within the VA medical system and MHV.
ROI Clerk	VAMC staff member who authorizes the release of information, thereby making the information accessible to a Veteran within their MHV account.
VSO Representative	VSO representatives help Veterans understand their VA benefits and can act in their behalf. They function as Advocates and can act as Delegates.
MHV Administrator	VA staff responsible for managing and maintaining the MHV system.
MHV Help Desk	VA staffs who supports users of MHV by responding to comments and questions submitted to the help desk.
VA Staff	VA staff that evaluate and measure the effectiveness of MHV in improving the health of Veterans and the quality of provided services.
HDR	HDR refers to the national instance of Health Data Repository. MHV will provide read access to the HDR through its middleware messaging framework.
MPI/MVI	MPI/MVI is a central index of unique MHV users. MHV users Health Administration's implementation includes assigning each patient an ICN and a Coordinating Master of Record (CMOR) site. The ICN assignment enables the sharing of patient data between operationally diverse systems. Each record, or index entry, in the MPI/MVI contains a small amount of patient data used to identify individual entries. MPI/MVI data is maintained in a centralized, dynamic database located at the AITC that is available to meet multiple information needs across many systems.
VistA	A rich automated environment that supports day-to-day operations at local VA health care facilities. VistA is built on a client-server architecture, which ties together workstations and personal computers with graphical user interfaces at VA facilities, as well as software developed by local medical facility staff. VistA also includes the links that allow commercial off-the-shelf software and products to be used with existing and future technologies.
Goals Presentation	Available via web browser.

Actor	Description of Action
VAPii	VAPii will allow a veteran to provide authorization for release of health information and other pertinent information between all entities where treatment has been provided or needed. MHV users can upgrade their account online via VAPii. Additionally, MHV users can view the signed eRAR form that they signed via VAPii.

3.2. Conceptual Data Design

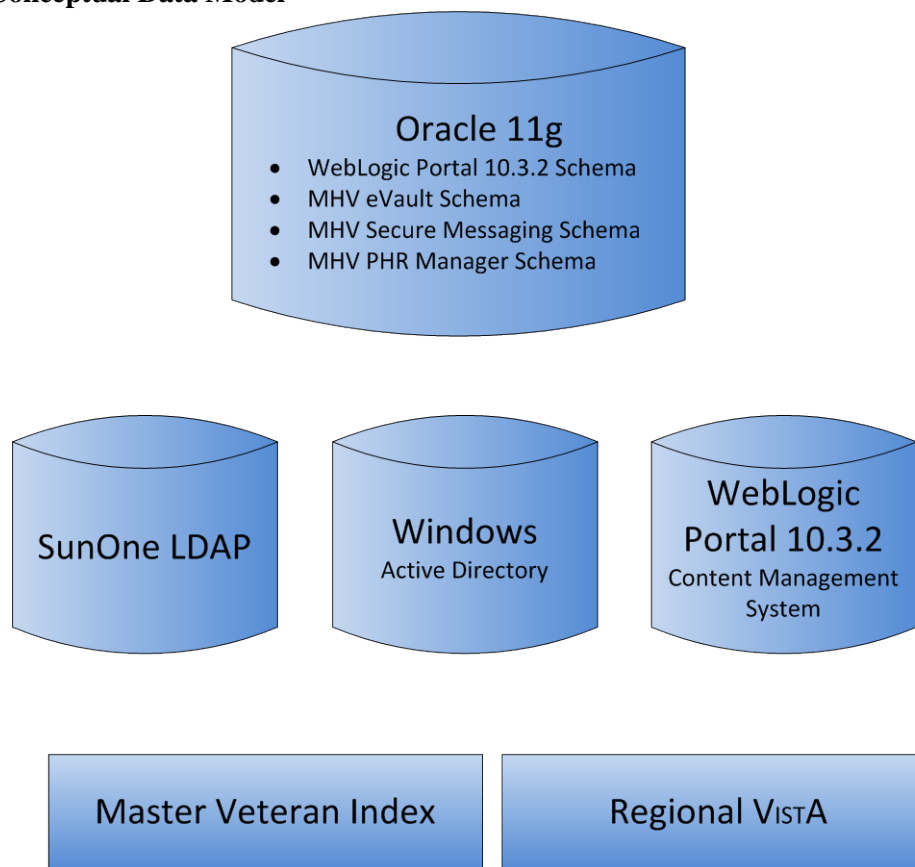
3.2.1. Project Conceptual Data Model

The data view is a logical high level view of the primary data repositories where MHV Portal information is stored.

- **Oracle11gi:** This is the primary storage area for My HealthVet Portal. There are two schemas within this repository i.e. WebLogic Portal 10.3.2 schema and eVault Schema. The WebLogic Portal 10.3.2 schema stores configuration information for the WebLogic Portal framework. The eVault schema persists user data generated by My HealthVet Portal. Detailed information on these schemas is found in the MHV Data Model.
- **SunOne LDAP:** SunOne LDAP contains external user information. Detailed information on the directory schema is found in the MHV Data Model
- **Windows Active Directory:** Windows Active Directory is used to get the user information for internal applications like Admin Portal
- **WebLogic Portal Content Management System:** All the content presented in MHV is stored in WebLogic Portal Content Management System
- **Master Patient Index:** This is an external system that contains a master index of VA patients
- **Regional VistA:** This represents the individual VistA systems located at the regional VISNs. These contain the medical information for VA patients.

The following figure represents the concept data model.

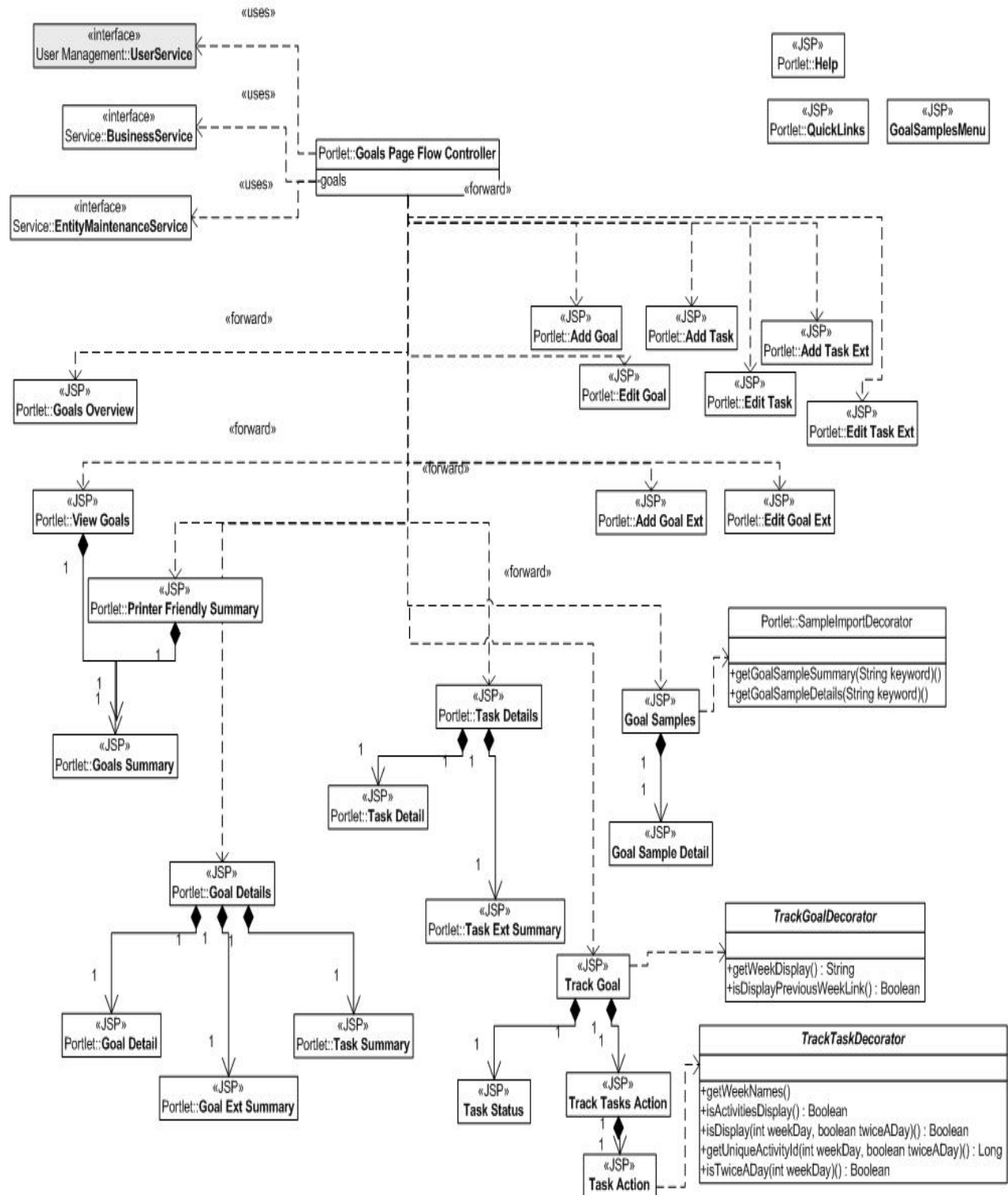
Figure 10: Conceptual Data Model



A project conceptual data model (CDM) is a high level representation of the data entities and their relationships. It does not typically include the data elements that comprise each entity.

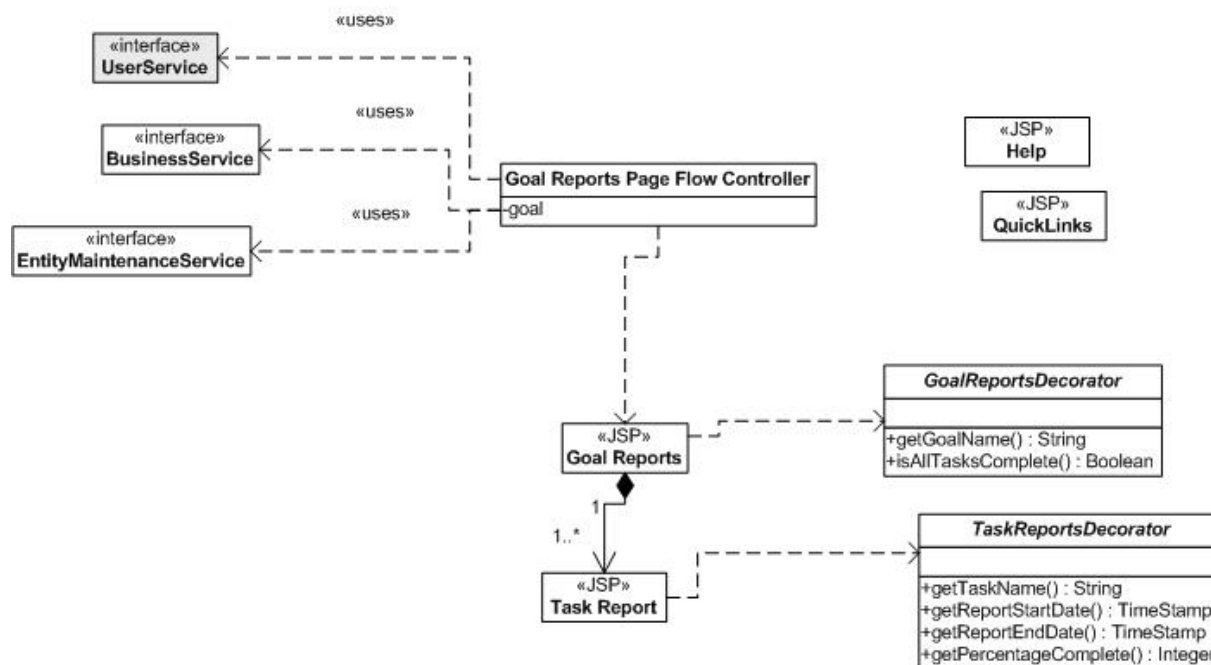
A Goal Maintenance Conceptual Data Model is show below.

Figure 11: Goal Maintenance Conceptual Data Model



A Goal Reports Conceptual Data Model is shown below.

Figure 12: Goal Reports Conceptual Data Model



3.2.2. Database Information

A description of a database inventory is in the table below.

Table 15: Database Inventory

Database Name	Description	Type	Steward
EVAULT	MHV Schema		
WEBLOGIC	Weblogic Schema for Weblogic Portal 10.3.2	Modify existing table. Creating some new tables.	
eVault	A secure electronic storage area where Veterans will maintain their Personal Health Record.	Create / Replace / Interface / Modify	My HealtheVet
PHRMGR	Schema utilized for operations and tracking progress of PHR data extraction and loading.	Create, Retrieve, Updated, Delete	My HealtheVet

3.2.3. User Interface Data Mapping

This section is N/A for MHV 12.9 as additions / changes per 12.9 are for feature additions and enhancements that do not add additional data mappings.

3.3. Conceptual Infrastructure Design

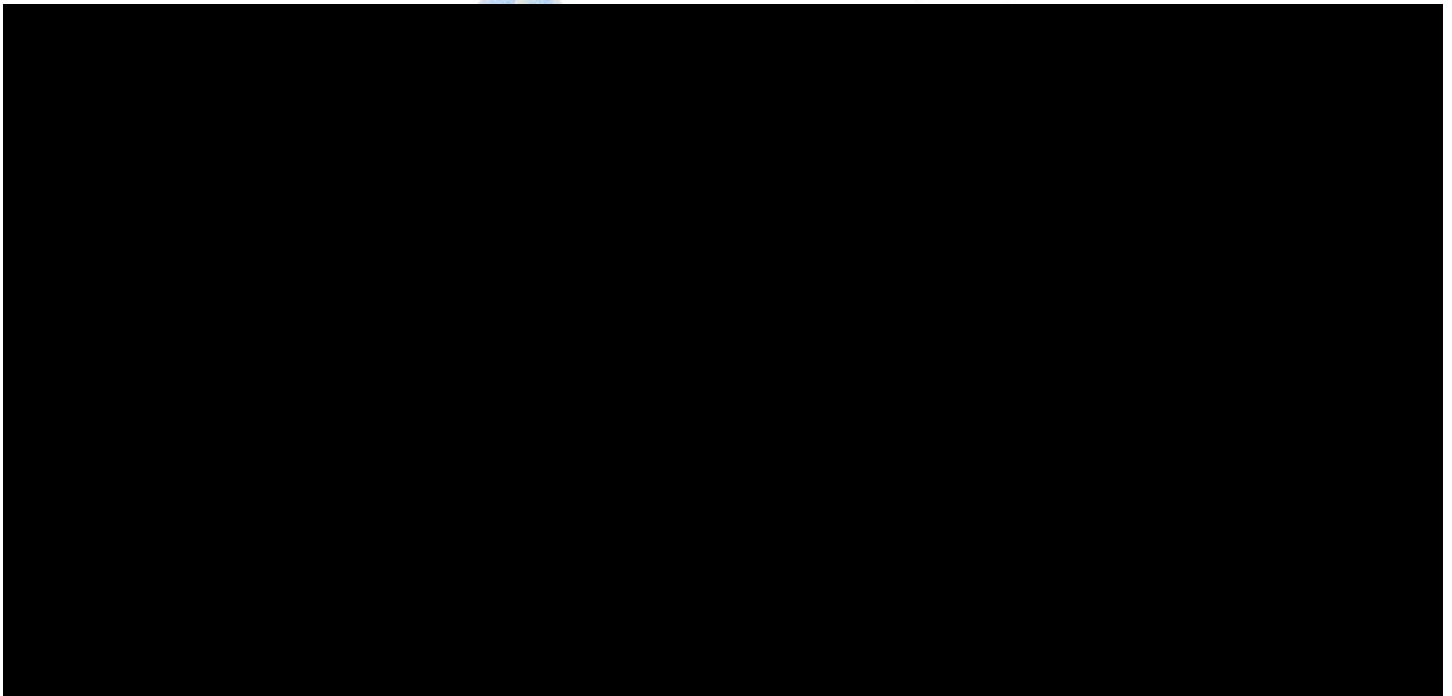
The MHV Conceptual Infrastructure Design is a very high level overview of the infrastructure that will be used to support the application. Primary emphasis is on the environments that will be required and the locations at which they will be installed. The Infrastructure Design will become far more detailed at later stages as more information regarding the system is collected and the Infrastructure requirements (i.e., capacity requirements) are better known.

Moreover, the Conceptual Infrastructure Design describes unique technology that is to be used, either to which this system will attach itself or which are part of this system. Because the system is at a very preliminary design stage, it is expected that the information provided can need to be changed during later design stages.

The deployment architecture view shows a detailed representation of the hardware and networking systems that are used by the MHV Portal in the production environment. This environment is located at the Corporate Data Center of Operations (CDCO) which provides hosting services to the MHV Portal system.

The CDCO environment is shown in the below image.

Figure 13: CDCO Environment



The My Recovery Plan (MRP) system is a component of the Improve Veteran Mental Health (IVMH) Initiative. It provides an interactive set of web-based tools, delivered via the MHV Personal Health Record, which allows Veterans who have behavioral or mental health concerns to track important aspects of their self-care and professional care.

3.3.1. System Criticality and High Availability

The approach that will be taken to meeting the System Criticality and High Availability Requirements was previously identified, including the extent to which geographically distributed, high availability designs are planned. The approach that is taken towards high availability as well as any workload distribution scheme that is planned to support the high availability implementation was also previously mentioned.

3.3.2. Special Technology

There is no special technology that is part of this system.

3.3.3. Technology Locations

Technology locations are listed in the below table.

Table 16: Technology Location

Technology Component Production	Austin Information Technology Center
Workstations	
Special Hardware	
Interface Processors	
Legacy Mainframe	
Legacy Application Server	
Legacy Databases	
Other	
Certification	
QAX	
Education	
Test	Austin Information Technology Center
Development	Austin Information Technology Center
Application Server	AITC
Interface Processors	AITC
Database	AITC

3.3.4. Conceptual Infrastructure Diagram

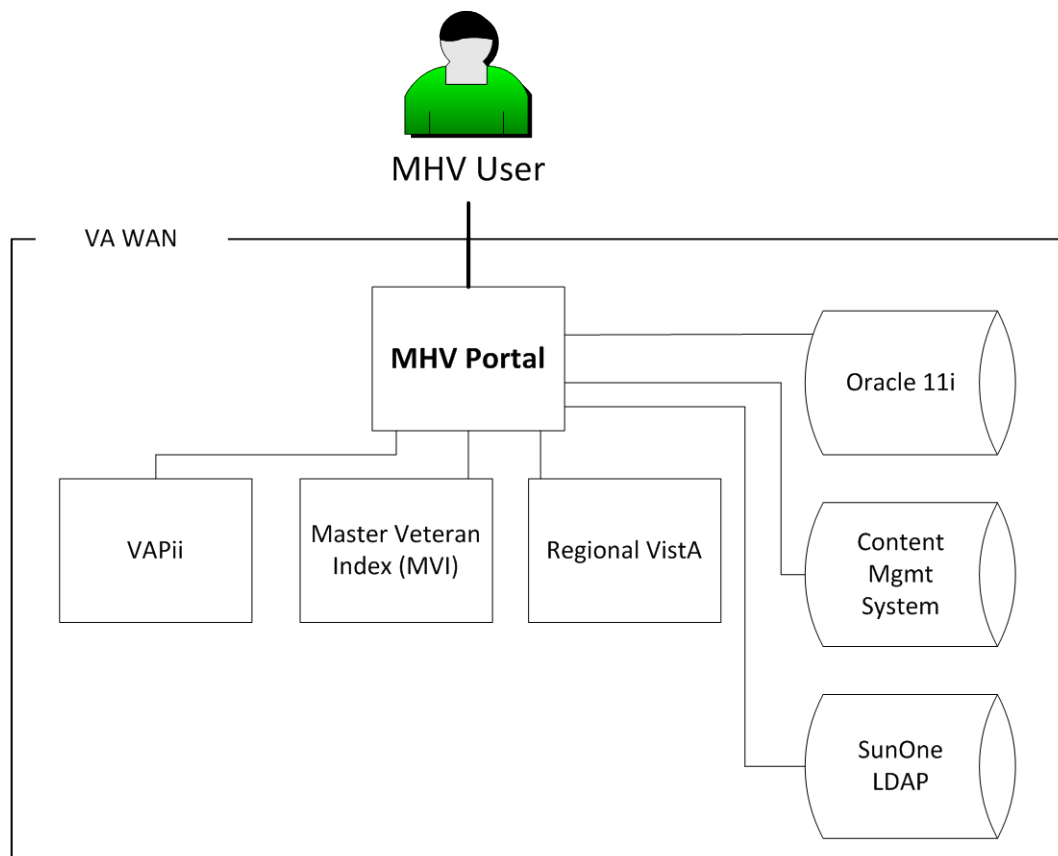
3.3.4.1. 3.3.4.1 Location of Environments and External Interfaces

Graphically, the diagram in the following figure shows the environments that will be supported, the local networks to which they will be attached (Production, Test, or Development), and the locations at which they will be installed. Also shown are the external connections, along with the external interfaces shown in terms of where they enter the network.

A layout of conceptual networks and environments is shown in the below image.

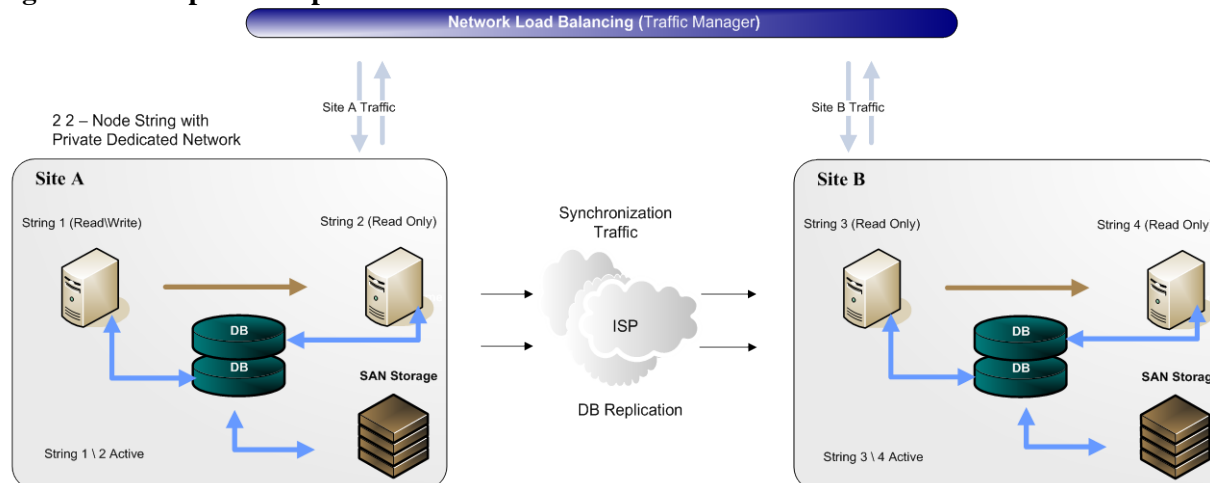
3.3.4.2. 3.3.4.2 Conceptual Production String Diagram

Figure 14: Conceptual Networks and Environments



Sample Conceptual Networks and Environments are shown in the image below.

