
SOA Design Patterns for VistA Evolution: Web Technologies Data Sharing for VistA Evolution

Office of Technology Strategies (OTS)

Architecture, Strategy, and Design (ASD)

Office of Information and Technology (OIT)

Version 1.2

Date Issued: 15 April 2014



THIS PAGE INTENTIONALLY LEFT BLANK FOR PRINTING PURPOSES

APPROVAL COORDINATION



Date: 8 MAY 2014

Joe Paiva
Chief Technology Strategist
OIT Architecture, Strategy, and Design (ASD)



Date:

8 Jul 14

Dr. Paul Tibbits, M.D.
Deputy Chief Information Officer (DCIO)
OIT Architecture, Strategy, and Design (ASD)

REVISION HISTORY

Version Number	Date	Organization	Notes
1.0	2/19/14	ASD TS	Initial Draft
1.1	2/26/14	ASD TS	Updated draft to incorporate CTS review and feedback
1.2	4/15/14	ASD TS	Updated draft to incorporate additional feedback from ASD stakeholders

REVISION HISTORY APPROVALS

Version Number	Date	Approver	Role
1.1	2/26/14	Joseph Brooks	Enterprise SOA design patterns Government lead.
1.2	4/15/14	Joseph Brooks	Enterprise SOA design patterns Government lead.

TABLE OF CONTENTS

- 1 Introduction 1
 - 1.1 Purpose 1
 - 1.2 Scope 1
 - 1.3 Document Development and Maintenance 1
- 2 Business Need 2
- 3 Design Pattern Description for To-Be Vision 2
 - 3.1 Overarching Web Technologies Data Sharing Concept 3
 - 3.2 Technical Attributes 3
- 4 Implementation Guidelines 5
 - 4.1 Extensions 9
 - 4.2 Caching Considerations 10
 - 4.3 Auditing Considerations 11
 - 4.4 Service Governance Considerations 12
 - 4.5 Mobile Application Considerations 12
- 5 Security Considerations 13
- Appendix A: RACI Chart 15
- Appendix B: Acronyms 16

1 INTRODUCTION

1.1 Purpose

The purpose of this document is to provide the VA Enterprise Design Pattern for the development and implementation of VistA Evolution systems in a services oriented manner. Enterprise design patterns will provide the technical strategy and implementation guidance required for VA to align to best practices, standards, and guidance that constitute the OneVA Enterprise Architecture (EA).

The design patterns will be leveraged by VA to inform and constrain solution architecture. Solution architecture represents detailed product configurations and interface specifications, and guide full-lifecycle system design, integration, testing, and deployment processes for individual programs. Enterprise design patterns provide the implementation details necessary for programs to ensure that their solution architectures comply with the VA Enterprise Technology Strategic Plan (formerly the VA IT Roadmap).

1.2 Scope

This document is part of a library being developed for VA Enterprise Design Patterns, and it provides implementation guidance to expand upon the first set of design patterns completed in January 2014 for VistA Evolution COTS and non-COTS applications (Increment 1). This document elaborates on how new healthcare applications can use enterprise IT infrastructure services provided by recent VA technology investments to share data efficiently and securely across the enterprise. The VA's IT infrastructure is designed to support an environment where all future VA applications and software will be responsive in nature. More specifically, the actual design patterns will universally apply to all types of web technologies – both mobile and non-mobile.

1.3 Document Development and Maintenance

Developed collaboratively with stakeholders from OIT Product Development (PD), Office of Information Security (OIS), Architecture, Strategy and Design (ASD), and Service Delivery and Engineering (SDE), design patterns will guide and synchronize the development of system designs to drive the realization of a common vision for the enterprise. This vision, which is documented in the VA IT Roadmap, leverages best-of-breed technologies to maximize the effectiveness, efficiency and security of the VA's IT assists. This creates a feedback loop which cultivates participation and collaboration between both enterprise architecture and solution architecture domains.