Figure 15: Sample Conceptual Networks and Environments



4. System Architecture

The figures below represent the MHV Architecture for the Development (INT-A, INT-B, SysTest), L&P, Pre-Production and Production environments.

Figure 16: MHV Development Architecture (as of 05/21/2014)

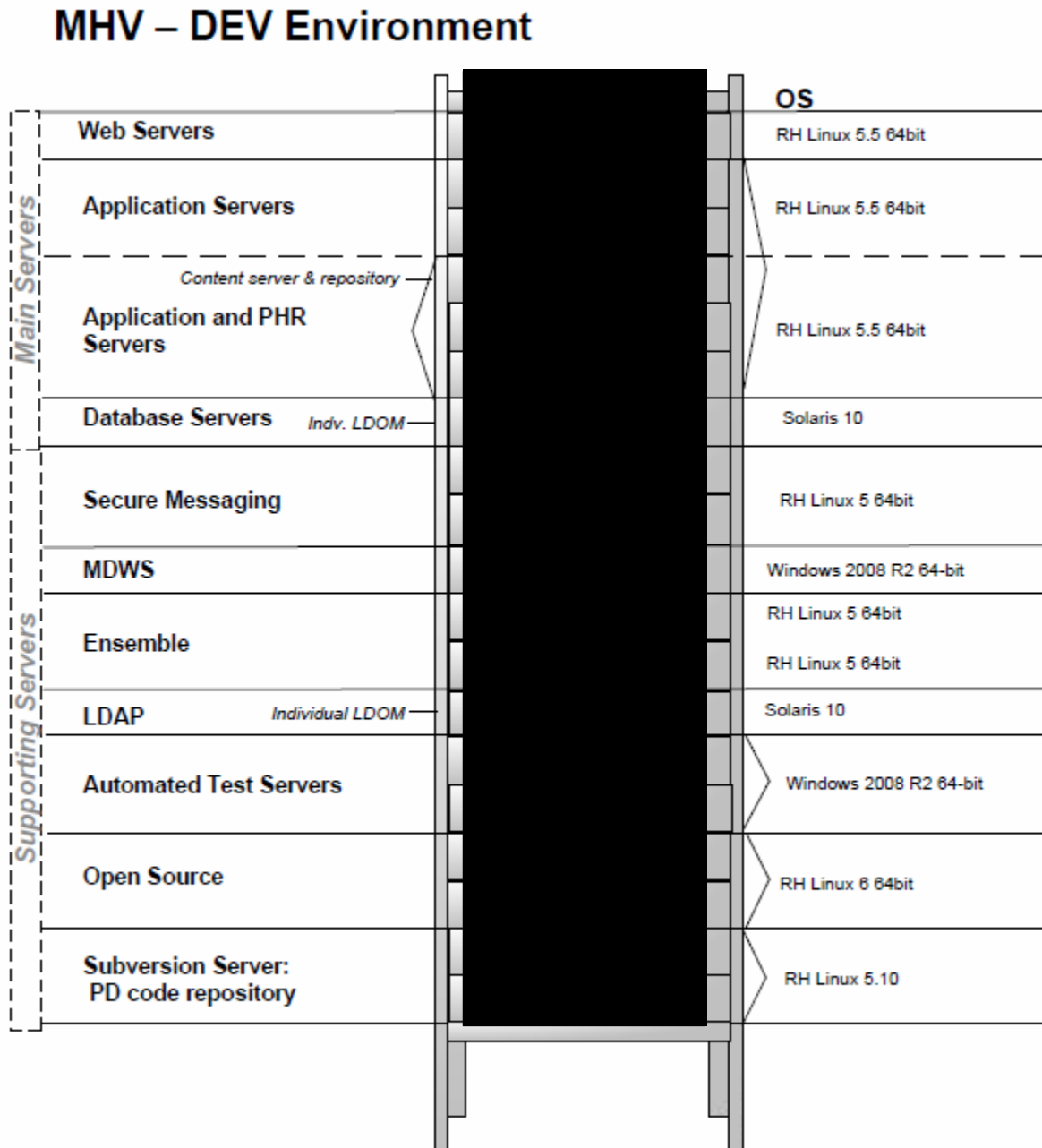


Figure 17: MHV L&P Architecture (as of 05

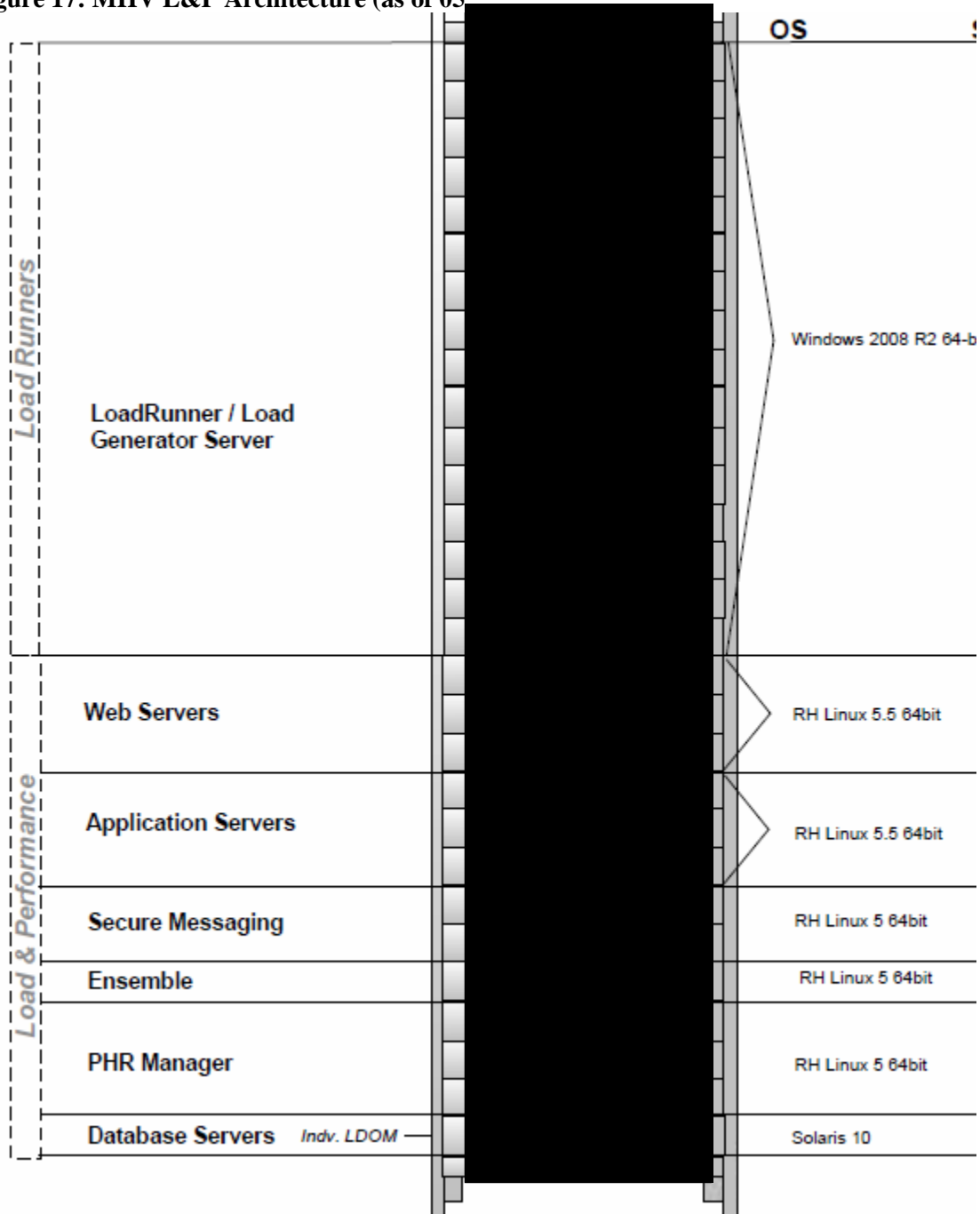


Figure 18: MHV Pre-Production Architecture (as of 05/21/2014)

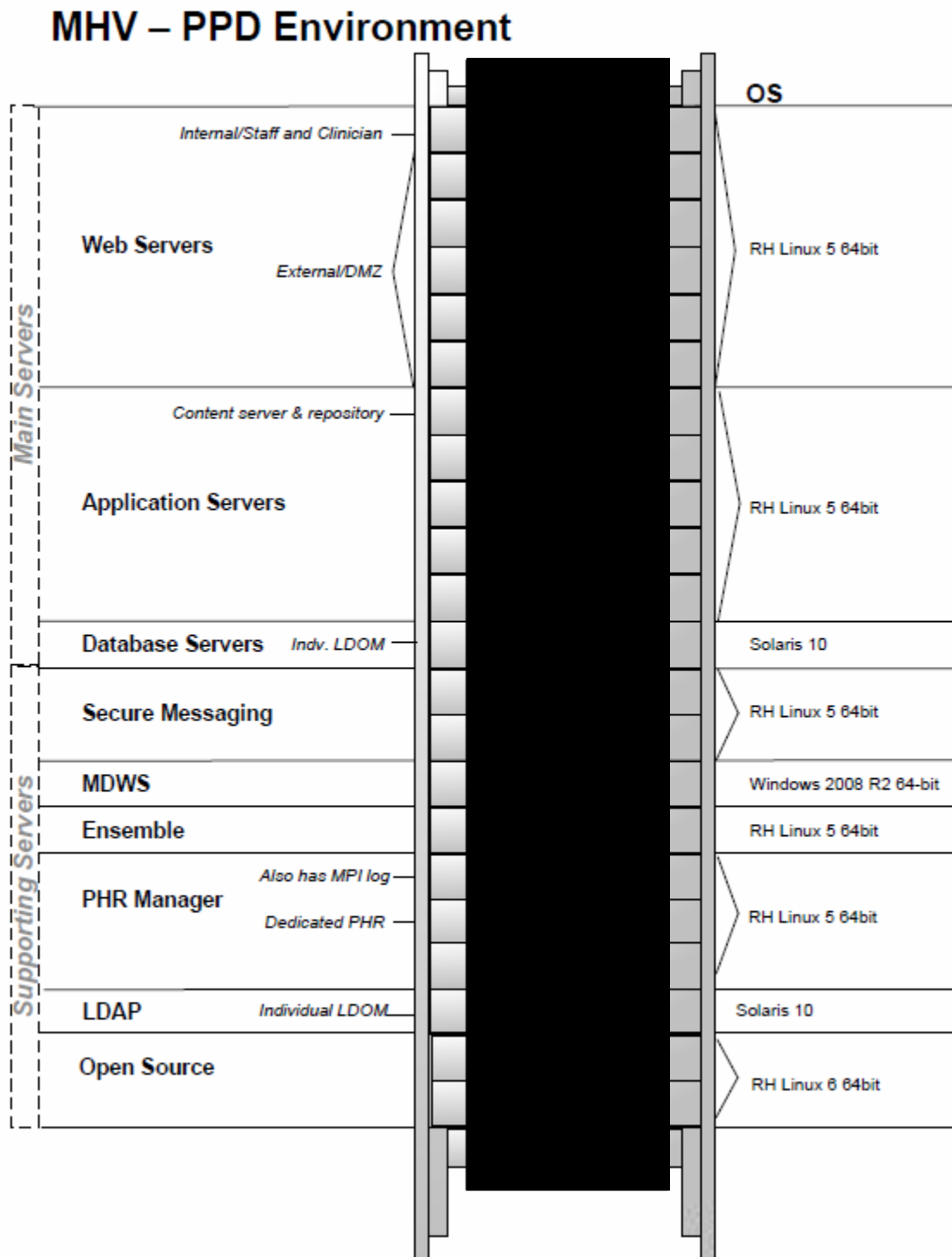
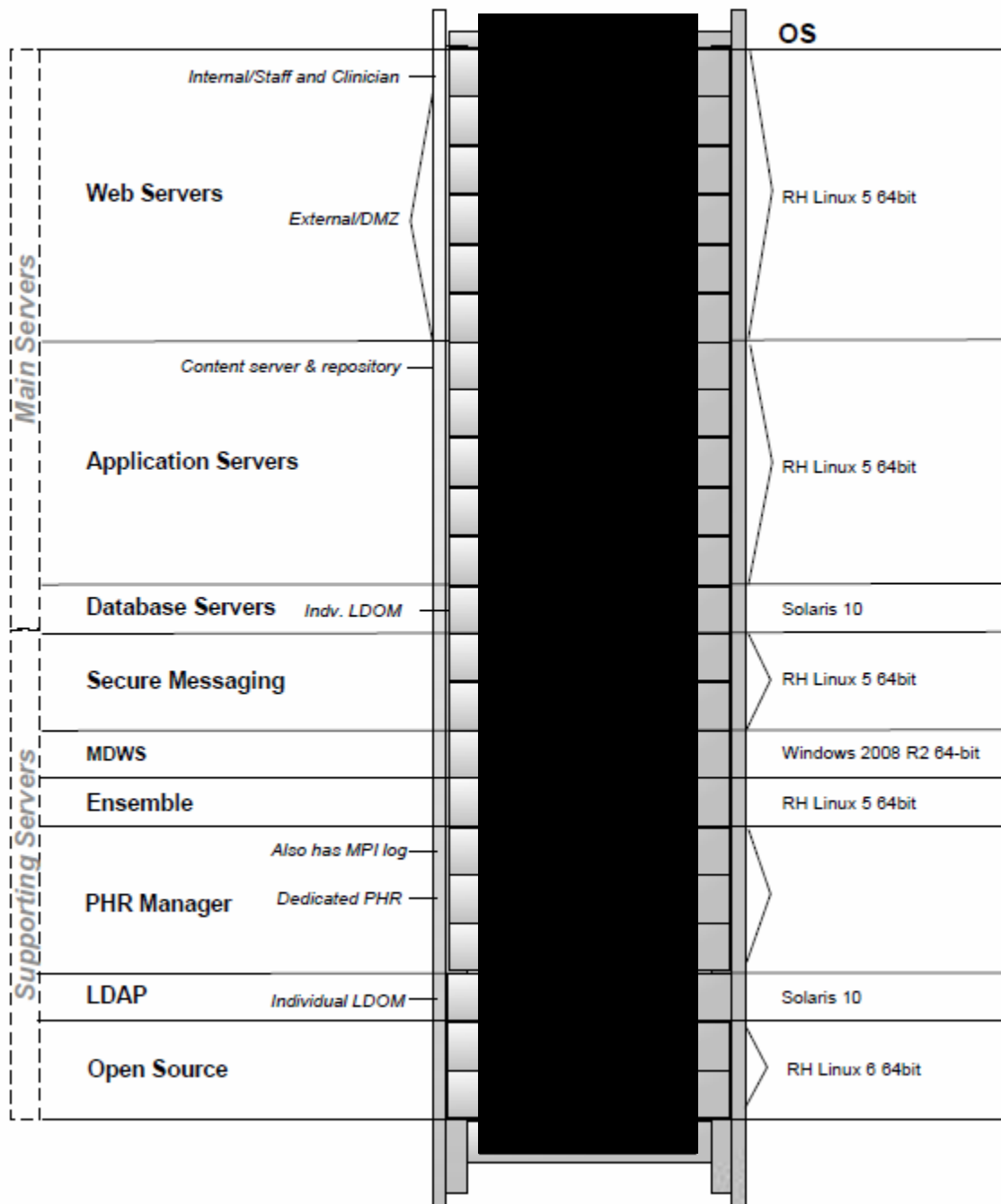


Figure 19: MHV Production Architecture (as of 05/21/2014)

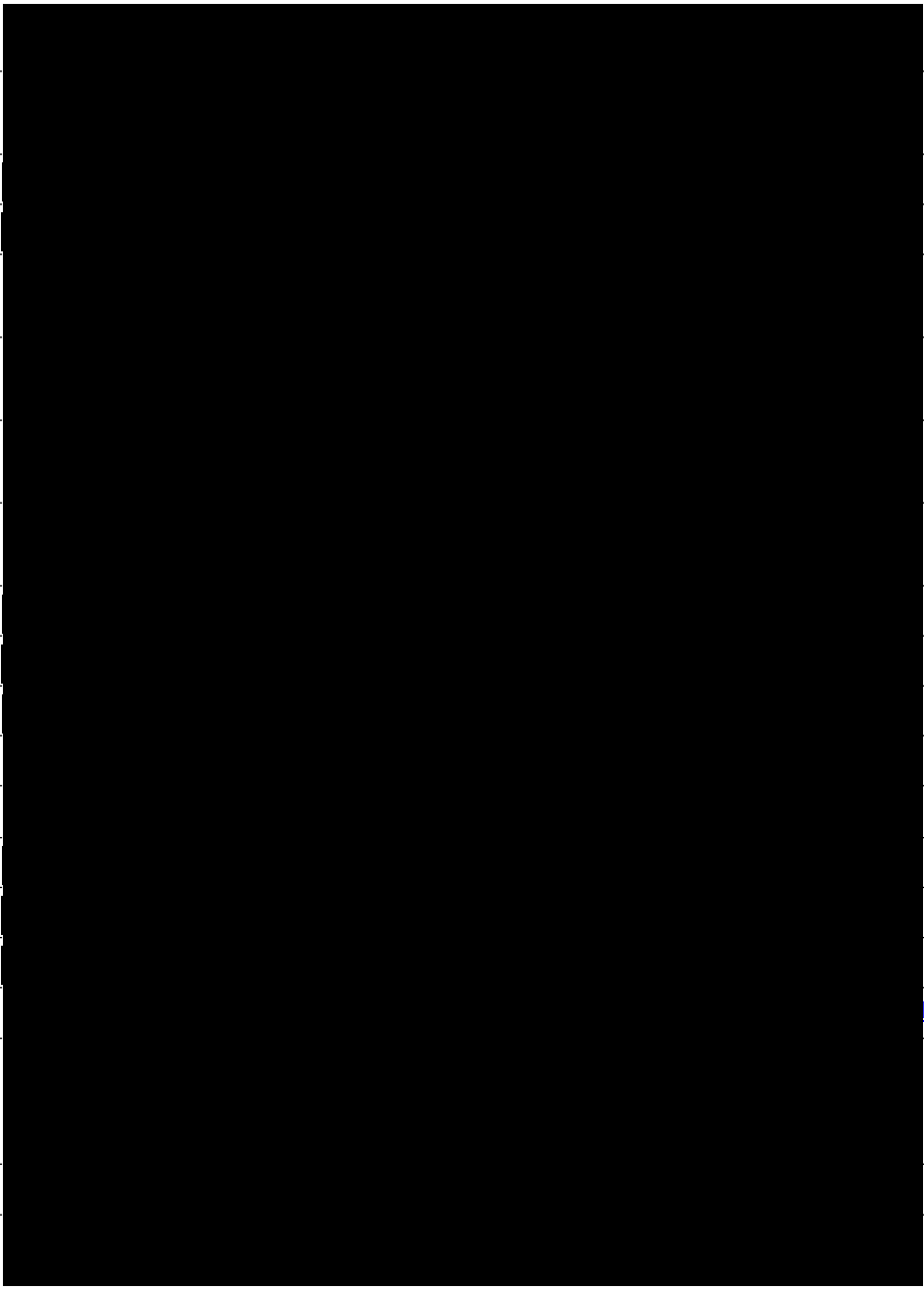


4.1. Hardware Architecture

MHV Application Server Logs Environment

Note: MPI/MVI, Database, Active Directory, MDWS, HDR, LDAP, and SMTP are currently shared across all environments. A hardware overview for all environments are displayed in the table below.

Table 17: SysTest Hardware Overview

SysTest	Description
Links—MHV Portal	
Links—ADMIN Portal	
Web Server(s)	
App Server(s)	
Admin Console—MHV	
Admin Console—SM	
Managed Nodes—MHV	
Managed Nodes—SM	
Database	
LDAP	
Active Directory	
VistA(s)	
Ensemble	
MDWS	
PHR Manager	
Email Cron	
MVI IAM Service	
HRA	
Direct	
VAPII	

SysTest	Description
PHR Redesign	

Table 18: Preview Hardware Overview

Preview	
Links—MHV Portal	
Links—ADMIN Portal	
Web Server(s)	
App Server(s)	
Admin Console—MHV	
Admin Console—SM	
Managed Nodes—MHV	
Managed Nodes—SM	
Database	
LDAP	
Active Directory	
VistA(s)	
Ensemble	
MDWS	
PHR Manager	
Email Cron	
MVI IAM Service	
HRA	
Direct	
VAPII	
PHR Redesign	

Table 19: Integration B Hardware Overview

Integration B	Description
Links—MHV Portal	
Links—ADMIN Portal	
Web Server(s)	
App Server(s)	

Integration B	Description
Admin Console—MHV	
Admin Console—SM	
Managed Nodes—MHV	
Managed Nodes—SM	
Database	
LDAP	
Active Directory	
VistA(s)	
Ensemble2	
MDWS	
PHR Manager	
Email Cron	
MVI IAM Service	
HRA	
Direct	
VAPII	
PHR Redesign	

Table 20: Integration A Hardware Overview.

Integration B	Description
Links—MHV Portal	
Links—ADMIN Portal	
Web Server(s)	
App Server(s)	
Admin Console—MHV	
Admin Console—SM	
Managed Nodes—MHV	
Managed Nodes—SM	
Database	
LDAP	
Active Directory	
VistA(s)	
Ensemble2	
MDWS	
PHR Manager	
Email Cron	
MVI IAM Service	

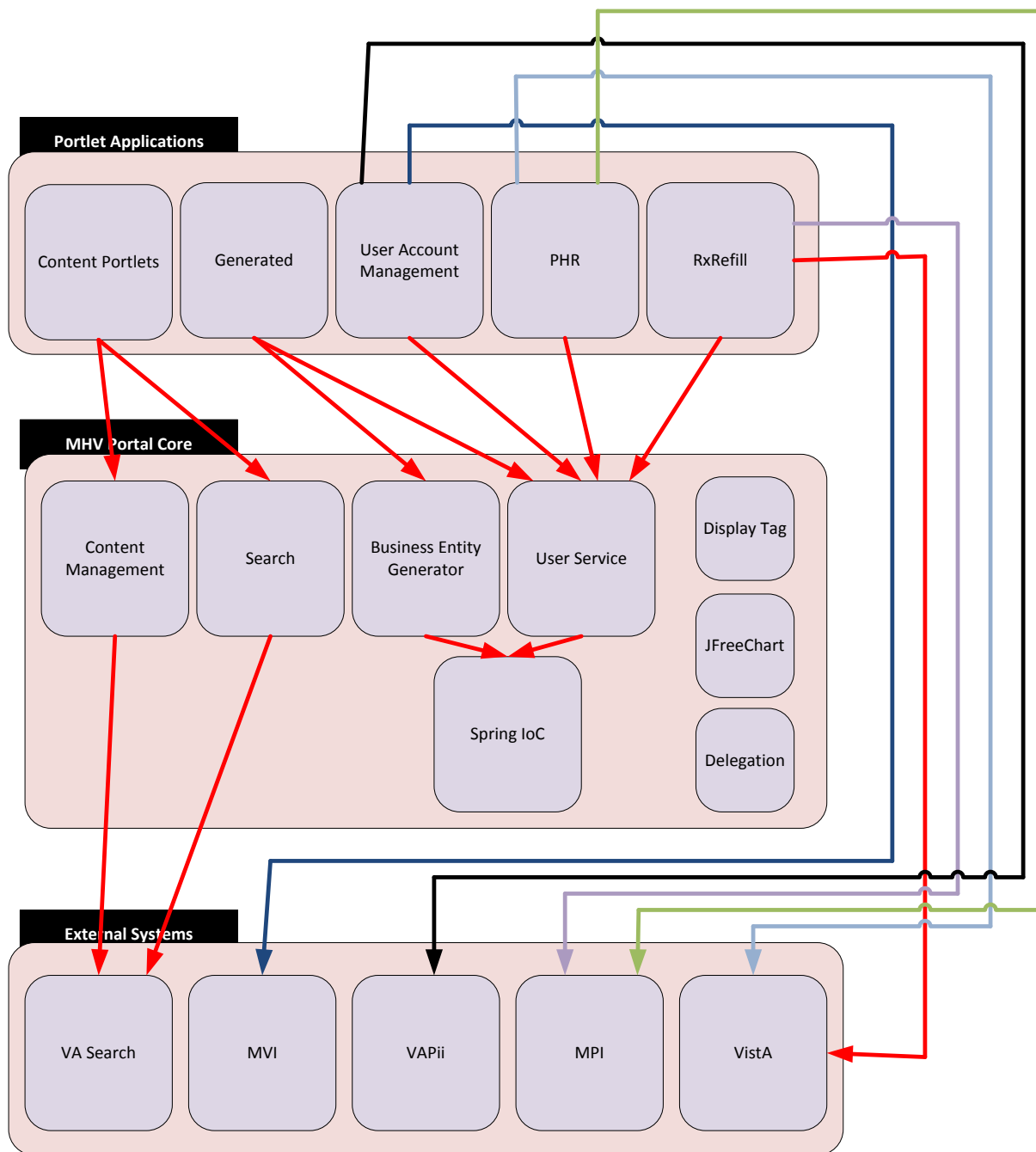
4.2. Software Architecture

The software architecture view shows the major software components of the MHV Portal system and their dependencies. As seen in the diagram below these components are logically divided into three main areas:

- **Portlet Applications:** This area contains the components that provide the content and services of the MHV Portal to the user. Each component in this area represents the entirety of individual portlet application or logical groupings of those applications. Detailed design information for each of the portlet application can be found in their respective design documents.
- **MHV Portal Core:** This area contains the services and utilities components that provide infrastructure services to the portlet applications. Detailed design information for the services and utilities can be found in their respective design documents.
- **External Systems:** This area represents components that are provided by other parties and are used by the MHV Portal system. The MHV Portal uses the services these component provide through interfaces and protocols defined by their owners.

Software Architecture layout is shown in the image below.

Figure 20: Software Architecture

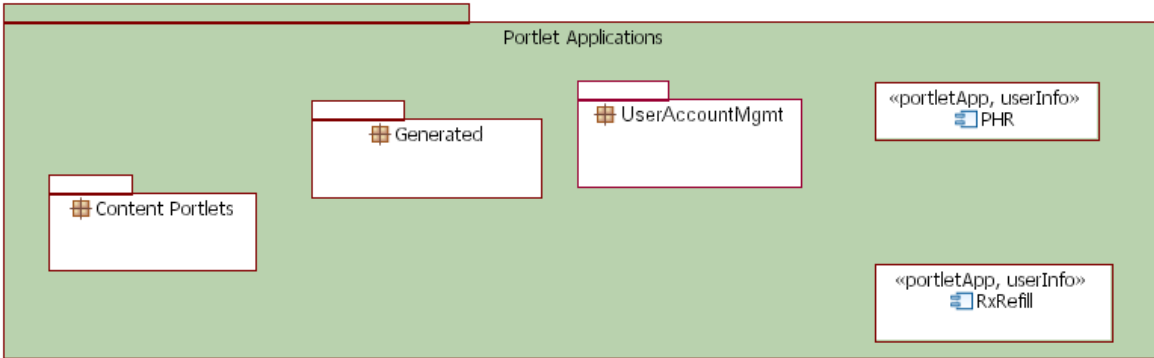


The following section describes each component or component grouping within each layer.

4.2.1. Portlet Applications

The following figure displays the portlet applications and their structure.

Figure 21: Portlet Applications

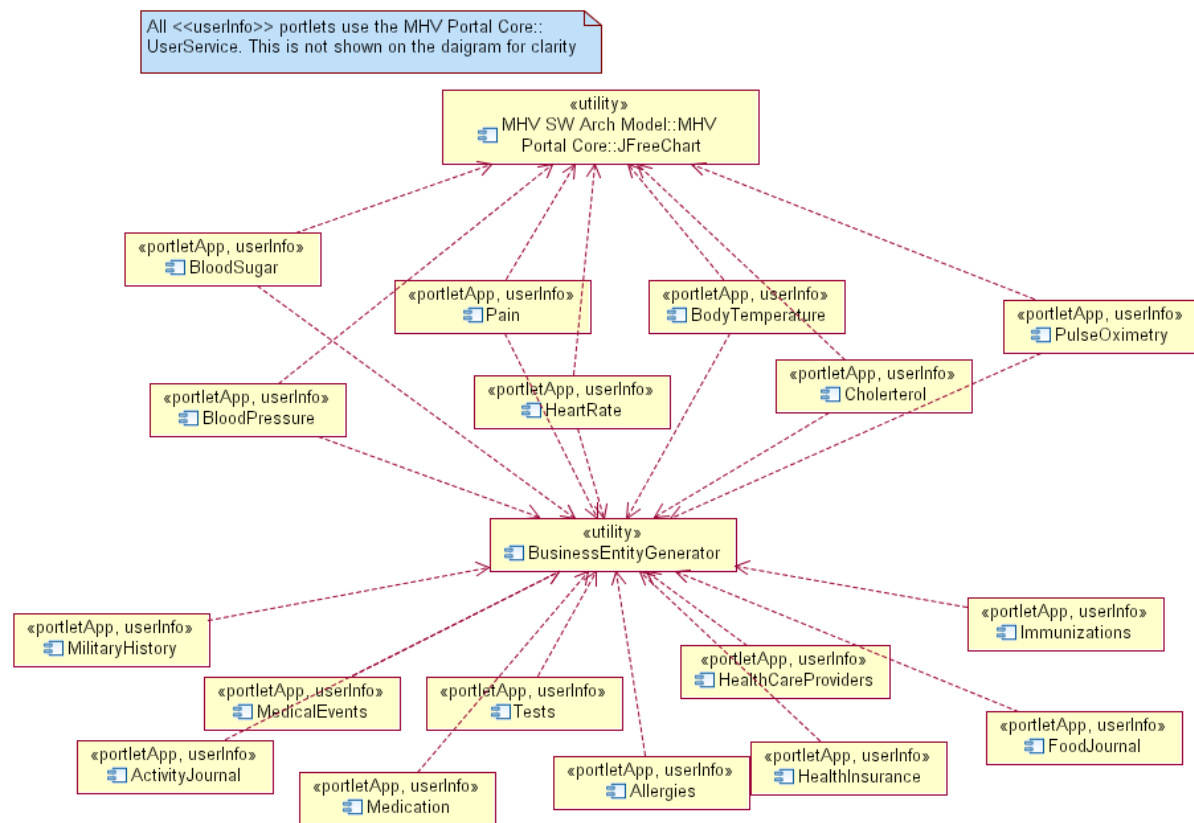


4.2.2. Generated

The generated portletApp components use the BusinessEntityGenerator utility to generate their service and domain layers. In addition certain portletApp components use JFreeChart utility to create graphs of data.

Generated portletApp Components are shown in the image below.

Figure 22: Generated portletApp Components

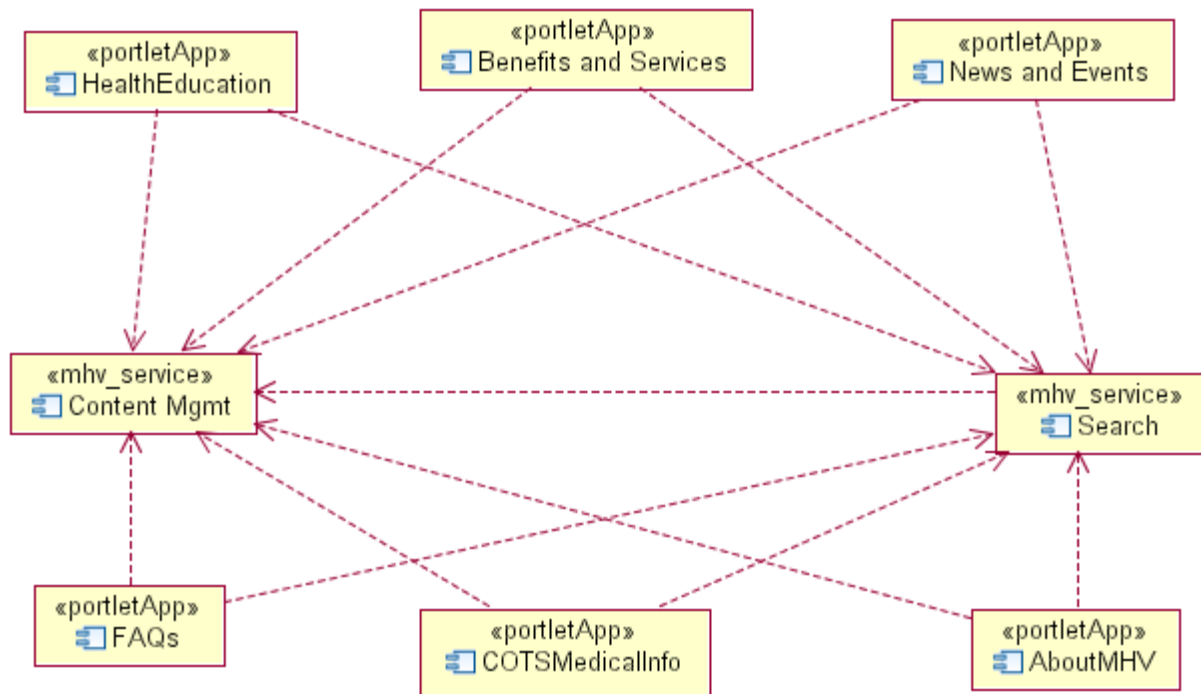


4.2.3. Content Portlets

The content portletApps represent components that are responsible for presenting content to the user. Each of the components represents a category of managed content. Each content portletApp interfaces with the Content Mgmt service which is responsible for hosting and managing content, and the Search service component which provides search capabilities across all of the content.

Content portletApps Components are displayed in the image below.

Figure 23: Content portletApps Components

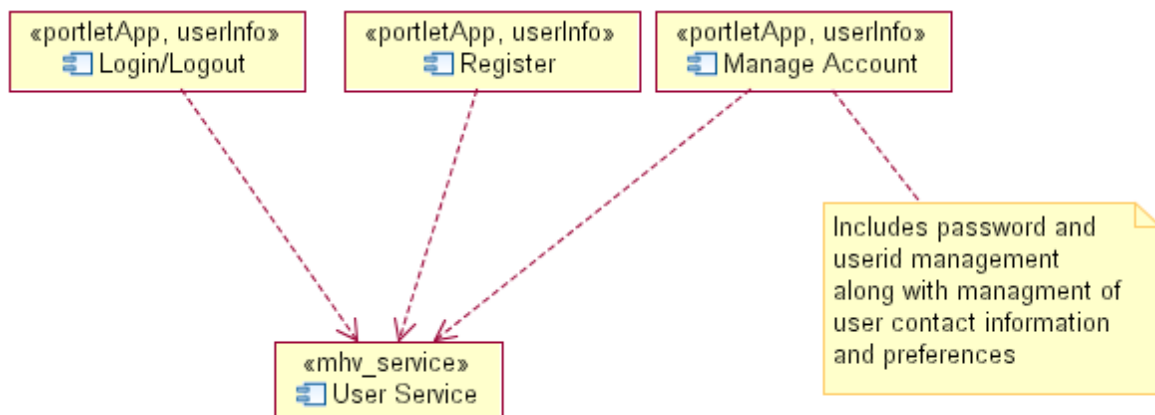


4.2.4. User Account Management

All portletApp components involved with the user and their account are grouped into the User Account Mgmt package. The components in this package depend on the User Service component to manage the user account information.

User account management package is displayed in the image below.

Figure 24: User Account Mgmt Package

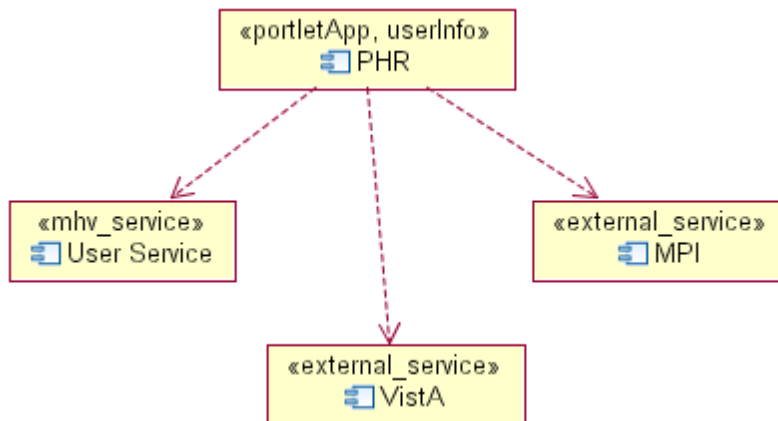


PHR portletApp Components are shown in the image below.

4.2.5. PHR

The PHR portletApp component is responsible for the retrieving, storage, and presentation of a user's personal health record. It depends not only on the User Service component but also the VistA and MPI external services.

Figure 25: PHR portletApp Components

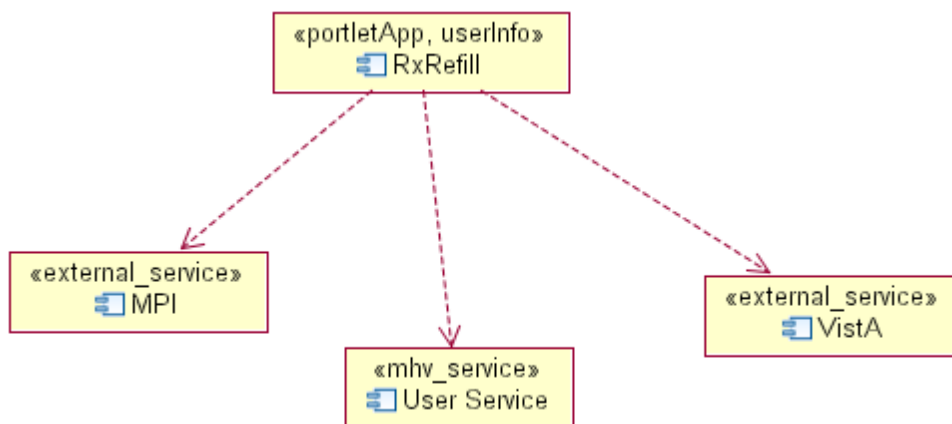


4.2.6. RxRefill

The RxRefill portletApp component is responsible for the retrieving and presenting users prescription information and also allowing the user to refill their prescriptions using the MHV Portal. The RxRefill depends not only on the User Service component but also the VistA and MPI/MVI external services.

RxRefill portletApp Components are shown in the image below.

Figure 26: RxRefill portletApp Components

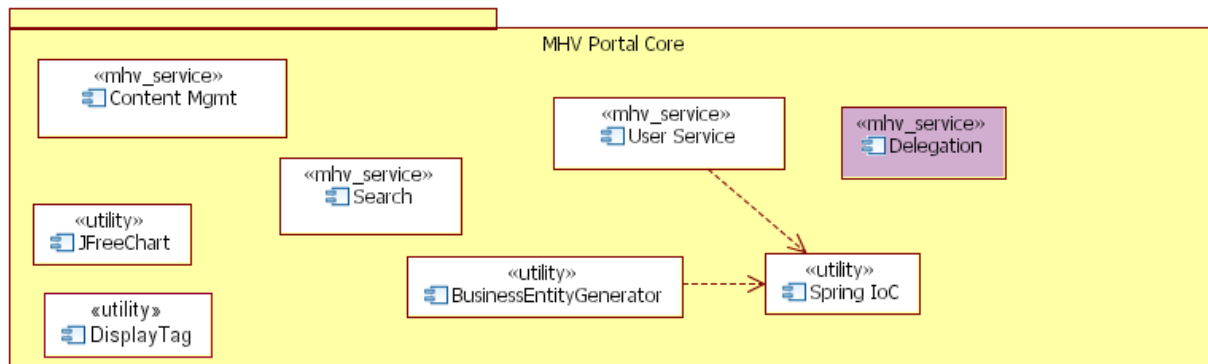


4.2.7. MHV Portal Core

The Portal Core components represent the MHV Portal specific infrastructure services and utilities that allow portletApps to be integrated into the system. Note that these are MHV Portal specific and don't include services and utilities that are provided by the BEA WebLogic Portal 10.3.2 framework.

Portal Core Components are displayed in the image below.

Figure 27: Portal Core Components



4.3. Network Architecture

4.3.1. MHV System Network Communications

Communication within MHV systems will occur over TCP/IP within the internal MHV local area network. All requests for protected resources from an end user's system will be transported via HTTPS to the externally facing MHV website or APIs. After passing a firewall, requests will be handled by an internal server for load balancing and then passed to an available application server for processing. Requests will then be serviced and returned by the application server. As necessary, the application server will make requests to internal services via a web service call, RPC, or other mechanism to fulfill the specific needs of the end user's request.

Internal service requests for resources will be passed over TCP/IP to the appropriate service. Methods in which services may be invoked include SOAP Web Services, RESTful Web Services, Java Messaging Services (JMS), and Remote Method Invocation (RMI). The method by which services are invoked depends on existing codebases and will be thoughtfully considered in order to ease development requirements for the transition to new architectures. Routing, load balancing, and lookup for RESTful resources may be handled by DNS. Data tier services will handle requests and connect via JDBC to data stores as needed. The vast majority of Veteran-facing services will follow this model. Additionally Direct protocol users would require communication via SMTP for sending secured email attachments.

4.3.2. Logical Communications Segmentation

An important piece of the communications architecture is the use of logical separations between network sections. Within MHV, this logical division is provided by the use of firewalls. Two firewalls are used within the system to separate disparate sections. These separations occur between external users and the DMZ (demilitarized zone on which load balancing servers and static content are placed) and between the DMZ and the internal MHV network.

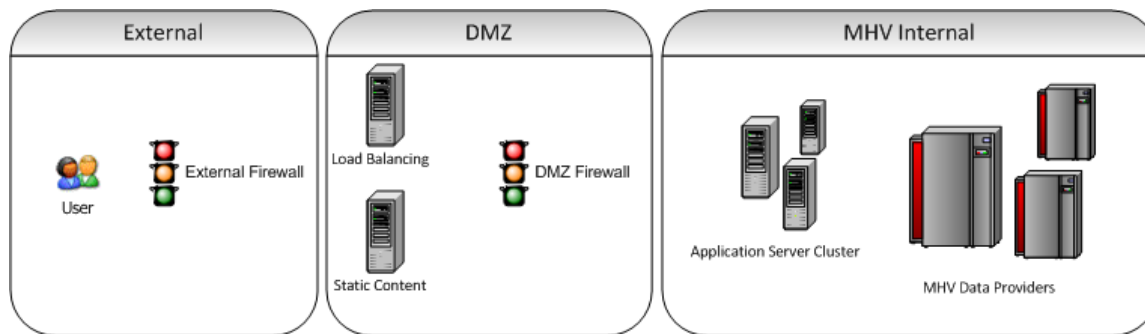


Figure 28: Logical Communications Segmentation

The above figure displays the Logical Communications Segmentation.

4.3.3. Backend Data Synchronization

Backend data synchronization and/or backup processes for many internal systems also occur on a routine basis. These systems include:

- VistA
- Active Directory
- MDWS
- Cache/Ensemble
- MVI

These data synchronization events occur over a number of protocols. HL7/MLLP is used for data synchronization between VistA and Cache/Ensemble as well as communication with MVI. LDAP is used for user synchronization and authentication when dealing with non-VA credentialing. The rsync network protocol is used for the synchronization of files from backend application server-based CMS systems to front-end servers located in the DMZ so that serving of static content may be expedited.