This document will be reviewed and updated as needed to account for additional feedback from stakeholders as well as lessons learned from enterprise design pattern implementation. Updates will be coordinated with the Government Lead for this document, who will facilitate stakeholder coordination and subsequent re-approval.

Major updates of this document will require formal re-approval per the approval chain listed in the “Approval Coordination” section.

2 BUSINESS NEED

VA is planning for the evolution of the Veterans Information Systems and Technology Architecture (VistA) to an integrated, modern SOA environment. Recently, the VA has committed substantial resources to the development and deployment of infrastructure capabilities to enable the use of Enterprise Shared Services (ESS) and a common enterprise data store to reduce the cost and complexity of new applications. Specifically, the VA has made significant investments in two capabilities: Virtual Lifetime Electronic Record (VLER) Data Access Services (DAS) managed by ASD Product Engineering (PE) and OIT Product Development, and the VA Enterprise Messaging Infrastructure (eMI) or Enterprise Messaging Platform (eMP) (formerly Electronic Health Record (EHR) SOA Suite) managed by the Interagency Program Office (IPO). The following sections discuss implementation guidance on how to use specific products provided by VLER DAS and VA eMI. These are example enterprise capabilities that may be used by application developers to enable secure and agile data sharing, and enable a smooth transition to next generation software.

3 DESIGN PATTERN DESCRIPTION FOR TO-BE VISION

New applications which integrate into the to-be VistA Evolution SOA will be configured to be interoperable with Enterprise Shared Services (ESS). This document elaborates on how this will be accomplished with ESS provided via the VLER DAS or the SOA Suite using open standards. Using these services will facilitate re-use, achieve economies of scale, and reduce development and maintenance costs. VA defines these services in two separate categories as follows:

- Application Services
 - CRUD (Create, Read, Update, and Delete) data services (e.g. direct data access services involving CRUD operations for service consumers)
 - Composite data services (e.g. may include a composition of functions that provide data manipulation or to provide aggregate responses to service consumers from multiple data sources)
- SOA Support Infrastructure Services
 - Messaging/Enterprise Service Bus (ESB) (e.g. message exchange transport, service description and discovery, XML parsing)
 - Enterprise SOA infrastructure services (e.g. end-to-end application monitoring, authentication, authorization, auditing, event management, orchestration)

The following sections explain the overarching data sharing concept and general attributes for implementing each of the above services for new VA healthcare applications.

3.1 Overarching Web Technologies Data Sharing Concept

Figure 1 provides a conceptual overview of how an application securely accesses VistA using any type of web technology or end-user device. At a high level, the user is authenticated via an enterprise service provider for identity and access management, and upon authentication, the user accesses the application via a web server or an application access point (e.g., application server). The application leverages business logic separately from presentation (user interface) logic to obtain authorized data from web services. These services are identified at run-time using the WebSphere Registry and Repository (WSRR). The application uses SOA support infrastructure services provided by either VLER DAS or VA eMI to access VistA and provide authorized responses back to the end user.