4.4. Service Oriented Architecture / ESS

The overarching architecture of the MHV software design is being transitioned to a service-oriented architecture (SOA). The use of SOA is considered a best practice for maintaining logical divisions in a complex system. Modules contain their own discrete and encapsulated purpose, and interact with another on a contract basis comparable to the interface/implementation Java construct. This allows modules to be easily integrated even while undergoing internal development as long as the service contract is not altered (loose coupling). Modules can be maintained by separate teams with their own development cycle and be deployed independently without dependencies on other systems.

For an SOA system to function properly, systems must know of each other's existence. For this purpose, a service registry is to be used. This will not only facilitate the communication of independent services with one another, but duplicate services can be eliminated or adapted to provide common functionality for use across the entire enterprise. For example, a service for saving files associated with an MHV user could be used by multiple subsystems. A secure messaging application may have the need to save file attachments for a user who wishes to share files with providers, whereas the core MHV application may need to allow file uploads such as an image of the user with whom the account is associated with. Using the same service to accomplish the same basic task of accepting and saving a binary file and associating it with a

user account can eliminate duplicate code that would otherwise be implemented in their own respective modules.

4.5. Enterprise Architecture

The objective of this system design is to provide guidance, architectural vision and design specifications to the MHV Product Development technical team for aligning MHV systems with the [OneVA Enterprise Architecture](#) (EA). The MHV application has a decade long history and over the course of those 10 years, the application's scope has grown to support many business and technical needs that were not originally imagined and planned for in the design of the application. Additionally, technologies and strategies for delivering services to customers have changed dramatically throughout the application's lifetime. The combination of these points necessitates a re-evaluation of the current architecture in order to maintain program viability for the future.

In order for MHV to align with OneVA Enterprise Architecture and meet new business objectives, steps must be taken to map the existing MHV systems and components (defined above) into a layered Service-Oriented Architecture (SOA) as is defined by the Enterprise. A short discussion of the Enterprise Architecture and more specifically, the architectural layers that the MHV components will be mapped in to is discussed below.

The One VA Enterprise Architecture defines a layered architectural approach in which system components are broken down into logical layers or subcomponents that fulfill a specific need of the applications (e.g. user interface, business need, persistence, etc). Additionally, guidance is provided for how logical subcomponents should interact with each other in order to maintain a good system design. These SOA layers as defined by OneVA SOA Technical Framework are described below.

- **Interface Layer** – The Interface Layer is the layer where applications services that interface with humans, devices or other systems are defined and exist. The Interface Layer provides a single point through which all external requests must flow, and where security can be applied to incoming external requests, and where input data can be verified and validated.
- **Business Logic Layer** – The Business Logic layer includes the core computational services that perform the bulk of the application specific capabilities or work of the system. The Business Logic layer provides implementation of core business processes.
- **Data Layer** – The Data Layer provides services for restricting which services can create, retrieve, update, and delete (CRUD) data stored and owned by the system. There are two categories of data services. The first are business data services, which abstract underlying data structures from the business services that utilize them, and the second are services explicitly dependent upon the underlying physical data store.
- **SOA Integration Layer** – The SOA Integration Layer includes all infrastructure required to support orchestration, choreography, transport, and format transformations required to facilitate inter-service integration. This layer might also include gateway services such as authentication, authorization, and auditing / logging capabilities.

Note: While it is true that some MHV components have been designed such that there is a distinction between Presentation, Business Logic, and Data layers, there has been no strategic emphasis in the past

on developing MHV components as services as defined by the enterprise architecture (Interface, Business, and Data).

With initiatives at the VA for increasing accessibility and reusability of services and data across all programs (e.g. mobile applications), it is increasingly important for applications to provide mechanism whereby external systems can securely interact with existing application capabilities in a controlled and consistent fashion. In order for MHV systems and applications to achieve this, it is necessary to refactor existing MHV presentation components to provide a more granular deployment and configuration.

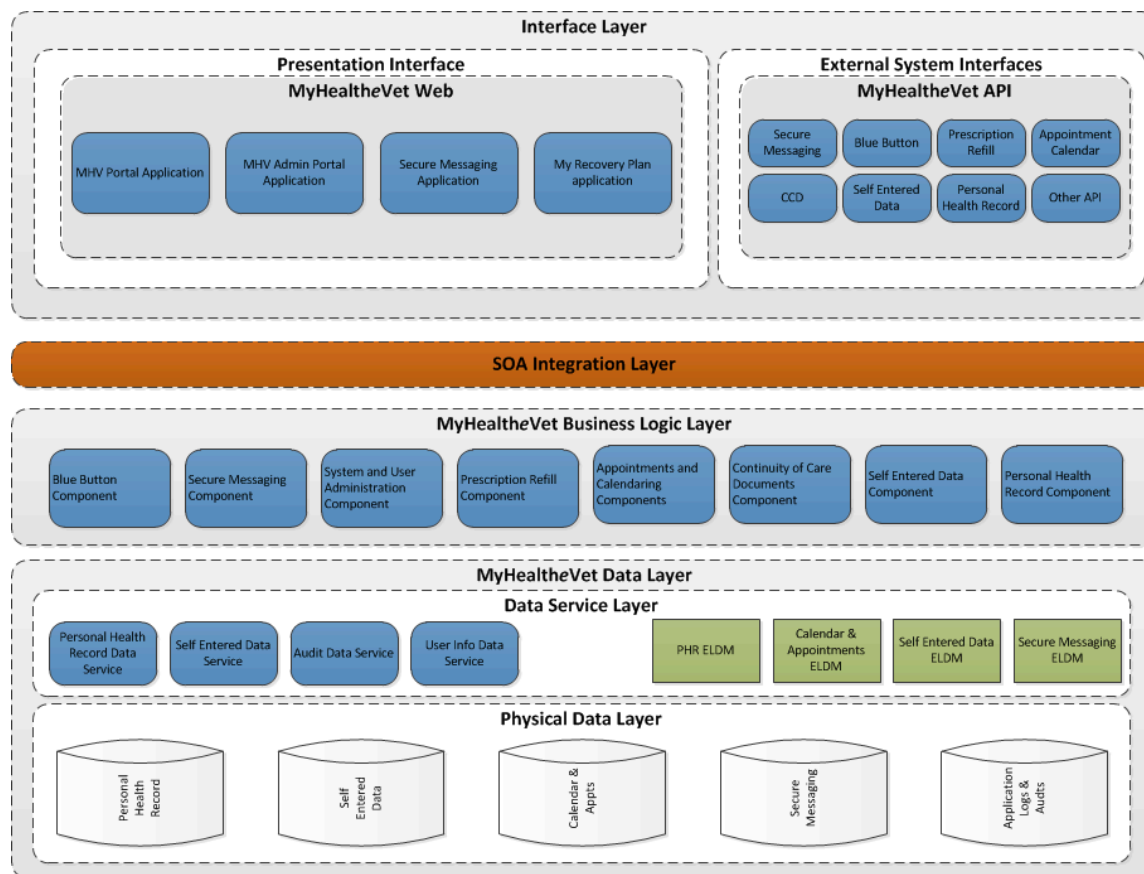


Figure 29: MHV Target System Design

The future state architecture of MHV applications, components and services is illustrated in the figure above. In this figure, the web applications (or user interfaces) are split apart into their respective architectural layers (i.e. interfaces, business objects, and data services).

The characteristics and objectives of the layered MHV architecture are described below.

- MHV application logic and business logic are fully decoupled from the data that it manages and process. This prevents scenarios where data stored locally at the application level presents barriers to information sharing across the enterprise [ETAC 2.2.2 – Data Independence].
- MHV application data is stored on enterprise servers without being saved on end-user devices or user workstations. Using enterprise resources to store permanent data lessens the burden on an application to be a proper data custodian (including security and privacy

concerns). It also promotes consistency in how data custodianship is executed and isolates changes to common services when policies are modified [ETAC 2.2.4 – Data Persistence].

- MHV layered application design supports enhanced and automated testing at the component and system levels. Components will be able to be tested more easily independent of dependencies (unit tested) and system external interface will support testability at the system level [ETAC 2.2.5 – Test Driven Development].
- MHV system architecture supports scaling horizontally across multiple commodity platforms. The layered architecture with well-defined interfaces supports packaging of application components in many different configurations. Allowing for many options for deployment, which provides the granularity needed to scale various aspects of the application [ETAC 2.2.7 – Scalability].
- MHV Redesign Architecture shall be refactored and redesigned to ensure that logic contained within the business layer remains stateless, which will allow for the same business layer objects to support both user interaction in the presentation sub-layer and system interaction via the MHV APIs [ETAC 2.2.8 – Stateless Business Logic].
- The MHV architecture shall employ and adhere to published data standards whenever possible and will be syntactically and semantically harmonized with VA Enterprise Conceptual Data Model (CDM) in order to support increased interoperability with systems within the VA [ETAC 2.3.1 – Data Standards, ETAC 2.3.3 – Enterprise Data Model].
- The MHV architecture will be designed such that the need for local caches or copies of authoritative data sources is minimized. Any caching of data for which MHV is not authoritative shall be done temporarily only and shall be purged at regular intervals to ensure that data for which MHV is not authoritative is not being stored in duplicate in MHV data stores. Permission for caching of data shall be granted by establishment of a Memorandum of Understanding (MOU) with the data steward. Creation or update of cached non-authoritative data by MHV application shall be considered effective only when that data has been successfully updated in the authoritative system [ETAC 2.3.4 – Local Copies of Authoritative Information Sources].
- MHV is better suited for reuse of components and integration through the exposure of external system interfaces or API layer
- MHV supports integration with services through a SOA integration layer (e.g. enterprise service bus infrastructure)
- MHV supports enterprise logical data model (ELDM), which facilitates standard messaging between it and other applications integrated with the SOA integration infrastructure. This further eases integration with external applications.

4.6. Services

The institution of a service-oriented architecture will also facilitate the exposure of an MHV application programming interface (API). Exposing an API for external applications to access

data on behalf of veterans is an important feature that any data holder such as the VA can expose. As is, Veterans have access to view data on the MHV website itself, and are able to download copies of their data in a few human readable formats via Blue Button. None of these features serve as a system-to-system contact point so that external applications are able to consumer veteran data, however. Implementing an external API will expand the care ecosystem, and allow developers outside of the VA to create connections and services that further benefit the Veteran outside of the abilities and budgets of the VA enterprise. Refactoring existing components of the core MHV application into services is the first step in building this extensible architecture. For example, the ability to retrieve self-entered data should be refactored into a service. This service would then be consumed by the MHV application to front-end the data into a presentable format for the Veteran. This service would also be consumed by an external-facing system acting as the API endpoint whose job it would be to accept requests from API consumers, translate that request into an appropriate service request, and format (if necessary) and transport service-returned data into a standards-based format to be consumed by the requesting application. Essentially, an API endpoint could be built to act as an interface between external data consumers and the internal architecture of a service-oriented architecture.

The following list describes the services for MHV 12.9.

- **Content Mgmt:** Provides content Mgmt services such as persistence and tagging
- **Search:** Provides search capabilities on information stored in the Content Mgmt service
- **User Service:** Provides access to user information
- **Delegation:** Provides delegation of user information to other selected by the user. This service has not yet been developed.
- **PHRMGR Service:** Provides service consumers the ability to perform refreshes of PHR data on demand and the capability to determine the progress or status of these refresh operations.

4.7. Utilities

The following list describes the utilities for MHV 12.9:

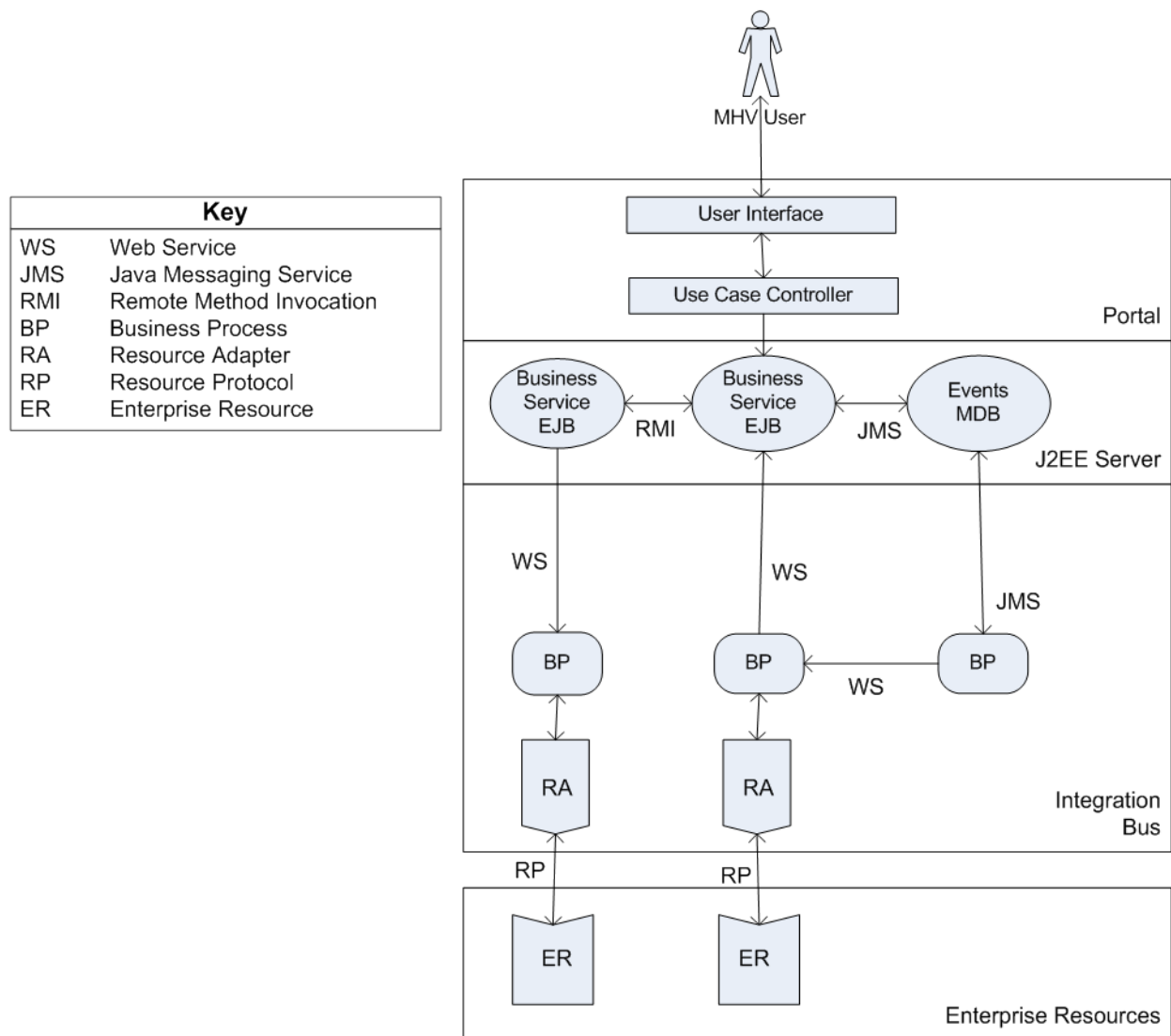
- **JFreeChart:** Open source utility for creating graphs and charts
- **DisplayTag:** Open source utility for displaying information in tables on portlet pages. Includes paging, sorting and exporting capabilities.
- **Spring IoC:** Inversion of Control (aka Dependency Injection) container
- **BusinessEntityGenerator:** Uses meta-data defined in XML files to produce business layer and domain layer elements for portletApps components

4.8. Communications Architecture

MHV uses TCP/IP communication between all subsystems.

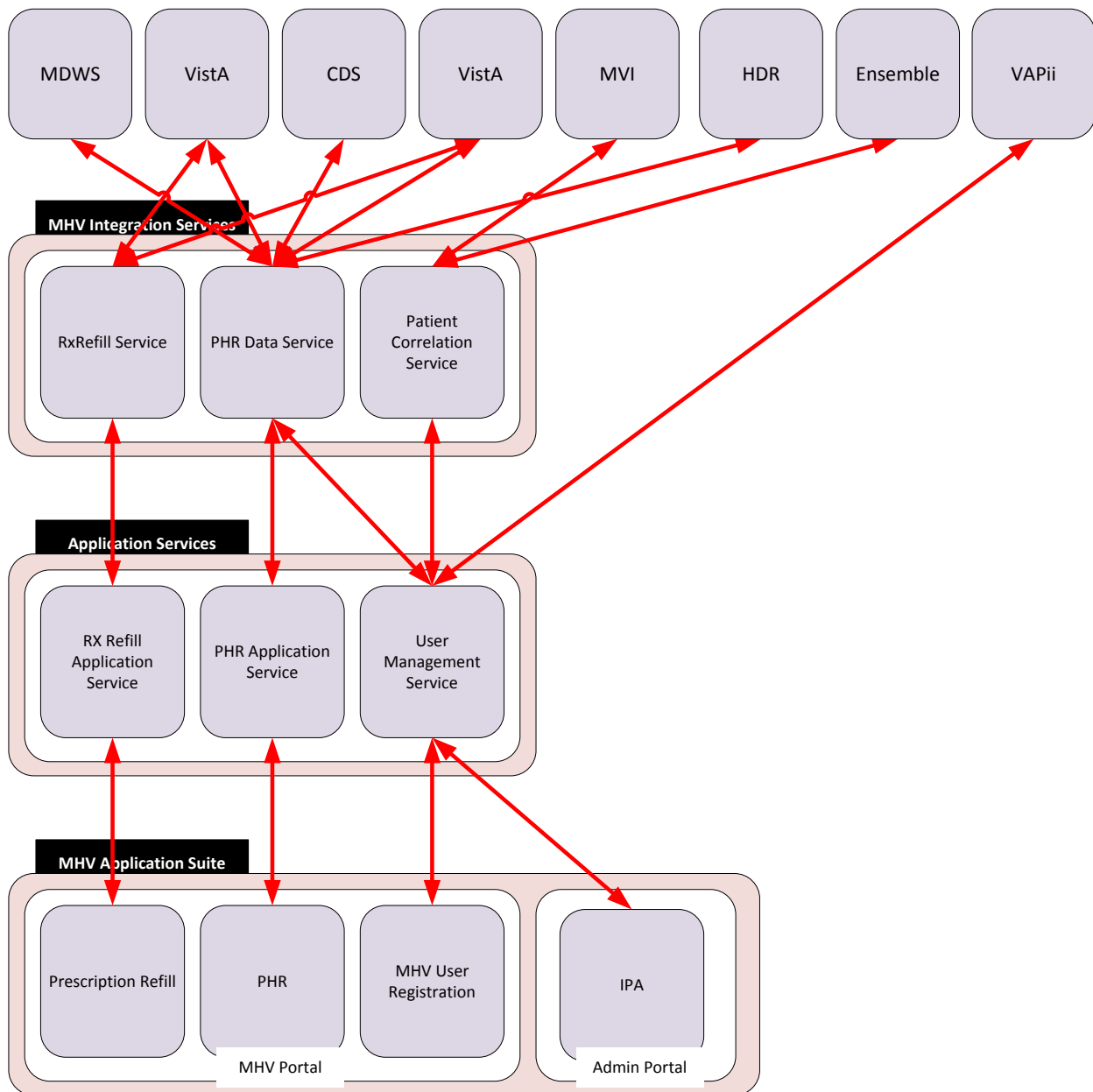
MHV Communications Architecture is shown in the image below.

Figure 30: MHV Communications Architecture



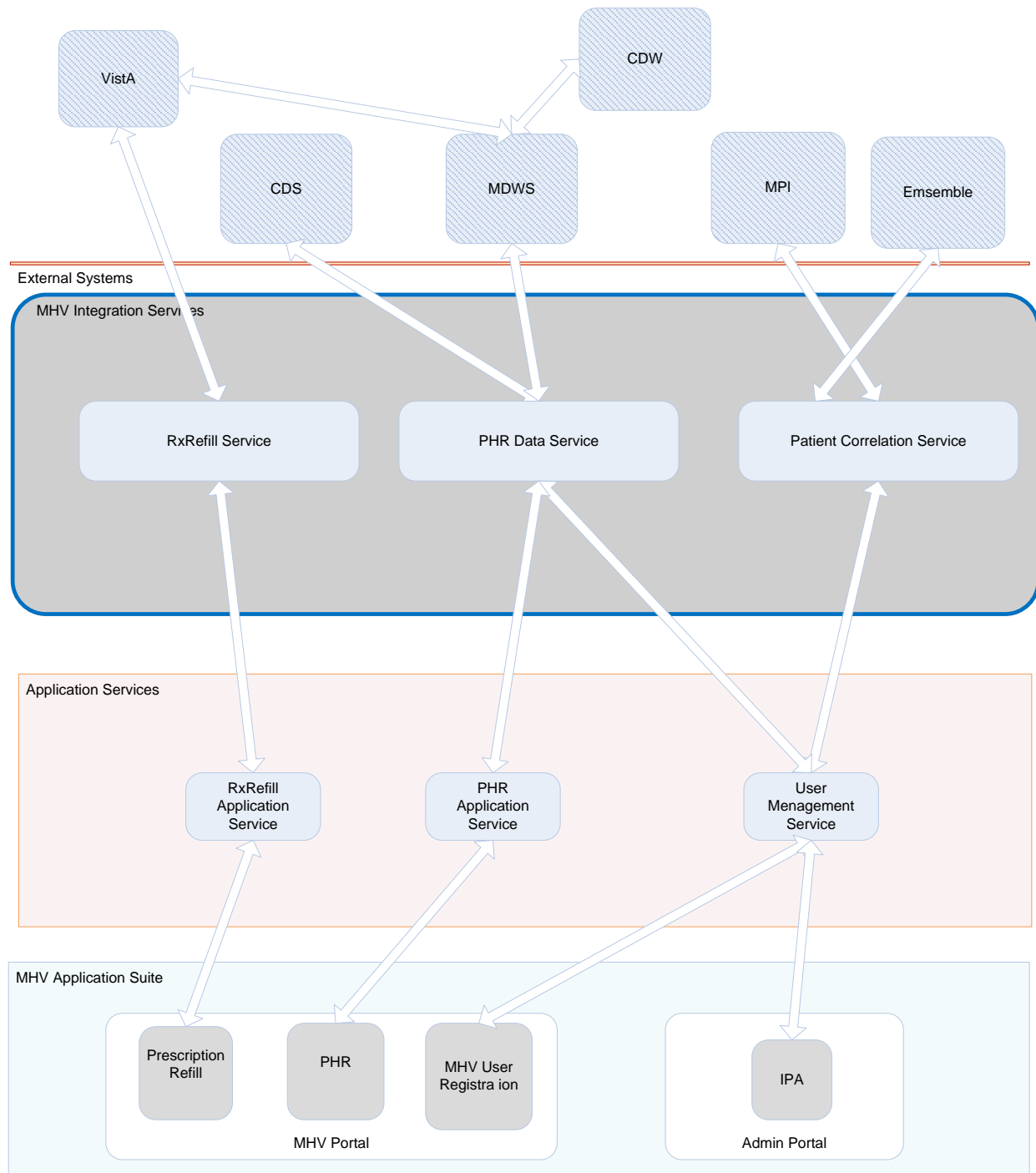
MHV Communications Architecture are shown in the image below.

Figure 31: MHV Communication Paths



External Communication with MHV layout is shown in the image below.

Figure 32: External Communication with MHV



Pharmacy—VistA

Wellness Reminders—from Vista via MDWS

Appointments—from VistA via MDWS

Allergy—from VistA via HDR through MDWS

Chemlabs—from VistA via HDR through MDWS

MHV—MPI/MVI user registration

MHV—MPI/MVI user updates through Ensemble

SM—VistA through Ensemble

5. Data Design

Oracle 11g is used for database design and implementation. All other development including ETL and reports are external to database.

5.1. Database Management System (DBMS) Files

MHV database is a patient-centric database where entire entities exist in relationship to patient entity. The purpose of the MHV database is to ensure that every database transaction is stored in the appropriate tables and columns to support the associated MHV and SM application.

5.1.1. Database Configuration

The following table details the MHV Database Configuration:

Table 21: MHV Database Configuration

Environment	Database Software	Operation System	Version
INT-A	Oracle 11g	Sun Solaris 10 (64 bit)	Oracle Enterprise Edition 11.2.0.2 (64 bit)
INT-B	Oracle 11g	Sun Solaris 10 (64 bit)	Oracle Enterprise Edition 11.2.0.2 (64 bit)
SysTest	Oracle 11g	Sun Solaris 10 (64 bit)	Oracle Enterprise Edition 11.2.0.2 (64 bit)
Preview	Oracle 11g	Sun Solaris 10 (64 bit)	Oracle Enterprise Edition 11.2.0.2 (64 bit)
L&P	Oracle 11g	Sun Solaris 10 (64 bit)	Oracle Enterprise Edition 11.2.0.3 (64 bit)
Pre-Production	Oracle 11g	Sun Solaris 10 (64 bit)	Oracle Enterprise Edition 11.2.0.3 (64 bit)
Production	Oracle 11g	Sun Solaris 10 (64 bit)	Oracle Enterprise Edition 11.2.0.3 (64 bit)

5.1.2. Database Schema

MHV database is effectively managed by grouping data tables into different schemas based on the application they serve. The following Table is schemas defined in MHV database and their purpose.

Table 22: Database Schema

Schema	Purpose
EVAULT	The eVault schema stores user data generated by My Health eVet Portal. Most of the data are populated through ETL processes and users' input.
PORTALMAIN	The portalmain scheme stores configuration information for the WebLogic Portal framework.
WEBLOGIC	Used mainly by application portal interface to access data from evault, portalmain, sms and sdsadm schema.
SDSADM	Provides facilities reference information Most of the data are populated manually by way of SQL scripts.
SMS	The SMS schema stores user data generated by Secure Messaging Application Portal. Most of the data are user-generated and references SDSADM schema for facilities information.

5.1.3. EVAULT Schema

[SP link](#)



EVAULT_129.pdf

5.1.4. SDSADM Schema

[SP link](#)



SDSADM Schema.pdf

5.1.5. SMS Schema

[SP link](#)



SMS_128.pdf

5.1.6. PHRMGR Schema

[SP link](#)



PHRMGR_schema.pdf

5.2. Non-DBMS Files

N/A

5.3. Data View

See section [5.1.6](#) above.

6. Detailed Design

This section describes the proposed MHV design in detail. The audience for the information contained within this section is the Product Development technical staff. The expectation is that the PD technical staff will use the information and guidance provided within this section to understand how current systems will need to be refactored and new capabilities designed and implemented to align with the architecture defined herein and by the OneVA Enterprise Architecture.

As MHV is an existing product the requirements for this product have been documented and are included in the program Requirements Traceability Matrix (RTM).

6.1. Hardware Detailed Design

MHV Systems are planned for deployment to Cloud Infrastructure-as-a-Service (IaaS) that includes all hardware system level resources necessary for operations of the application. The design of the cloud infrastructure is being pursued as part of the MHV Master Cloud project and is thus outside of the scope of the MHV Site Redesign effort.

Hardware systems that will host redesigned MHV systems will be designed to support horizontal scaling of hardware systems to meet increasing application load. All MHV application components will support running in a clustered configuration, which in simple terms, means the ability of application components to be deployed across multiple commodity hardware systems that coordinate their efforts to handle the total application load. [ETAC 2.2.7 – Scalability]

The hardware infrastructure for MHV is largely virtual, being hosted as virtual machine guests within a set of VMWare ESXi clusters. The selected hosting provider has deployed the MHV environments in a highly redundant virtualization stack consisting of Cisco UCS blades for virtualization hosts utilizing NetApp storage filers. The virtual clusters are maintained with at least n+1 host redundancy and are actively replicated to the standby DR site using either VMWare SRM or NetApp Snapmirror. Networking to the VA is accomplished via the hosting provider's established Business Partner Extranet connection to the VA. No other network access to the MHV enclaves exists other than to traverse the VA controlled BPE connection. The following diagram details the network and virtualization infrastructure at the primary hosting site in Culpeper Virginia.

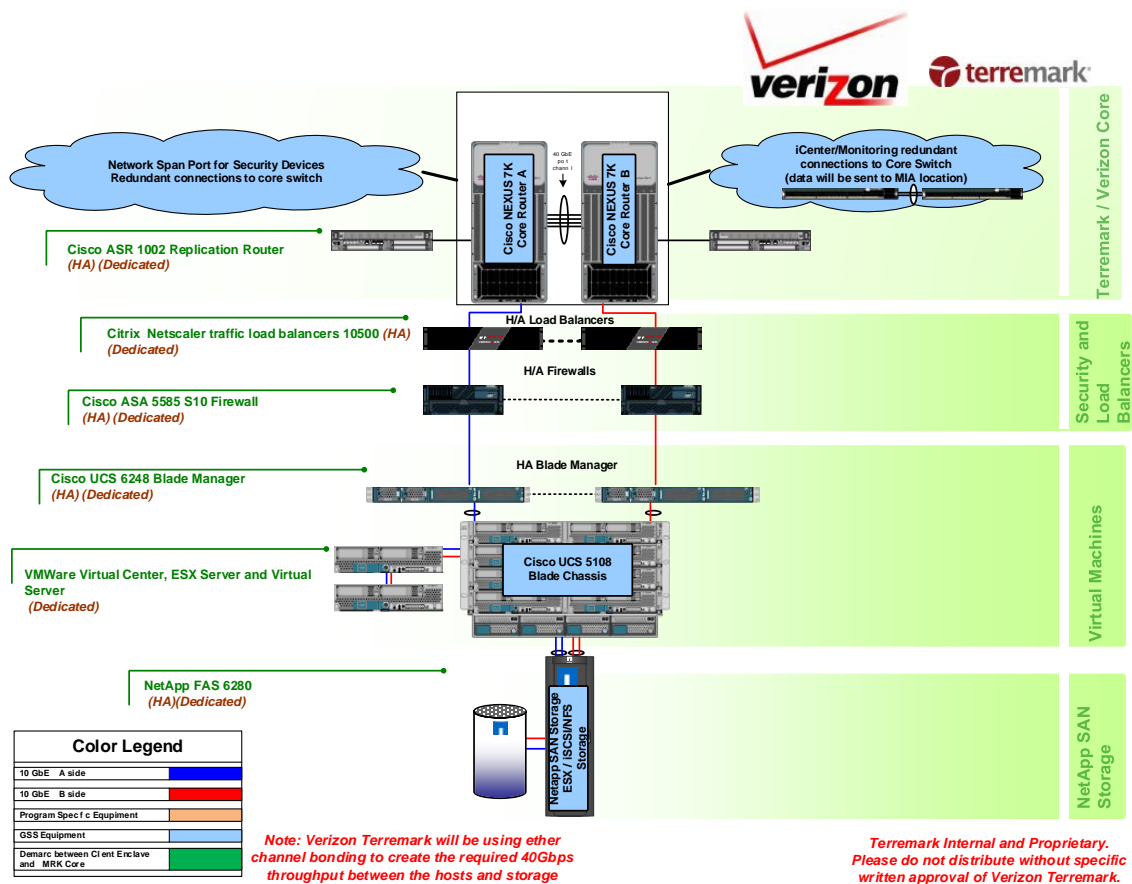


Figure 33: Hardware Detailed Design – Terramark Culpeper, VA

The above figure displays the Hardware Detailed Design of the Terramark Culpeper, VA installation.

Off-site redundancy is provided via a nearly identical hardware stack at the secondary hosting site in Miami Florida.

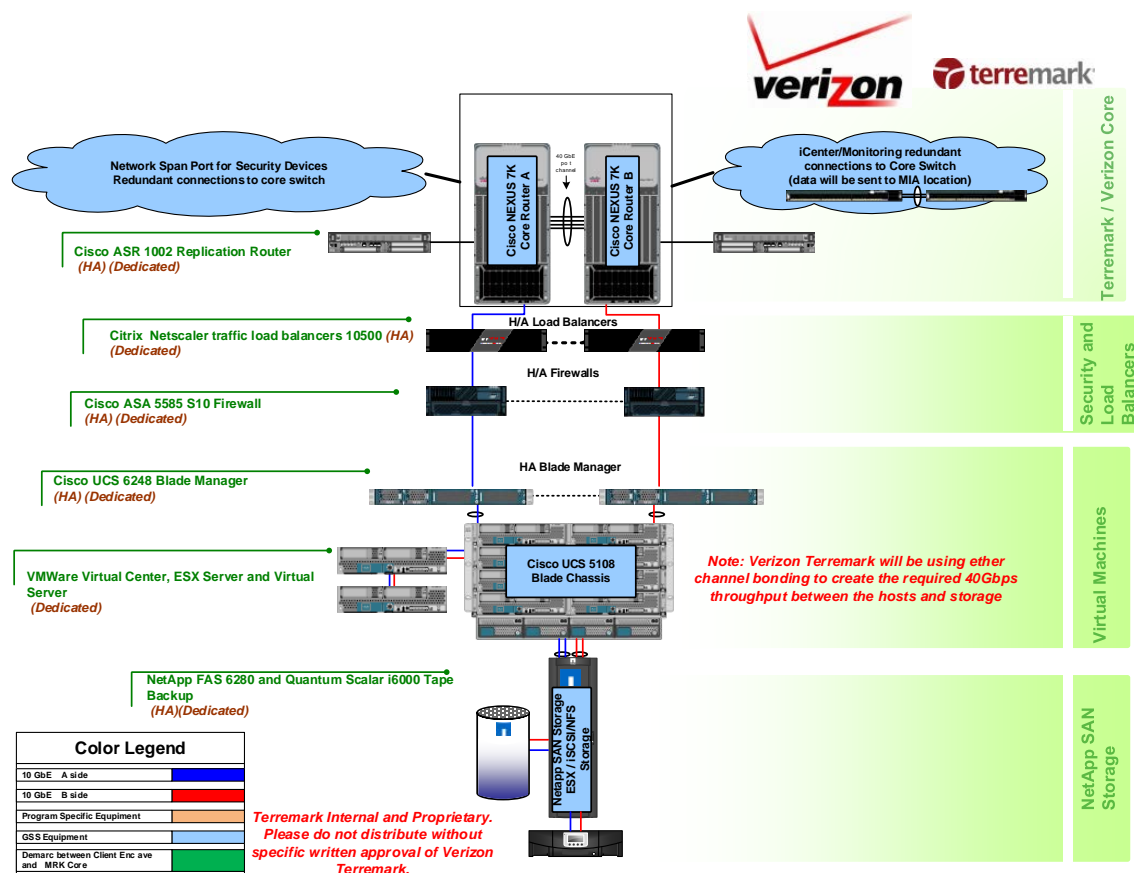


Figure 34: Hardware Detailed Design - Secondary Site Terramark Miami, FL

The above figure displays the Hardware Detailed Design of the Secondary Site Terramark at Miami, FL.

To meet the needs of the MHV project, additional resources and capacity beyond those typically provided by the hosting provider were required. To meet database performance requirements, database hosts in Development, Performance, Pre-Production and Production are implemented in hardware rather than virtualization. To maintain a high level of performance capacity, dedicated database hosts for Performance, Pre-Production and Production are connected to solid-state storage arrays via Infiniband 40Gbps links. The hosting provider provisions co-location space in close proximity to the virtualization hardware stack at each site. Front-side network connectivity is provided via 1Gbps Ethernet. Storage network access for spooling backups and archival data to the virtualization storage filers is provided via 10Gbps Ethernet. The diagrams below detail the conceptual layouts of the virtualization and co-located databases for each enclave within the primary hosting site at Culpeper Virginia and the secondary hosting site at Miami Florida.

Studies and analysis of current data storage capacity will be performed in addition to forward looking analysis of aspects of the system that could dramatically impact capacity requirements (e.g. Secure Messaging Attachment Storage Requirements) in order to identify storage capacity. A report of current and forecasted storage needs will be maintained and reported to MHV management in order to ensure data storage capacity in the MHV production environments is meeting the need. [ETAC 2.4.6 – Storage]

6.2. Software Detailed Design

The following sections define the detailed design or architectural guidance that MHV developers will use to transition existing components and implement new components of the MHV application. The objective of this detail design section is to ensure that MHV technical resources have the information they need to be able build MHV components that are consistent with the MHV Site Redesign architecture and to ensure that MHV systems and components are well aligned the OneVA EA.

6.2.1. Conceptual Design

The design of the MHV enterprise application is specified as a set of services that operate in concert to provide a platform whereby Veterans or their representatives can view and track their health information as well as interact with their healthcare providers online. Typically, this type of architecture is termed, Service Oriented Architecture (SOA). In this style of architecture, systems are segmented into both functional components and logical layers. Functional components are cohesive groupings of interfaces, business logic, and data necessary to fulfill a functional aspect of the system. Logical layers within an application provide boundaries within and between application components. The objective of designing a system in this way is to deliver a system that is more flexible, that promotes a clean code base, that promotes ease of testing, is more scalable, and that promotes more opportunities for reuse throughout the enterprise than a non-layered / non-tiered application could. [ETAC 2.2.1 – N-Tier Architecture, ETAC 2.2.5 – Test Driven Development, 2.2.7 – Scalability, ETAC 2.6.1 – System Integration]

In order for developers and other technical resource to be successful in implementing MHV system components that are consistent with these design principles, they must understand how an assigned task and the affected implementation apply these software design patterns. For example, given a particular task, the developer must understand which components and logical layers will require modification in order to provide the desired capabilities. Given the simple example of adding a new field to a form or list for display as part of a browser based or mobile application view, what changes need to be implemented and, more importantly, where should they be implemented (which layers) to ensure compliance with the architecture. There are two approaches to identifying the impact to the system. A top-down approach starts with the presentation layer and proceeds down-stream through business, messaging and data layers in order to assess overall impact to systems. A bottom-up approach starts by assessing impact to the data layer (physical and logical data models) and proceeding upstream, identifying impacts to business and presentation layers. Once the total impact has been assessed and captured, a plan for implementation can be created which sequences changes to the system. If interfaces between systems are well defined, work within the various layers should be able to proceed concurrently by different parties to expedite changes.

Additionally there are considerations for integrating with services that may be either internal or external to MHV and maintaining consistency with this design and with enterprise architecture guidance. Finally, consideration must be given during the design phase of how the components proposed for development might be deployed at the data center or to the Cloud such that there are the greatest number of options for leveraging systems and system resources.

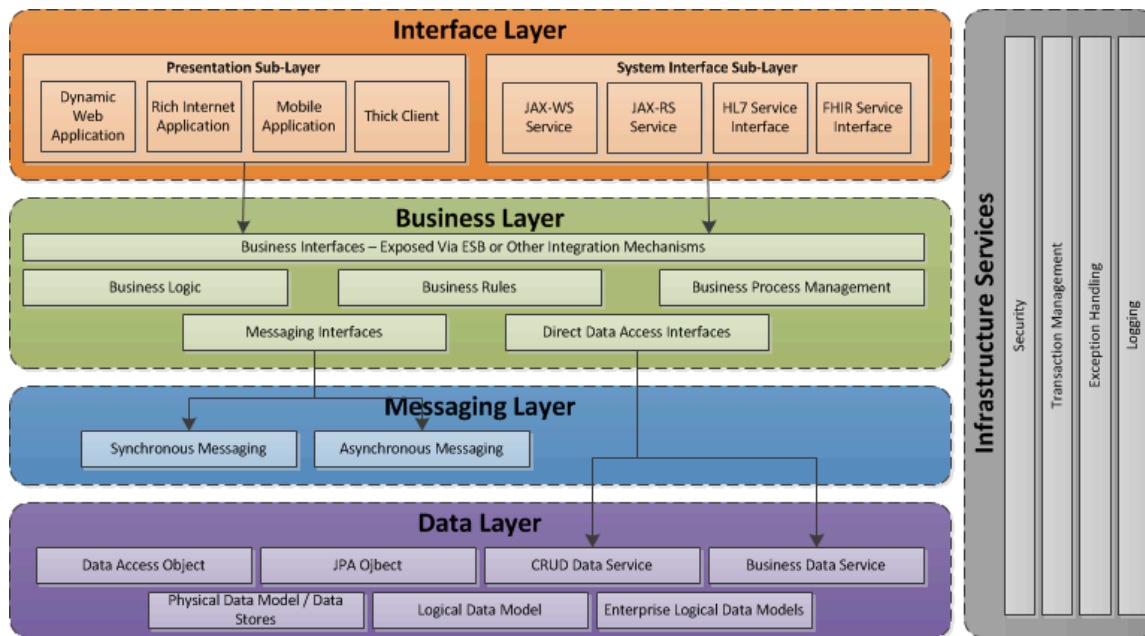


Figure 35: MHV Site Redesign Logical Model

The above diagrams illustrate a conceptual view of logical layers in an application and the types of components that live within those layers and how these logical layers map into a component model of the design.

The logical model above demonstrates how SOA application components are mapped into different logical layers of the application. The layers are described below.

- **Interface Layer** – The interface layer of the application contains components of the application that interface with users, systems, or devices. This is the layer of the application where presentation components and system interfaces should be implemented. All interaction between the interface layers and the other layers will happen via well-defined APIs.
- **Business Layer** – The business layer consists of business objects where business logic & business rules are implemented. Furthermore, business process management capabilities might be employed to allow the application to adapt to support different business processes or rules over time. This type of capability might also be implemented as an infrastructure service. The business layer interacts with other services and with the data layer through messaging between objects.
- **Messaging Layer** – The messaging layer provides interfaces and concrete implementations for interactions between application components and between applications. At the most basic layer, messaging can occur within an application as one object invokes a method of another object. More complex messaging paradigms also exist that support synchronous and asynchronous messaging between the business layer and other applications via remote protocols such as SOAP based web services (JAX-WS) and Java Messaging Service (JMS). The key to maintaining loose coupling of business layer objects with an underlying messaging implementation is to have your business objects interact with messaging provider implementations through interfaces. This will allow the application to be more flexible and utilize any concrete messaging

provider (e.g. POJO, EJB, Web Service) without being tied to the specific messaging implementation is to utilize an interface. [ETAC 2.6.1 – System Integration].

- **Data Layer** – The data layer defines the set of services, classes, and schemas (both logical and physical) that support an application component. Data services and business data services provide interfaces and implementations that the business layer interacts with to perform persistence logic. Using this approach, the business objects need not understand and should be completely decoupled from the underlying physical data model of an application. This style of design removes barriers to information sharing across the enterprise by ensuring that data consumers are insulated from changes to physical data models and specific implementations. [ETAC 2.2.2 – Data Independence] Persistence providers facilitate physical data storage and are implemented using Data Access Objects (DAOs) and Object Relation Mapping (ORM) technologies, such as Java Persistence API. Logical data models are employed to define message payload models to be utilized to transmit data between the data layer and external consumers of data layer services. These logical data models promote increased interoperability by utilizing known standards that consuming applications have implemented. Support for external standard logical data models, such as those defined by the Enterprise Logical Data Model (ELDM) (e.g. HL7 3, NEIM, FHIR, etc.) might also be considered for implementation in cases where increased interoperability is desired or required. [ETAC 2.6.1 – System Integration] Finally, physical data models such as those defined by relational database schemas and data definition languages are included as part of the data layer and act as the physical storage specification for any data persisted by the application component.
- **Infrastructure Layer** – The infrastructure layer is the set of crosscutting services that provide the infrastructure for SOA systems. There are a variety of these service provided by the enterprise. Several of these services are include in the illustration above. The MHV application will draw from infrastructure services to build a platform through which MHV services can be exposed to the enterprise. MHV will integrate with Identity and Access Management (IAM) enterprise shared services for securing the MHV APIs. These services include authentication, authorization, specialized access control, and security auditing purposes. Additionally, MHV application services could be streamlined to take advantage of additional IAM services for Identity Proofing and Provisioning of services that are currently managed internally by MHV application components. In order to become a first class service provider within the VA enterprise, services (APIs) exposed by MHV systems shall be registered with the enterprise Service Registry, which supports the dynamic discovery of MHV services to the enterprise and ensures that services offered by MHV are not duplicated by other organizations. [ETAC 2.6.2 – Service Registry, ETAC 2.6.3 – Shared Enterprise Services, ETAC 2.6.4 – Identity and Access Management (IAM) Service].

It is important to note that, as developers start building new or transitioning existing components into this architectural scheme, that care is taken to ensure proper alignment with the architecture.