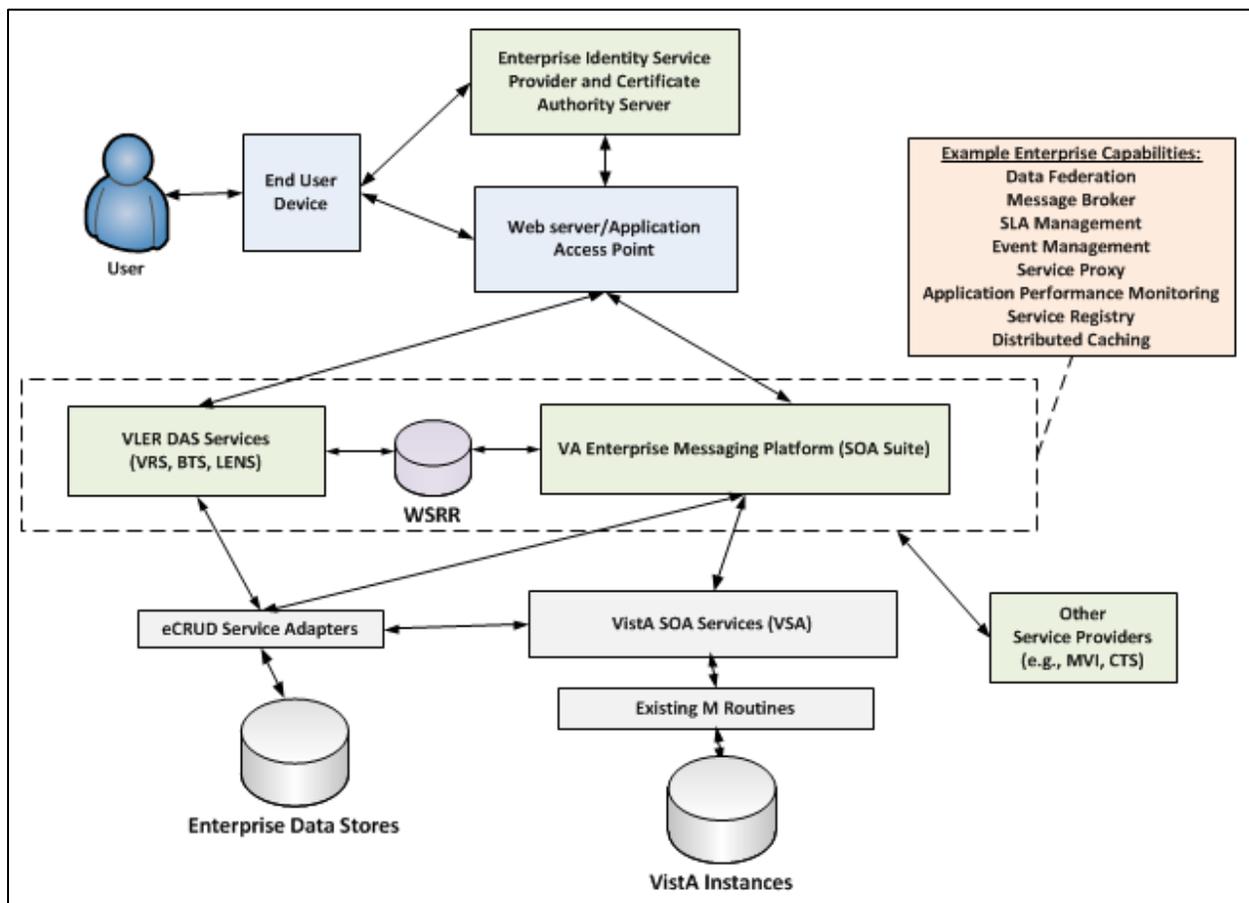


Figure 1. Overarching Concept of Web Technologies Data Sharing for VistA Evolution

As shown in Figure 1, services provided by VLER DAS and eMI will provide an abstraction or service layer that separates VA developed user-facing applications from the data layer in which enterprise data (Enterprise Data Stores or VistA Instances, per Figure 1) is stored and persisted. This abstraction enables new applications to be developed in pure HTML5 and JavaScript, using responsive design techniques that

ensure they will work on any type of end user device regardless of screen size or operating system. It also significantly reduces the complexity, time and cost of developing a new application by relieving project-level developers of the work associated with developing application specific “back-ends” by providing a single enterprise data store and reusable services by which all data is stored and retrieved. These services are integrated with many infrastructure capabilities (as shown in the orange box), such as SLA management, data federation, and end-to-end application performance monitoring, thereby relieving programs from acquiring or developing their own application-specific software for these purposes.