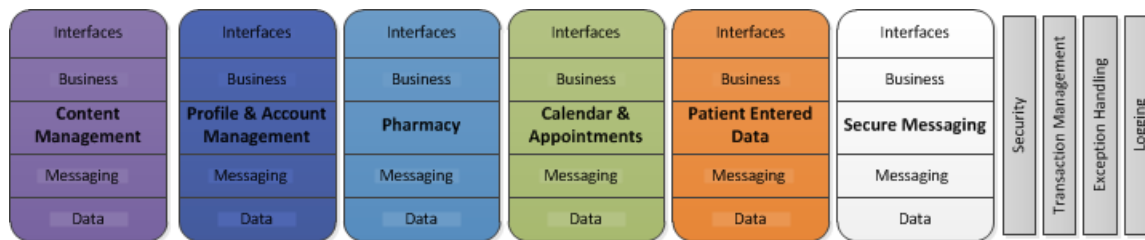


Figure 36: MHV Site Redesign Functional Model

The above diagram shows how these logical SOA layers could be applied to the MHV application components.

Another thing that must be considered and planned for in the design of MHV system components is how they will be deployed. Application components can be deployed in many different configurations and, for many reasons, may not all be deployed to the same application container. In some case, it may make sense to deploy the business objects and service interfaces to an application server cluster that has been setup specifically to offload workload to minimize impact that these services have on systems supporting the interface layer. Additionally, some components may have strict SLAs attached to them that require them to respond quickly to client requests, while other components may not. Isolating components with strict SLA requirements can further ensure that those SLAs will be not be impacted by less critical aspects of the system. The goal is to ensure as much flexibility as possible to the deployer so that they are able to meet the SLAs that have been defined, while optimizing system usage to keep operation costs of the system low. Application developers can develop application components that support flexibility through implementation of interfaces that define contracts between application modules and between the logical layers of the application. The following recommendation is being proposed as the standard project layout for MHV application components. This layout provides mapping to the layered SOA architecture defined by the enterprise (OneVA ETA) and a strategy for migration from current architecture to the MHV Site Redesign architecture.

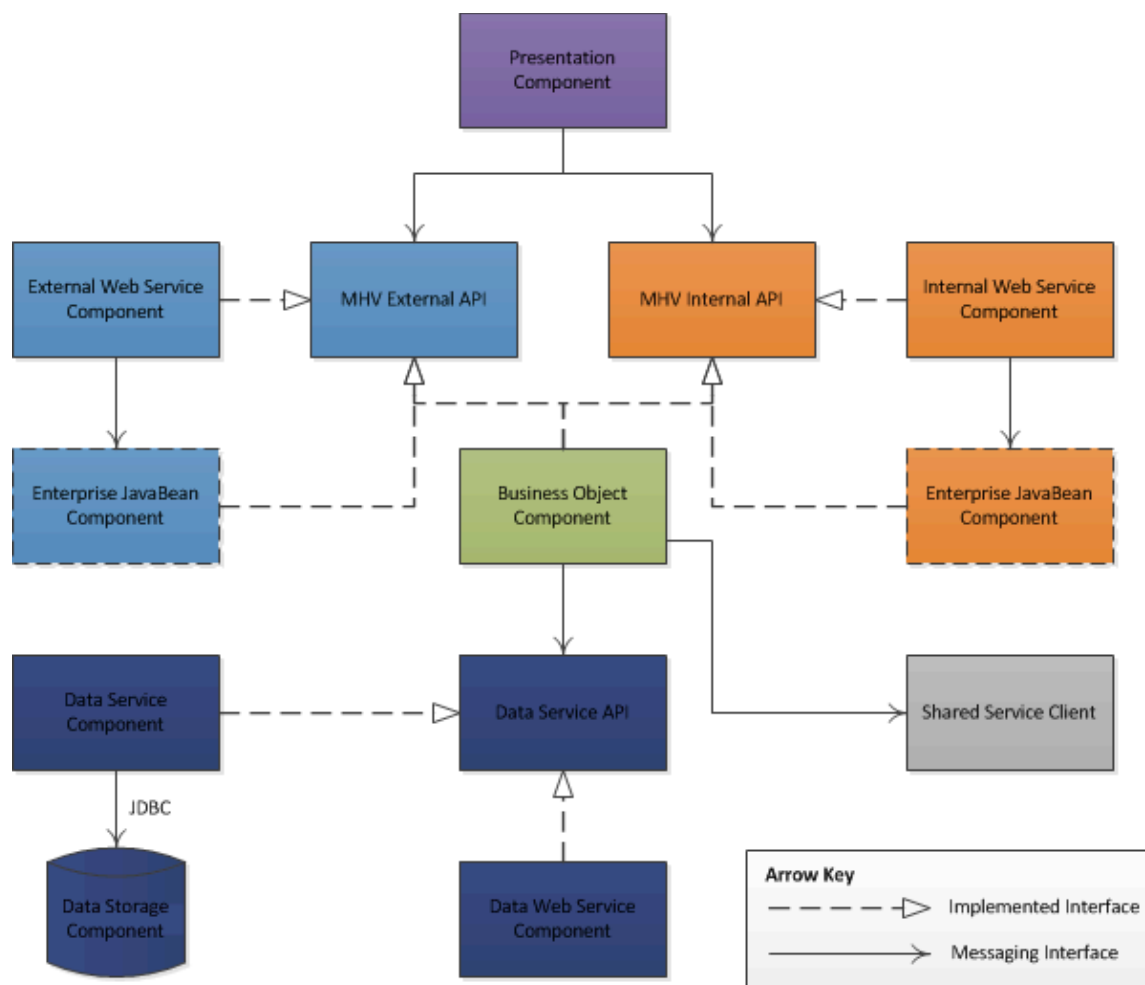


Figure 37: Project Component Breakdown Structure

The above layout provides mapping to the layered SOA architecture defined by the enterprise (OneVA ETA) and a strategy for migration from current architecture to the MHV Site Redesign architecture.

The purpose for each component type is described in detail below.

- **Presentation Components** – Project Module that encapsulates logic, which facilitates the user interfaces to the system. These components will provide the only stateful components of the system and shall interact with the business layer through well-defined interfaces. No data access shall be performed directly by presentation components.
- **External APIs / Interfaces** – Project Module that contains external interface components. These interfaces define the external system integration points or domain boundary of the system. It is at this layer that verification that appropriate security tokens have been provided by the calling application or component. It is expected that the interfaces presented in the External APIs will be the most mature, most robust interfaces to the system and are not expected to change significantly over time.
- **External Web Service Component** – Web Service module that implements one or more external APIs and exposes them as web services for consumption by external and internal systems

- **Internal APIs / Interfaces** – Project Module that contains interfaces exposed internally within MHV but that are not release for external consumption
- **Internal Web Service Component** – Web Service module that implements one or more internal MHV APIs for consumption by MHV components via web service remoting technologies
- **Enterprise Java Bean Component** – Project Module which exposes J2EE EJB components for consumption by web services and serves as the entry point to the business layer and serves as the glue between the external system interfaces and the business layer (optional)
- **Business Object Component** – Project Module which encapsulates a cohesive set of business capabilities that it exposes through well defined (external or internal) interfaces or APIs. These interfaces serves as the contract between the business layer and the presentation and system interfaces to the system.
- **Shared Services Client** – Client implement project module that the implementation of client for interacting with external enterprise shared services
- **Data Service API** – Project Module with encapsulates the interfaces or APIs for interacting with the data tier components of the application
- **Data Service Component** – Project module that includes the implementation of a specific data service. Depending upon the type of data service, this might be the dynamic data language component for interacting with the physical data store or, in the case of business data service will act as an intermediary which abstract the physical data model implementation and the business tier objects. These business data services would also provide the implementation of necessary to expose service data through Enterprise Logical Data Models.

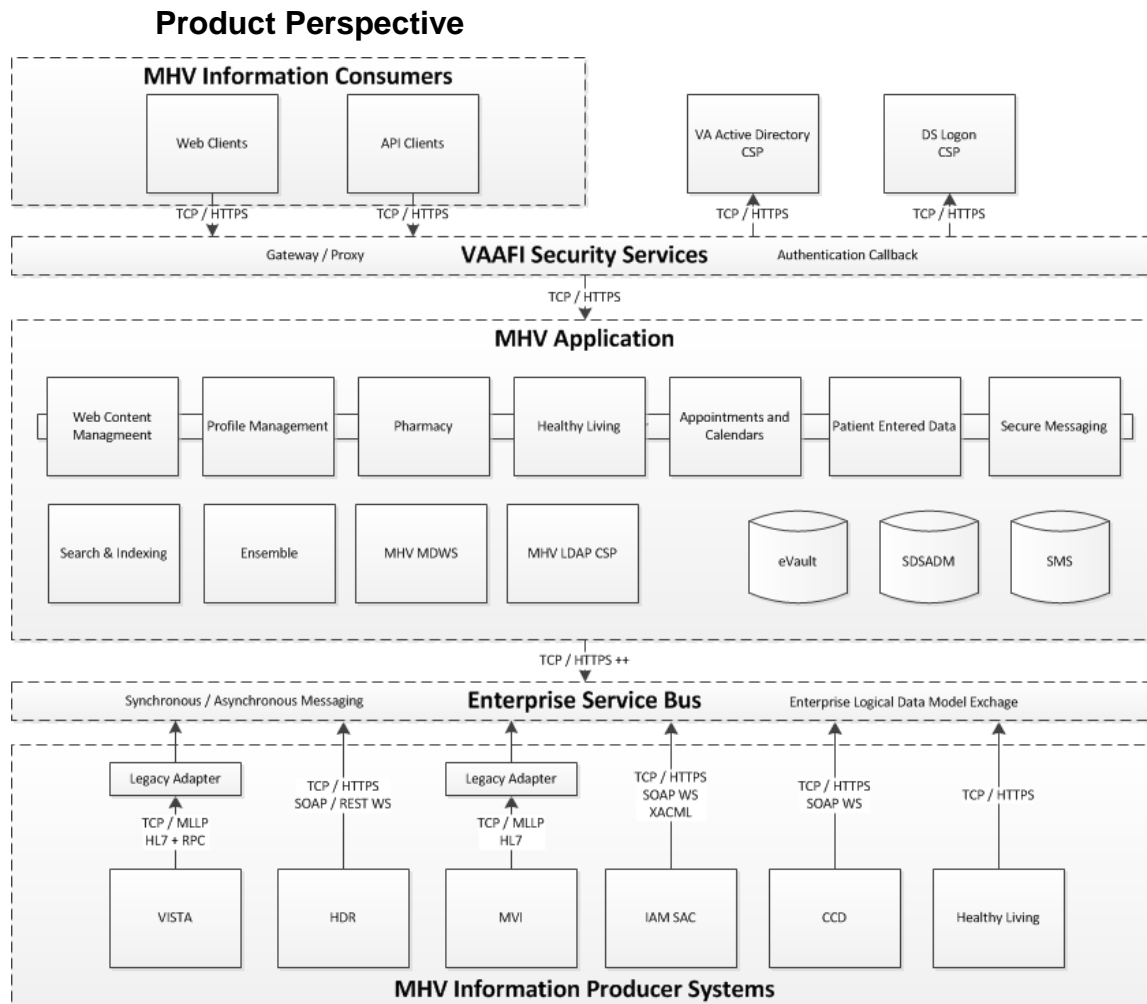


Figure 38: MHV Product Perspective Block Diagram

The above block diagram illustrates the major components of the MHV system, interconnections and external interfaces.

The product perspective diagram is broken down into three segments: Information Consumers, the Application, and Information Producers. VAAFI security services are represented as a layer that provides the gateway between information consumers and the MHV applications and services. This layer of the application is responsible for the security of the MHV application. These security services include authentication, authorization and auditing of requests.

6.2.1.1. User Interfaces

This section is N/A to the MHV Site Redesign Project. The MHV Site Redesign project shall not modify or alter user interfaces exposed to MHV system users. As such each of the MHV Portal, MHV Admin Portal, Secure Messaging, and Secure Messaging Administration user interfaces exposed by MHV systems shall not be impacted and shall each be supported once all technical solutions defined by this effort have been implemented.

6.2.1.2. Hardware Interfaces

The MHV enterprise application is designed for an Infrastructure-as-a-Service (IaaS) cloud platform. The design of this cloud platform falls with the scope of the Master Cloud MHV project.

6.2.1.3. Software Interfaces

The following describes the software interfaces that MHV system will utilize in delivery of the redesigned MHV system architecture. Several of these software interfaces will be provided as either COTS products or open source technical solutions. An analysis of all proposed COTS products will be performed to ensure that the provider of any technical off-the-shelf solution is capable of supporting the solutions over MHV's lifetime. The following characteristics will be weighed in the selection of technical solutions. [ETAC 2.6.8 – COTS Products]

- In cases of commercial products, the company is adequately mature and sized to support the product over the lifetime of the application
- In cases of open source products and technical solutions, a large enough and active enough community surrounds the technical solution to give confidence in its continued viability
- The technical solution must be included on the VA Technical Reference Model. If it is not, MHV Site Redesign team members must perform a search for a suitable alternative. If there is no suitable alternative, steps must be taken to introduce the technical solution into the VA TRM. No technical solution will be allowed within the MHV ecosystem without it first being approved by the VA TRM.

HTTP Web Server

Virtualized web servers will be utilized by the MHV systems in cases where static web content must be served up to consumer applications. Modern web servers such as Apache web server are exceptionally robust and mature and well suited for serving static web content to MHV consumers. By serving static web content from web servers, the load associated with fulfilling these types of requests are offloaded from the application servers, enabling these servers to stay focused on fulfilling request for dynamic web content. Apache HTTP Server is proposed for use in the initial MHV Site Redesign implementation. Apache HTTP Server is approved by the TRM. Alternative HTTP servers may be experimented with and / or implemented as specialized needs are identified. Changes to HTTP server implementation should not impact other aspects of the system.

J2EE Application Server

J2EE enterprise application server provides the J2EE container required for hosting MHV components (UI, APIs, etc.). Currently the MHV application consists of both custom (in-house) developed and COTS or Open Source J2EE applications and components. The J2EE application server container provides application clustering, management, concurrency management, data source management, and transaction management to the applications that it hosts. The J2EE Application Server projected for use by the redesigned MHV application is Oracle WebLogic Server. Oracle WebLogic Server is approved for use by the TRM.

Portal Server

Portal Servers provide a mechanism for aggregating information and presenting it to the end-user in a customizable fashion. The Portal server hosts one or more miniature applications called portlets that provide a dynamic window to information relevant to the users. Portlets are a specification based Java technology ([JSR-168](#) and [JSR-286](#)). There are several Java-based and non-Java based Portal servers available on the TRM today. The current Portal server utilized by the MHV website is WebLogic Portal. In an effort to reduce costs and risk of the program the MHV Site Redesign effort is working to identify a modern Java based portal solution consistent with current program and VA needs. Many TRM approved solutions are available.

Web Content Management System

Web Content Management Systems (CMS) provide mechanisms whereby non-technical resources can publish web content. The selected vendor must support deployment to commercial or open source application servers supported by the TRM, integration with major database vendors, clustered configuration, workflow management, WYSIWYG HTML editing capabilities, integration with search platforms, and the ability to stage updates prior to publishing site content. There are several Web CMS approved for use by the VA. Team technical staff is considering several options based on consistency resource skills and migration path from the current solution.

Document Management System and Content Delivery

Document Management Systems provide workflows and publishing capabilities for media such as images and video for consumption by end users. Content delivery addresses how to efficiently deliver media content to system end users. Content Delivery Networks (CDN) provide the ability to distribute content to geographically dispersed locations and then choosing an appropriate CDN node to service a request for a particular media file based on the client system's physical location. The select Document Management vendor must support capabilities for implementing workflows for publishing new media and support integration with various CDNs available in the marketplace today. Team technical staff is considering Liferay Community Edition and Alfresco Document Management Systems. Both solutions are approved for use by the TRM.

Apache Solr (4.5.1)

Apache Solr is a standalone enterprise search server with a REST-like API. Apache Solr will be utilized by MHV systems for indexing of web content in support of search operations. Solr uses Apache Lucene search library to perform indexing and searching of site content. Solr is the standard search platform supported by Liferay server. Additional information can be found on the [Solr](#) website.

Oracle 11g R2 Real Application Cluster Database Server

Oracle Real Application Cluster (RAC) database server will host the physical data required and generated by MHV applications. Oracle RAC provides a solution for horizontally scaling the physical data storage mechanism to meet increased demands on systems. Oracle RAC is a standard data storage service supported by the application server (Oracle WebLogic Application Server) and is a TRM approved tool.

6.2.1.4. Communications Interfaces

Communications between MHV application components and external systems and users will be performed across TCP/IP networks using secure protocols. MHV is a web application and as

such, most communication between end users and external system will utilize Hypertext Transfer Protocol Secure (HTTPS) as per VA Information Security Direction 6500. MHV may also support communication protocols such as T3S, a secure proprietary protocol defined by Oracle WebLogic Server for perform Remote Method Invocations. This protocol will be used in cases where requirements stipulate use of Java Messaging Service (JMS) APIs and or Enterprise Java Beans as a means for integration.

6.2.1.5. Memory Constraints

Memory constraints for MHV systems shall reflect memory constraints of the software and applications installed on those systems. These constraints will be determined based on the requirements of the software being hosted by the systems and the deployment configuration of the applications being hosted on the systems. The redesigned MHV systems shall be hosted by a cloud provider. The specifications and characteristics of these cloud-based host systems are currently being developed by the MHV Master Cloud project.

6.2.1.6. Special Operations

This section is N/A to the MHV Site Redesign Project. Backup, recovery, and archiving operations are performed currently by operations and data center staff and are determined based on business and functional requirements, service level agreements between the MHV program and the data center, and through memorandums of understanding. These characteristics of the MHV system are outside of the scope for the MHV Site Redesign project.

6.2.1.7. Product Features

The purpose of the MHV Site Redesign project is to design a system architecture for MHV that modernizes the MHV technical architecture and aligns the program with the VA strategic missions and initiatives. The features of this technical architecture will address the following MHV requirements.

- Increases data visibility and accessibility through introduction of APIs for accessing MHV data programmatically
- Increase data interoperability through implementation of Enterprise Logical Data Models that are consistent with VA standards
- Increase infrastructure interoperability by redesigning MHV systems to support deployment to a Cloud infrastructure platform and horizontal scalability across commodity server platforms
- Enforce information security through integration with enterprise services that enforce identity and access management
- Decrease program cost and complexity through integration with enterprise shared service

6.2.1.8. User Characteristics

See Section 3.1.4: Application Users

6.2.1.9. Dependencies and Constraints

- Solution must support Veteran centric mission need or capability
- Solution must be compliant with the appropriate business architecture
- Application must be partitioned into logical layers (i.e. presentation, business, logical, data access) with each layer containing functionality specifically related to that layer
- Application layers must expose interface components that promote loose coupling between layers
- Application business logic and data management process must be fully decoupled
- Application user interfaces must follow enterprise common UI templates and style guidelines
- Application must store data on enterprise servers and not on end-user devices or workstations
- Unit tests must be developed for application functions and publicly exposed methods
- Application must implement procedures for communicating and resolving unhandled exceptions
- Application must be designed to scale horizontally and designed to operate on a series of loosely coupled commodity platforms
- Application must be designed to scale out without requiring code changes
- Application business logic must be stateless (i.e. session information is not stored within the business logic layer)
- Application user interfaces must be Section 508 compliant
- Application must comply with VA Enterprise Architecture published data standards (NIEM, HL7, LOIC, SNOWMED, VIM, HITSP, and FHIR)
- Authoritative information sources must be identified and leveraged for data retrieval and manipulation
- Information captured by the Application must be syntactically and semantically harmonized with the VA Enterprise Conceptual Data Model (CDM)
- Application must function optimally using information from the authoritative source or receive permission for caching data locally
- Data gathered and generated by the Application must be defined and registered in the VA Meta Data Registry
- The System must adhere to the VA Cloud First Policy and promote Cloud computing
- System / Server platforms used by the application must be configured using standard system images published in the current VA Release Architecture
- Relational and object oriented databases utilized by the solution must be published in the current VA Release Architecture

- The solution must be designed to operate on the standard OIT defined virtual environments
- Application production capacity requirements must be based on workload analysis, simulated workload benchmark tests, or application performance models
- Application storage capacity requirements must be based on detailed capacity analysis and/or models
- The solution must be designed to operate within the current VA LAN and WAN network configurations
- The deployment environment must meet the performance and downtime monitoring requirements of the solution
- A disaster recovery plan must be developed and provisioned
- All critical infrastructure components (including data) must be located at multiple physical locations
- Solution backup and restore solution must meet data recover requirements (Recovery Point Objectives (RPO) and Recovery Time Objectives (RTO))
- Application user interfaces must be thin or browser based
- Application must adhere to all applicable information security rules contained within the VA Policy Handbook (6500)
- If application is deployed externally, all guidelines for using commercial partners must be followed
- Application must establish secure access paths for accessing the application and application data
- Solution must document specific reasons for all limited, external access to data, including the need to know along with security, privacy and other legal restrictions
- Solution must implement appropriate controls that prevent unwarranted disclosure of sensitive, Personally Identifiable Information (PII), or Protected Health Information
- Solution must be smart-card enabled to handle logical logon using Public Key Infrastructure (PKI)
- All system interfaces (both external and internal) implemented by the solution must be based on open standards such as SOAP, REST, JMS, MQ, SFTP and standard message formats such as NIEM, HL7, EDIFACT
- Application must leverage existing services published in the VA services registry
- The solution must utilize Core common Business Services and Core Common Infrastructure Services rather than deploying local services
- The solution must utilize the Enterprise Identity and Access Management (IAM) service
- The solution must utilize VLER information services where appropriate

- The enterprise information used by this solution must be available through services
- All products and standards used by this solution must be listed and identified as permissible for usage in the VA Technical Reference Model (TRM)
- All COTS products used by the solution must be from mature companies large enough to support those products over the expected life of the product

6.2.2. Specific Requirements

Sections 6.2.2 “Specific Requirements” and all of 6.2.2 subsections will be determined at a later date. Technical Refresh is redesigning MHV and this information will be TDB at a later date.

6.3. Communications Detailed Design

N/A

6.4. Network Detailed Design

The MHV application shall be implemented as a SOA based solution that requires access via TCP/IP networks in order for application components to communicate with both internal and external application components. Because the system is also designed to operate on Cloud infrastructure, discussion of servers and clients become irrelevant at all but the most basic level.

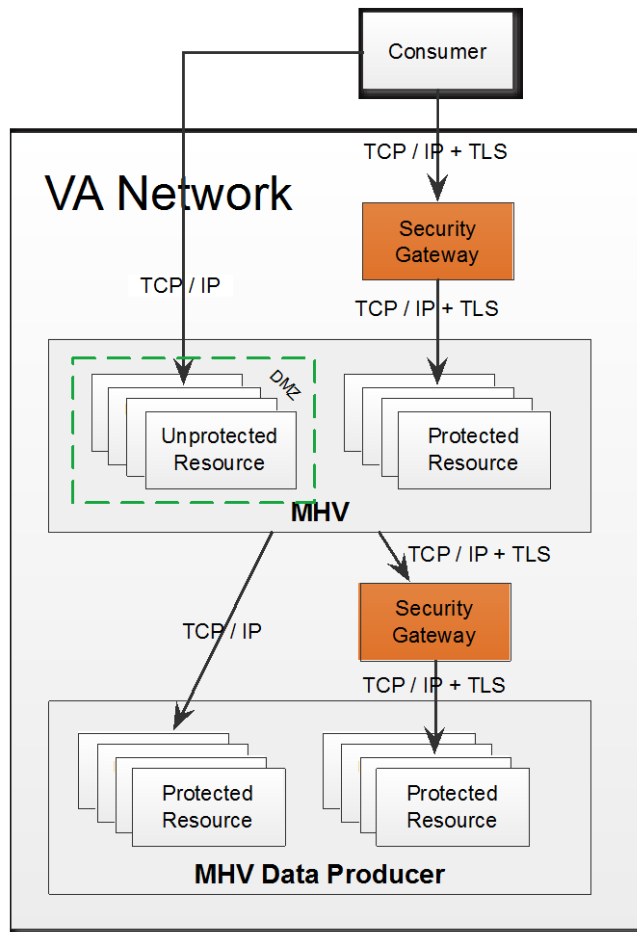


Figure 39: Detail Communication Diagram

The above figure illustrates the communications detailed design.

In the MHV system, there are the following basic client and server systems.

- **Consumer Client Applications** – These are systems or users that consume data or services provided by MHV. There are two types of resources that consumer client applications can interact with. First, unprotected resources. Unprotected MHV resources are system resources that don't require authentication or authorization to be performed in order to consume the information. Unprotected resources are hosted in a special segment of the MHV system network known as the DMZ. Second, protected resources. Protected resources are resources that are protected and require authentication and authorization to be performed before access to these system resources is granted. All consumer client based communication happens via TCP/IP network connectivity between the client system and the system hosting the MHV unprotected resources.
- **Security Gateway** – Security Gateways provide the infrastructure services to verify the identity of the system or user making a request for a protected resource. The security services provided by the Security Gateway include authentication, authorization, and auditing services. All traffic coming in and out of the security gate way is encrypted via Transport Layer Security (TLS). Once an incoming request has been processed by the

Security Gateway and found to be valid, the request is proxied to the appropriate system to fulfill the request.

- My HealthVet (MHV) – MHV systems host both protected and unprotected resources for client consumption. All protected resources must be accessed over network links that are secure (TLS). Verification of access to all protected resources is performed through examination of network packets for the appropriate tokens / headers, provided by the Security Gateway services.
- Data Producer Systems – Data Producer Systems that provide data that MHV exposes to systems or end user in some fashion. All communication with data producer systems is over TCP/IP network links. Whether or not the transport is secure or not depends on the requirements for secure communication that the producer system has. It is anticipated that some data producer systems will expose their services through the Security Gateway services. In these cases, MHV will include the appropriate tokens and headers in the request message being made to the downstream system in order to perform the necessary verification that the request is appropriate.

All network communication between MHV systems and its consumer clients shall be utilize Hyper-Text Transfer Protocol (HTTP) or Hyper-Text Transfer Protocol Secure (HTTPS). Data payloads supported by MHV systems for client consumption are detailed below.

- Hyper-Text Markup Language (HTML) – Consumed by client side browser applications to produce client side user interface
- Simple Object Access Protocol (SOAP) XML – Consumed by system clients requiring standard JAX-WS integration with MHV systems. SOAP is a standards based integration technology published and maintained by the W3C.
- JavaScript Object Notation (JSON) – Consumed by system clients requiring that integrate with MHV system via a Representational State Transfer (REST) web service interface. JSON is a lightweight data interchange format that has been widely used and accepted for system to system integrations.

Specification of the data being represented by the aforementioned payloads will vary greatly depending upon use and is outside of the scope of this document. However, message payloads will follow OneVA Enterprise Architecture guidance for implementing standards based Enterprise Logical Data Models and Coding Systems. By selecting standards based message payloads, a higher level of interoperability with external data consumers and producers will be realized by the program.

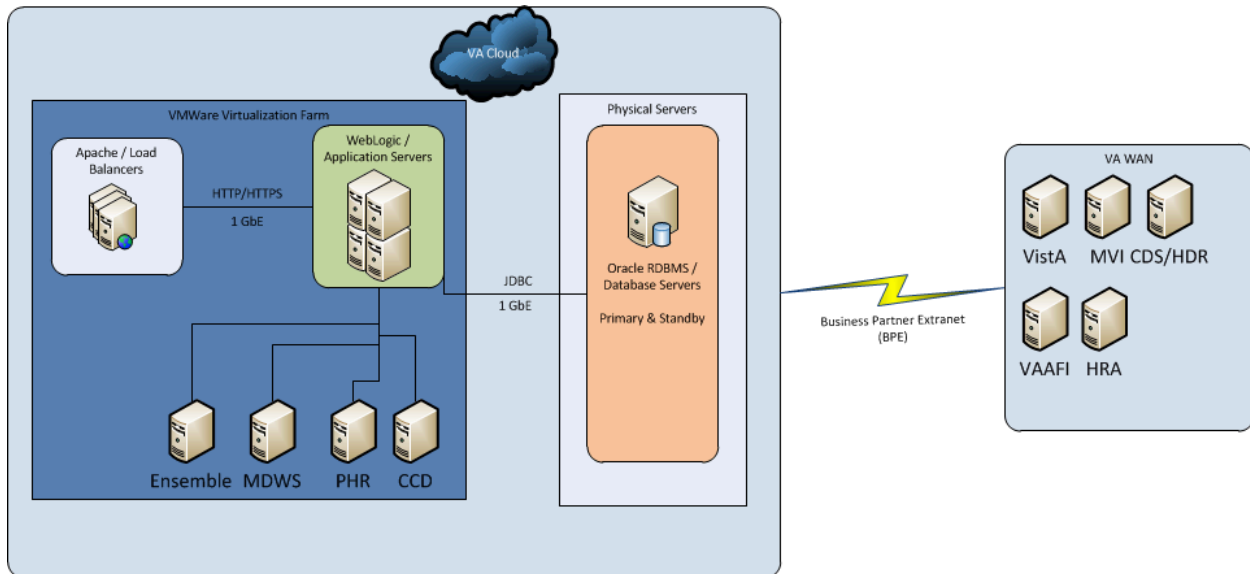


Figure 40: MHV System LAN Topology

The LAN topology provided above details the general LAN topology required and supported by MHV systems. This diagram is purposefully left in general form as the topology of the application is expected to follow a general form within each environment. As environment and systems scale in and out to support increases or decreases in system load, additional nodes will be added or removed from the network in order to adjust to system demand.

6.5. Service Oriented Architecture / ESS Detailed Design

TBD – This will be addressed in a future release. The MHV team is working with Architecture, Strategy, and Design (ASD), the MHV Program Architects, and the MHV Business Owners to develop this design. See [Section 4.5](#) for a high-level overview of MHV approach to SOA and ESS.

6.6. MHVAPI Façade & PHRMGR-R Design

The MHV 12.9 release will provide hybrid PHR functionality to the end user wherein the majority of users will be utilizing existing PHR capabilities, while a select group of field testers will verify that the PHRMGR-R components are functioning as designed. In order to run legacy and redesigned PHR Managers concurrently plus the ability to field test the functionality, changes were required throughout the web, application, and data tiers of the MHV application.

The MHV PHRMGR-R requires a completely updated technical stack which includes upgrades to the Java Runtime (Java 7) and WebLogic Server (12c). Additionally, due to considerations with TRM compliance, many dependencies of the PHR Manager have been upgraded to come into compliance.

Figure 1 below provides a high-level overview of the 12.9 PHR Manager architecture.

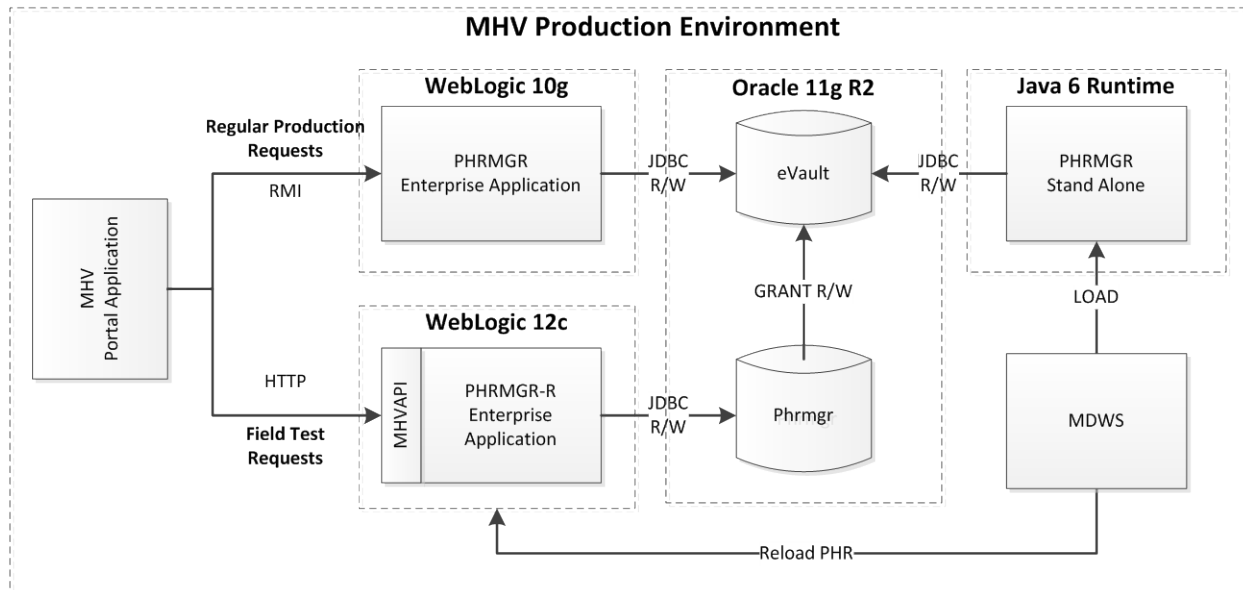


Figure 41: MHV PHRMGR Design

The reengineering of the PHR manager included the design and development of two components: the MHV API Façade and the redesigned PHR Manager. Additionally, changes were required in the data tier to ensure that legacy and redesigned PHR Managers could operate in parallel without issue. Following is a description of each component

6.6.1. MHV API Façade Enterprise Application

The MHV API Façade application component is an enterprise application that provides a centralized facility for aggregating services exposed by MHV. The API façade also provides common facilities for authenticating, auditing, and reporting errors back to its clients.

6.6.1.1.

6.6.2. PHRMGR-R Enterprise Application

The PHRMGR-R Enterprise Application component wraps PHRMGR capabilities and exposes them behind a RESTful (JAX-RS) web service interface. Additionally, the PHRMGR codebase was upgraded to used standard J2EE approaches for managing resources (threads, database connections, and queuing). Additionally, enhancements have been made to the overall workflow of the PHRMGR to support a more streamlined and deterministic approach to refreshing PHR for the user.

The following diagram illustrates the deployment architecture of the MHV PHRMGR-R components and their interrelations.

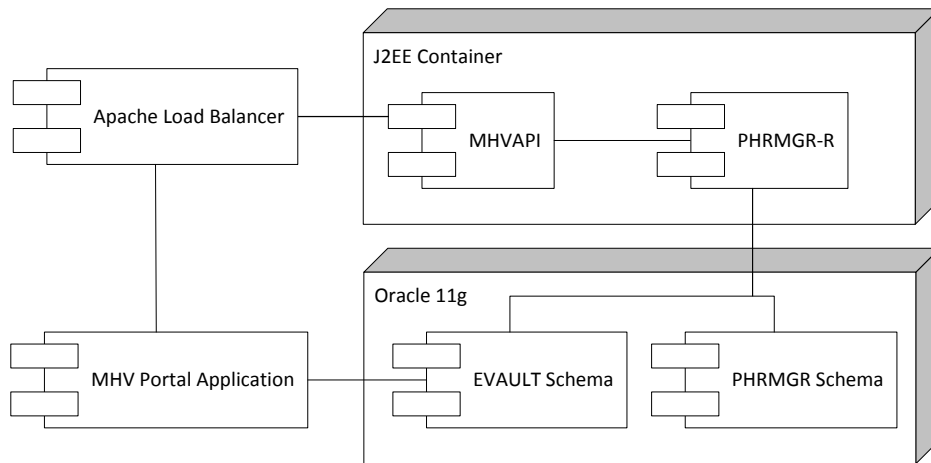


Figure 42: PHRMGR-R Deployment Diagram

Both the MHV API Façade and redesigned PHRGRM are made up of several sub-modules.

- MHV API Façade includes an Enterprise Application Resource (EAR) or container for aggregating services that the MHV application exposes. Additionally, the MHV API Façade component includes a suite of integration tests that validate that the API expose by the service façade are operating as expected.
- MHV Common API includes Java classes that are common, or in other words, may be utilized by many web services exposed by MHV. These common classes include helper classes for authentication, authorization, auditing, and exception or error handling. New service implementations can import these classes and utilize them to implement common service capabilities.
- PHR Manager Redesign implements a JAX-RS based web service for interacting with the PHR Manager services. In addition to the web service, the PHRMGR includes standard interfaces for interacting with both the business and data tier object, Enterprise JavaBeans for performing synchronous and asynchronous business operations, data access objects for persisting information to the database, as well as, the business logic that satisfies the business need.

The following diagram provides a drill-down into each PHRMGR-R component.

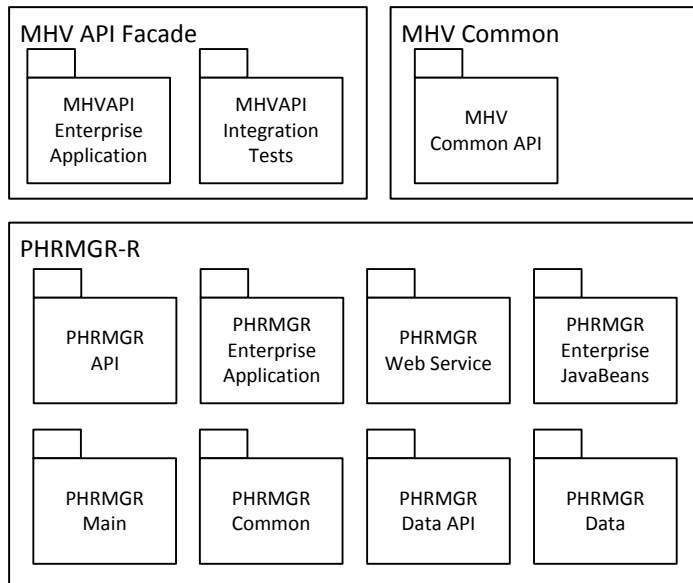


Figure 43: PHRMGR-R Component Drill-Down

A primary objective of the MHV TR architecture is to create system components that are loosely coupled so that changes to individual components have less of a tendency to ripple through the entire code-base. The following diagram illustrates the dependencies between component sub-modules.

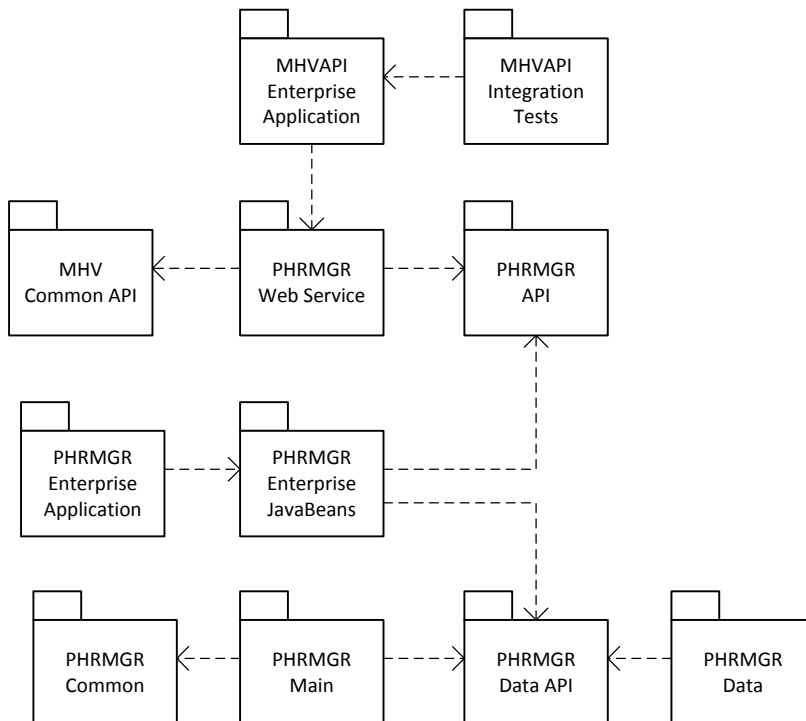


Figure 44: PHRMGR-R Component Dependency Graph

6.7. Heracles Toolkit

Heracles is an extensible, Java enterprise, architecture framework that includes both java components and design patterns to execute business logic, integrate with repositories, and render the user interface. The framework's runtime presence exists within a Spring Framework container that leverages Aspect Oriented Programming (AOP) for transactions and exception handling, Dependency Injection for wiring POJOs, and base data access and service domain objects. Heracles is written and managed by the MHV architecture and development teams.

Heracles represents the new design and development framework for all software systems in the MHV program. New features and projects will be developed using Heracles while older features will be refactored in time. Refer to the Heracles Software Architecture Document for more detailed information on the architectural views of the framework.

6.8. Integration Framework

The Integration Framework is a set of middleware architectural layers with which the My HealthVet application services can integrate to access enterprise level services. The framework is logically divided into two architectural layers, a layer that integrates the MHV application service to the middleware and a layer that forms a message bus.

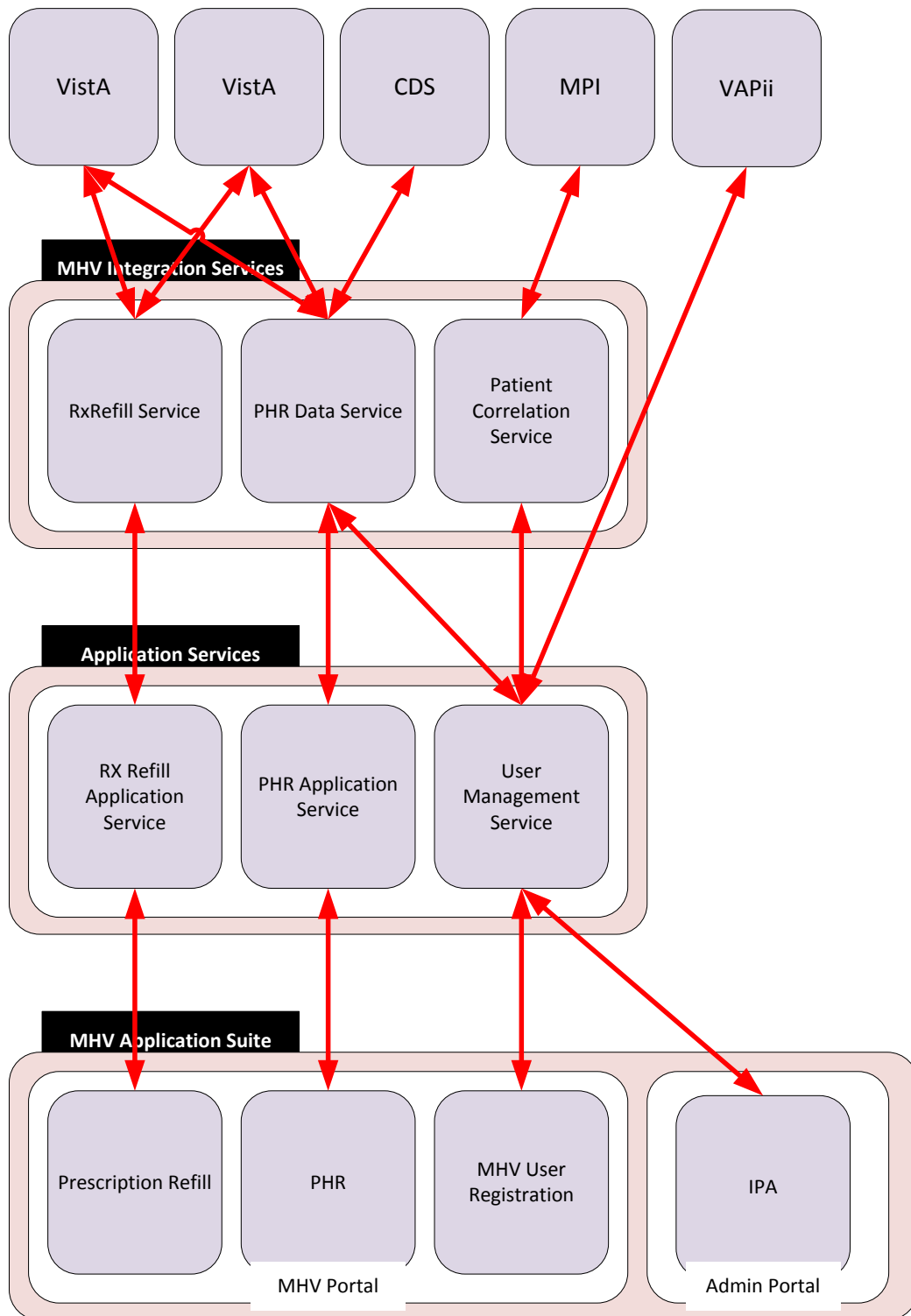
6.8.1. Integration Services

Integration services provides interfaces to enable the integration and processing of specific MHV User or System Requests for PHR Data Extracts, MPI/MVI synchronization, and for the integration and maintenance of the received data into the MHV eVault data store.

Integration services implement a messaging framework that encapsulates communications between MHV Portal and the Integration Framework. Details of communications with external systems are abstracted by the Integration Framework. The Integration Framework maintains the contract with the external system while the PHR Data Service and Patient Correlation Service maintain a contract with the application.

Integration services layout is depicted in the below image.

Figure 45: Integration Services



The Application Services layer in the above diagram is a component of the integration framework that exposes services that are used by the application suite to communicate to external systems.

The application suite must be modified to use the new services exposed by the application services layer for RxRefill, PHR, and MPI/MVI patient correlation. If the application services are deployed outside of context of the application suite, the services will be exposed as EJBs. This is to avoid the need to include MHV application jars in the integration framework project. If the Integration project will be deployed in the same context as the application suite, EJBs are not needed because Spring can be used to expose the services. These services would be exposed in a similar manner to Heracles services that are currently deployed in the application suite.

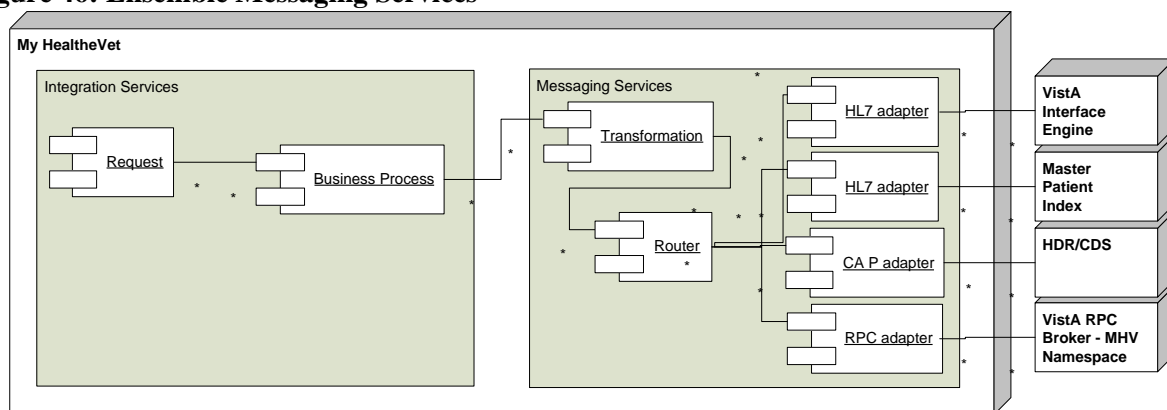
6.8.2. Ensemble Messaging Services

Ensemble has been selected to support messaging services as an interim solution. Ensemble guarantees order of delivery of messages, provides monitoring and exception handling, and provides a clear movement to the HealtheVet v2.0 architecture. A future migration to VIE is planned when it supports in-order processing. Migration to VIE for delivery is the first step in a migration to an ESB solution to be determined.

Ensemble implements the message bus pattern by intelligently routing messages based upon their content and interfaces. The Ensemble rules engine facilitates the routing of messages based on the type and contents of incoming messages. The routing rule set determines the correct destination for each message and how to transform the message contents prior to transmission.

Ensemble Messaging Services layout is shown in the below image.

Figure 46: Ensemble Messaging Services



6.9. Inversion of Control

The MHV portal architecture depends upon an Inversion of Control (IoC) container. An Inversion of Control container is currently leveraged to facilitate only a few capabilities:

- Declarative dependency injection
- Declarative transaction management
- Facilitated unit testing

- ORM Integration
- Declarative resource configuration

The Spring Framework is widely used within the Java community and is leveraged within the Heracles Toolkit.

6.10. Data Access Object (DAO)

A standard DAO tier is utilized to separate the data access interface from its implementation. Within this tier, several data access mechanisms are utilized, with the most prominent mechanisms being Hibernate and LDAP.

6.11. Presentation Control

The user interface components for Toolkit-based applications are developed as portlets using the WebLogic Workshop (WLW) development environment. These portlets will be deployed using WebLogic Portal (WLP). The entry point into the services from the presentation tier is through the service delegate classes from web-tier action classes. In most cases the action classes are developed as part of a Java Page Flow (JPF) in WLW. Presentation components are comprised mainly of JPF, the display tag library, cewolf, netui tags and some other minor tools.

6.12. MPI/MVI and VistA Communication Mechanism

A generic communication backbone supports the transmission and reception of messages originating from and targeted to MHV by other VA applications using a variety of protocols (socket, HL7, HTTP). These communications take two forms: synchronous and asynchronous.

Synchronous communication involves a call to an external system that requires an immediate response. The requesting program waits to receive the response or, in the event of an unreachable system, until the request times out. An example of this anticipated communication is depicted in the figure below in the context of receiving an HL7 message over the network. SM employs this communication type with VistA through Ensemble as middleware.

Asynchronous communication involves one system sending a message to another system and waiting only for an acknowledgement that the message has been received. After the acknowledgement of receipt is returned by the remote system, the local system continues processing normally. MHV will employ this mode of communication when processing demographic update messages from MPI/MVI.

6.13. VAPii Communication Mechanism

The communication between the MHV application and VAPii is one way synchronous communication from MHV to VAPii using HTTPs and socket. For signing the eRAR form online, MHV calls iFrame of VAPii form via HTTPs. For downloading the signed eRAR form, MHV communicated via web services.

6.14. MhvDomainApp Framework

This MhvDomainApp, or Domain Framework, is common to applications operating under the MHV WebLogic Server domain and provides common functionality for application development. Messaging service is a common service that is part of the Domain Framework and provides communication services to VistA systems. Prescription refill uses this service in 8.0,

but will migrate its messaging functionality to the Integration Framework in 8.1. After the migration is complete, the domain framework will be removed from the MHV architecture.

6.15. InterApp Framework

The InterApp framework is a legacy collection of common components that is designed to serve applications requiring its capabilities. Components of InterApp are iaCore, iaControl, iaEAI, iaEjb, iaSchema, and iaService. These components are in scope for MHV architecture as long as there is a dependence on the Domain Framework which relies on InterApp components.

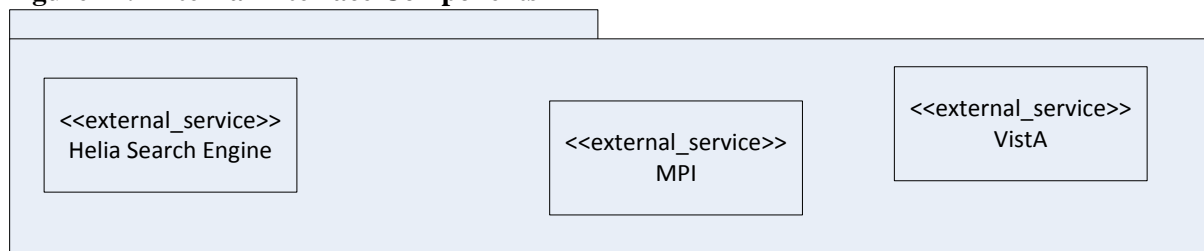
6.16. Secure Messaging Logical View

N/A

7. External System Interface Design

The MHV Portal utilizes services from different outside systems for certain portletApp components, as shown in the image below.

Figure 47: External Interface Components



Helia Search Engine: Provides search capabilities on medical information libraries. Currently this is being used with the HealthWise medical information.

MPI/MVI: Provides a central index of VHA patients.

MVI comprises MPI, PSIM and ToolKit.

VistA: Comprising multiple packages or subsystems each geared toward a specific medical area these systems, located at the VHA VISN centers; provide MHV Portal with medical information on VHA patients.

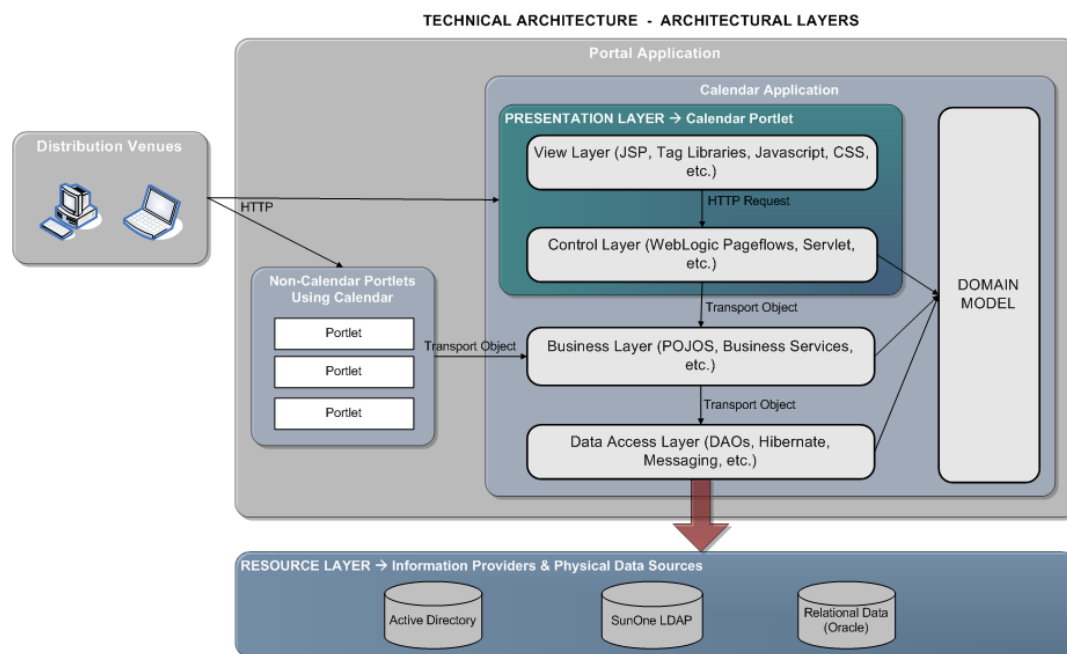
7.1. Interface Architecture

Typical medium- and large-scale applications separate classes and components according to architectural concern and a layered architecture define clear interfaces between these concerns. These interfaces allow localization of changes, so that changes to one concern will have minimal impact on the remaining concerns. This achieves loose coupling between layers while yielding highly encapsulated and cohesive components within each layer. A layered architecture simplifies construction, improves testability, reduces risk, and simplifies maintenance. The layering approach imposes certain dependencies, the most important of which are the following:

- Layers communicate from the top down. A layer depends only on the layer directly beneath it
- Each layer is unaware of any other layers except for the layer beneath it

This layering is reflected in the diagram of the calendar application below.

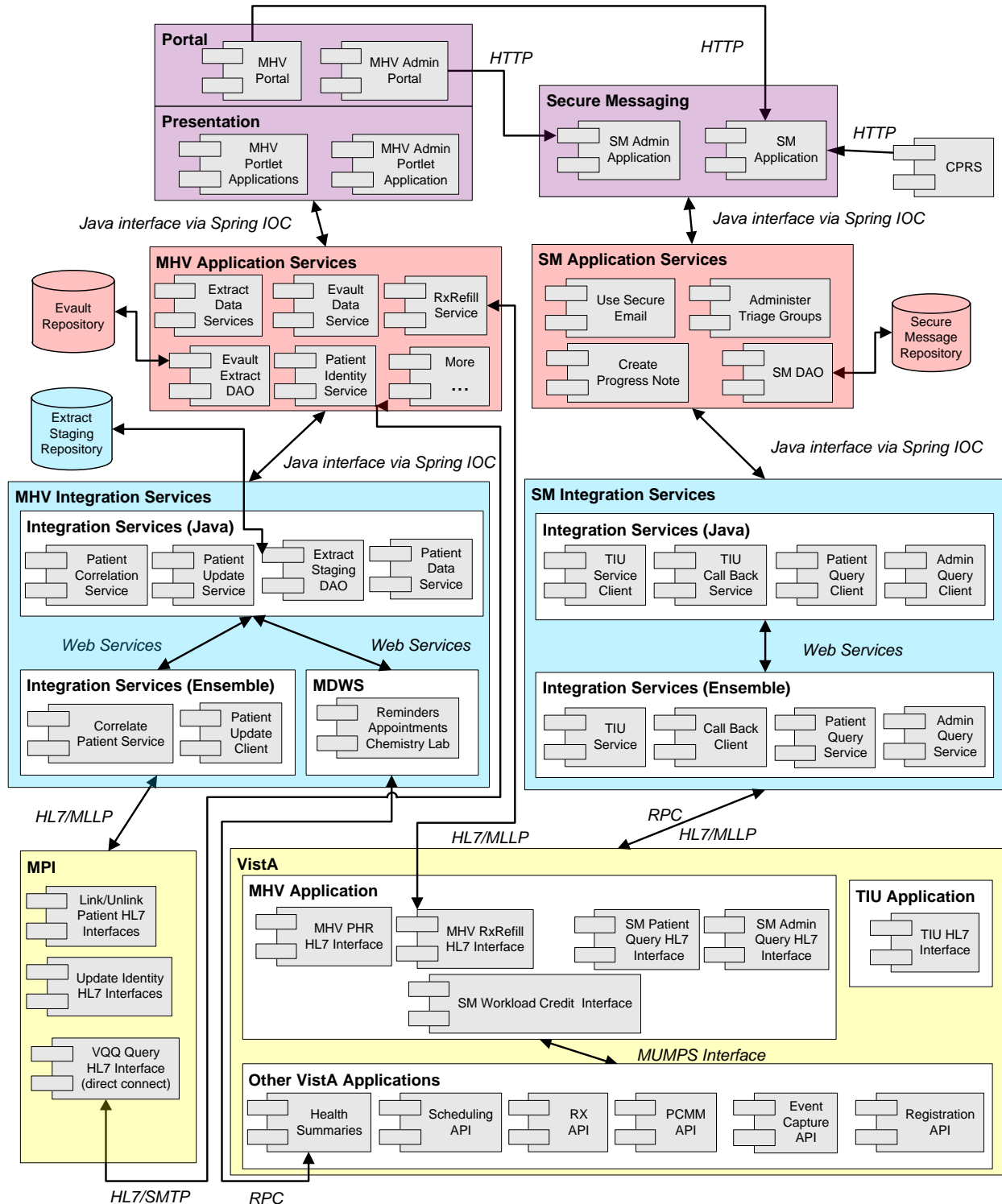
Figure 48: Technical Architectural Layering



7.2. Interface Detailed Design

Interface Detailed Design layout is displayed in the image below.

Figure 49: Interface Diagram



8. Human-Machine Interface

The capabilities enhancements were implemented on back end software systems noted in section 3.1.3 Applications Locations table. The Human-Machine Interface or GUI on the MHV national website was not changed or impacted. Therefore, users noted in section 1.9 User Characteristics Table 1: Use Case Actors are not going to be impacted by the enhancements. Only the back end administrators will be impacted by the enhancements made to the systems such as Weblogic and Oracle.

8.1. Interface Design Rules

TBD

8.2. Inputs

TBD

8.3. Outputs

TBD

8.4. Navigation Hierarchy

TBD

8.4.1. Screen [x.1]

TBD

8.4.2. Screen [x.2]

TBD

8.4.3. Screen [x.3]

TBD

9. Security and Privacy

9.1. Security

My HealtheVet (MHV) is currently going through the process of migrating from AITC to Terremark Worldwide, LLC. The projected “go live” date at Terremark is July 2014. April 22-24, 2014, a Security Control Assessment (SCA) was performed on MHV at Terremark. The SCA was performed against the System Security Plan (SSP) and all supporting documentation for MHV. The SSP was created based on the NIST 800-53 rev. 4 security controls consisting of 18 controls, plus the 8 Privacy controls, generating 354 pages. The Privacy controls were documented based on MHV being rated as FISMA HIGH due to Sensitive Personal Information (SPI) and Personally Identifiable Information (PII) existing within the system.

- The 18 control families are as follows: RA, PL, SA, CA, PM, PS, PE, CP, CM, MA, SI, MP, IR, AT, IA, AC, AU, and SC.
- The 8 Privacy control families are as follows: AP, AR, DI, DM, IP, SE, TR, and UL.

A Privacy Impact Assessment (PIA) was completed, signed, and approved by the Privacy Officer and the PIA Support team. The PIA is to be made public by the PIA support team.

All of the accreditation documentation has been completed along with the supporting documents for the SSP. The SSP, PIA, and other accreditation and supporting documents exist in RiskVision. For access to these documents, please contact the MHV ISO.

9.2. Privacy

A Privacy Impact Assessment (PIA) was completed, signed, and approved by the Privacy Officer and the PIA Support team. The PIA is to be made public by the PIA support team. The 8 Privacy control families from NIST 800-53 rev. 4 have been documented in the SSP: AP, AR, DI, DM, IP, SE, TR, and UL.

All of the accreditation documentation has been completed along with the supporting documents for the SSP. The SSP, PIA, and other accreditation and supporting documents exist in RiskVision. For access to these documents, please contact the MHV ISO.

Attachment A - Approval Signatures

This section is used to document the approval of the System Design Document. The Chair of the governing Integrated Project Team (IPT), Business Sponsor, IT Program Manager, Project Manager, and the members of the Technical and Enterprise Architectural Review Team are required to sign. Please annotate signature blocks accordingly.

Signed:
< *Integrated Project Team (IPT) Chair* >

Date:

Signed:
< *Business Sponsor* >

Date:

Signed:
< *IT Program Manager* >

Date:

Signed:
< *Project Manager* >

Date:

Signed:
< *Enterprise Architecture* >

Date:

Signed:
< *Service Delivery & Engineering* >

Date:

Additional Information

Attach any addition information that supplements the design specification.

A.1. RTM

N/A

A.2. Packaging and Installation

N/A

A.3. Design Metrics

N/A

A.4. Acronym List and Glossary

AITC

- Austin Information Technology Center, 3

CAB

- Clinical Advisory Board, 18

CCD

- Continuity of Care Document, 25

CPRS

- computerized patient record system, 1

EPG

- Engineering Process Group, 2

ESR

- Enrollment System Redesign, 3

HDR

- Health Data Repository, 3

HIPAA

- Health Insurance Portability and Accountability Act, 4

ICN

- Integration Control Number, 6

IDL

- Iterative Development Lifecycle, 8

IdM

- Identity Management, 1

IPA

- In-Person Authentication, 5

IT

- Information Technology, 17

IVMH

- Improve Veteran Mental Health, 1

LMS

- Learning Management System, 18

MDWS

- Medical Domain Web Service, 25

MHV

- My HealtheVet, 1

MPI/MVI

- Master Patient/Veteran Index, 3

MRP

- My Recovery Plan, 1

MVI

- Master Veteran Index, 5, 25

OIT

- Office of Information Technology, 1
- PEMS
- Patient Education Management System, 2
- PHR
- Personal Health Record, 6
- PMAS
- Project Management Accountability System, 2
- PTSD
- post-traumatic stress disorder, 2
- ROI
- Release of Information, 6
- RSD
- Requirements Specification Document, 8
- SCM
- software configuration management, 9
- SUD
- Substance Abuse Disorder, 1
- TRM
- Technical Reference Model, 2
- VA
- Department of Veterans Affairs, 1
- VAMC
- VA Medical Center, 6
- VSO
- Veterans Service Organization, 7

Term	Definition
AITC	Austin Information Technology Center
AOP	Aspect Oriented Programming
ARB	Architecture Review Board
BSH	Bean Shell
C&A	Certification and Accreditation
CAB	Clinical Advisory Board
CCD	Continuity of Care Document
CDCO	Corporate Data Center of Operations
CDM	conceptual data model
CHAMPUS	Civilian Health and Medical Program of the Uniformed Services
CMOR	Coordinating Master of Record
CMOR	Coordinating Master of Record
CPRS	computerized patient record system
CSP	Credential Services Provider
DAO	Data Access Object
DBMS	Database Management System
DSM IV	Diagnostic and Statistical Manual of Mental Disorders
EA	Enterprise Architecture
EPG	Engineering Process Group
ERD	Entity Relationship Diagram

Term	Definition
ESR	Enrollment System Redesign
FISMA	Federal Information Security Management Act
HDR	Health Data Repository
HIPAA	Health Insurance Portability and Accountability Act
ICN	Integration Control Number
IDL	Iterative Development Lifecycle
IDM	Identity Management
IOC	Inversion of Control
IPA	In-Person Authentication
IPT	Integrated Project Team
IT	Information Technology
IVMH	Improve Veteran Mental Health
IVMH	Improve Veteran Mental Health
IVMH	Improve Veteran Mental Health
JPF	Java Page Flow
LGPL	GNU Public License
LMS	Learning Management System
MDWS	Medical Domain Web Service
MHV	My HealtheVet
MPI/MVI	Master Patient/Veteran Index
MRP	My Recovery Plan
MRP	My Recovery Plan
MVI	Master Veteran Index
NIST	National Institute Standards and Technology
OIT	Office of Information Technology
PEMS	Patient Education Management System
PHR	Personal Health Record
PIA	Privacy Impact Assessment
PMAS	Project Management Accountability System
PMAS	Program Management Accountability System
PTSD	post-traumatic stress disorder
ROI	Release of Information
ROI	Release of Information
RSD	Requirements Specification Document
SA CMM	Software Acquisition Capability Maturity Modeling
SCM	software configuration management

Term	Definition
SDD	System Design Document
SP	Standards Profile
SSP	System Security Plan
SUD	Substance Abuse Disorder
SVG	Scalable Vector Graphics
TAR-TAS	Technical Analysis Review-Technical Analysis Summary
TRM	Technical Reference Model
TRM	Reference Model
VA	Department of Veterans Affairs
VAMC	VA Medical Center
VAPII	Veterans Authorization Preferences Interface Improvements
VISTA	Veterans Health Information Systems and Technology Architecture
WLP	WebLogic Portal
WLW	WebLogic Workshop

A.5. Required Technical Documents

The following documents must be submitted for review to support proper approval:

- Product Architecture Document
- Disaster Recovery Plan
- Interface Data Mapping
- Security Assurance Strategy

For additional information regarding how to obtain proper approval for this project, refer to the following documents:

- [IT Infrastructure Standards](#)
- [Technical Analysis Review-Technical Analysis Summary \(TAR-TAS\) process](#)
- [Enterprise Architecture Web page](#)
- [One-VA TRM](#)