3.2 Technical Attributes

The primary technical attributes for implementing new healthcare applications (either via acquiring a COTS product or developing a new application internally) in alignment with the to-be IT strategic vision are outlined in the following documents:

- ***VA SOA Design Patterns for VistA Evolution - COTS Applications***
- ***VA SOA Design Patterns for VistA Evolution – Non-COTS Applications***

Programs should take into account the following implementation considerations when developing new applications that leverage enterprise application and SOA support infrastructure services provided by either the VLER DAS or eMI:

1. Use the eMI (formerly SOA Suite) to provide API management services
 - Serve as an application gateway
 - Catalog the service provider metadata
 - Match consumer request and provider based on SLA
 - Provide runtime web service governance from inception to retirement whether they are SOAP or REST
2. Use the Data Management Service (DMS) provided by the eMI for data transformations and federation
 - Federate across DoD and VA data sources
 - Aggregate results from each data source (leveraging Master Veteran Index (MVI))
 - Xforms aggregated result to a RESTful Fast Health Interoperability Resources (FHIR) canonical model
 - Provide semantic interoperability via Common Terminology Service (CTS)
 - Cache at consumer and provider locations
3. Use the VLER DAS for the following:
 - Business Transaction Service (BTS) to dynamically route messages to registered service providers
 - Read Service (VRS) for RESTful fetching of documents in the enterprise NoSQL data store

- Publish/Subscribe service via Life Event Notification Service (LENS)
 - Java EE and Node.js execution environments
 - eCRUD service adapters that facilitate an endpoint solution
 - Provide caching and a persistence layer that abstracts developers from the enterprise data persistence service (including NoSQL and SQL)
4. Both eMI and DAS will leverage VistA service adapters including those provided by the VistA Services Assembler (VSA) toolkit
 5. The eMI and DAS services provide metadata that are identified at both design time and run-time by the WSRR
 - Maps service consumers to service providers
 - Intended for all ESS in the VA

4 IMPLEMENTATION GUIDELINES

This section focuses on the use of two specific products currently acquired and deployed by the VA for implementing web technologies data sharing regardless of end-user device. This guidance discusses products that the VA already owns and is now used in the VA eMI. The following guidelines include a reference to a data layer, which is where shared enterprise data stores including the enterprise shared persistent data stores provided by the VLER DAS is deployed. This represents a single platform of shared enterprise data stores for which data can be stored and retrieved. Appendix A shows a high-level list of products discussed in this document and their respective managing organizations.

The VA eMI provides Enterprise Service Bus (ESB) capabilities via the IBM Integration Bus (IIB) that map to an Application Mediation Layer in the application's architecture and includes features such as SOAP-to-REST conversions (highlighted in red), as shown in the following concept diagram:

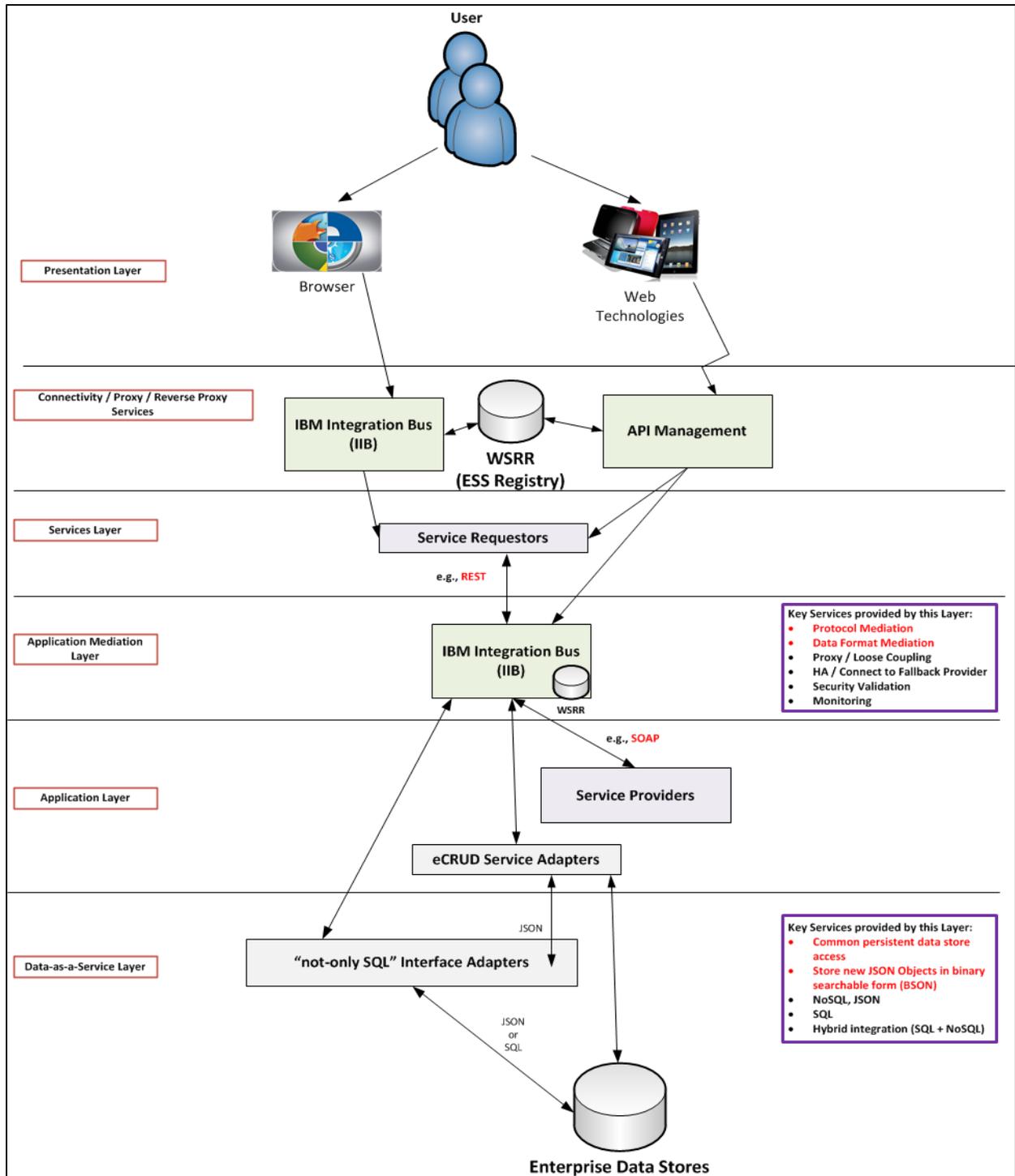


Figure 2. Use of IIB and WSRR to Provide Application Mediation in eMI

The Application Mediation Layer (IIB/WSRR) provides:

1. Proxy - Loose coupling between service requestors and providers
2. High Availability (HA) - Ability to connect to alternate service provider instance (utilizing WSRR for catalog of service endpoints and runtime endpoint selection when needed)
3. Security validation
4. Enforcement of desired monitoring and audit functions

Figure 2 shows how the IIB may be used to translate SOAP-based (highlighted in red) service provider to a RESTful response to a service requester.

One of the key functions here is the ability to provide a higher availability or Service Level so that simple application services do not need to build this into the application itself. Additional detail on this Service Level Agreement (SLA) management use case is shown in the following concept diagram:

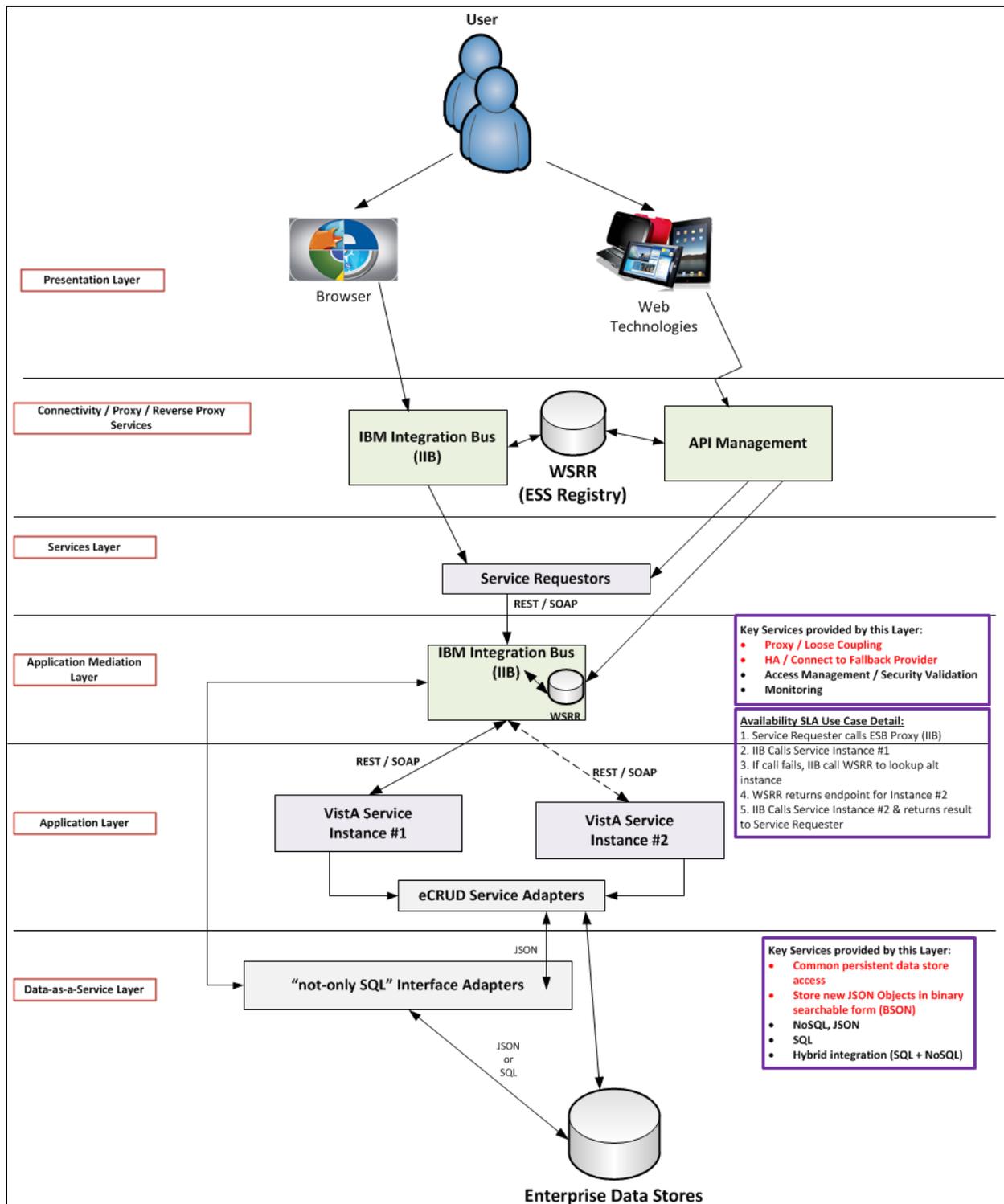


Figure 3. SLA Management Functionality Provided by SOA Support Infrastructure Services

An example use case of SLA management functionality would be the following:

1. Service Requestor (User) calls ESB Proxy using the IIB

2. IIB Calls Service Instance #1
3. If call fails, IIB calls WSRR to lookup alternate instance for the service
4. WSRR returns endpoint for Instance #2
5. IIB Calls Service Instance #2 & returns result to Service Requestor

4.1 Extensions

This general pattern can be extended to provide additional infrastructure services when needed. An example involves the mediation of protocol and/or data format between service requestors and service composite providers, and the orchestration of composite services access by federating calls to multiple application or data sources, as shown in Figure 4:

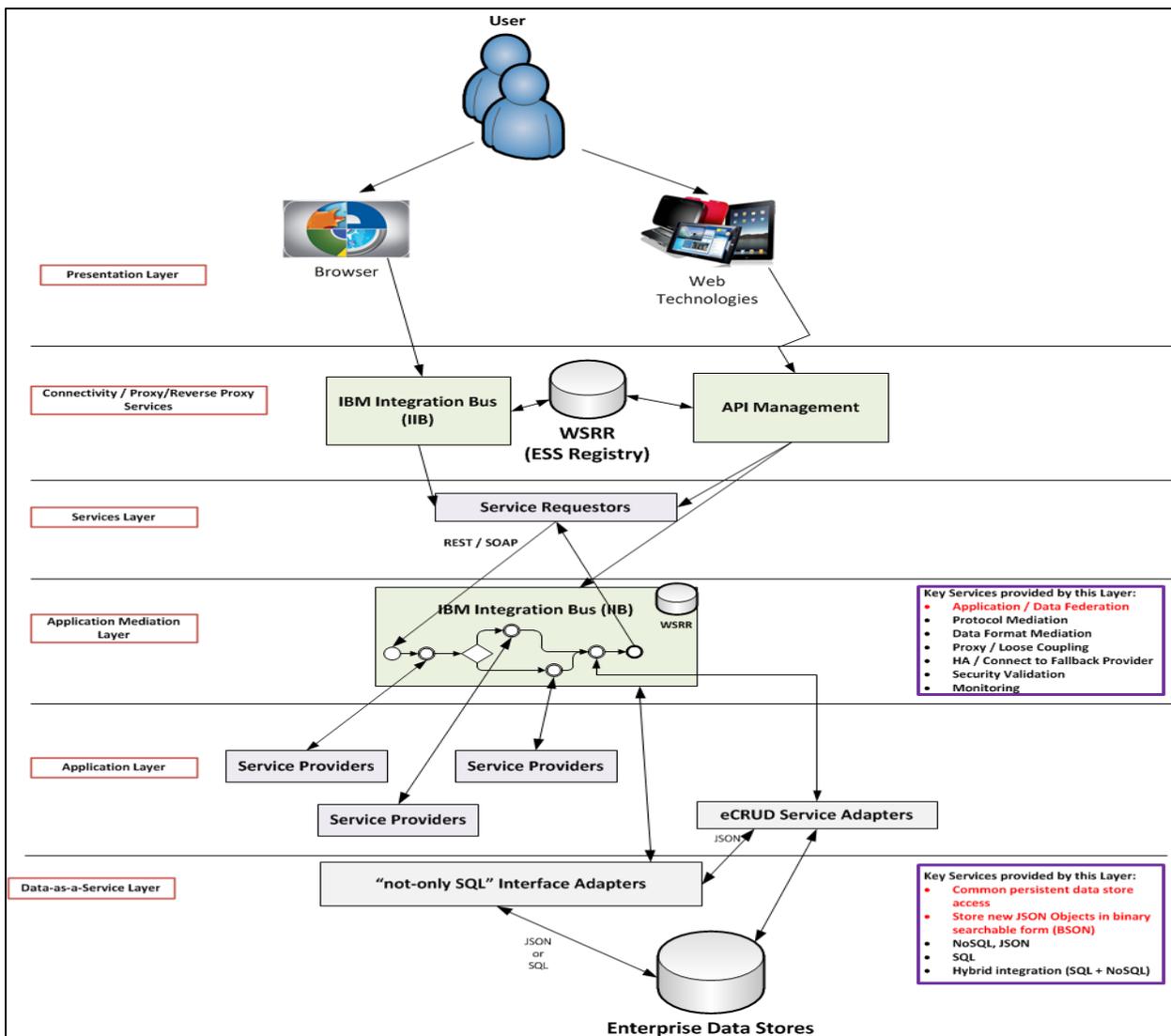


Figure 4. Internal Data Access for Stateless Composite Services

4.2 Caching Considerations

The following concept diagram shows how the eMI may be used for cross-cutting caching concerns across the architecture for service calls to reduce latency and to increase load balancing on the back-end services for future service calls.

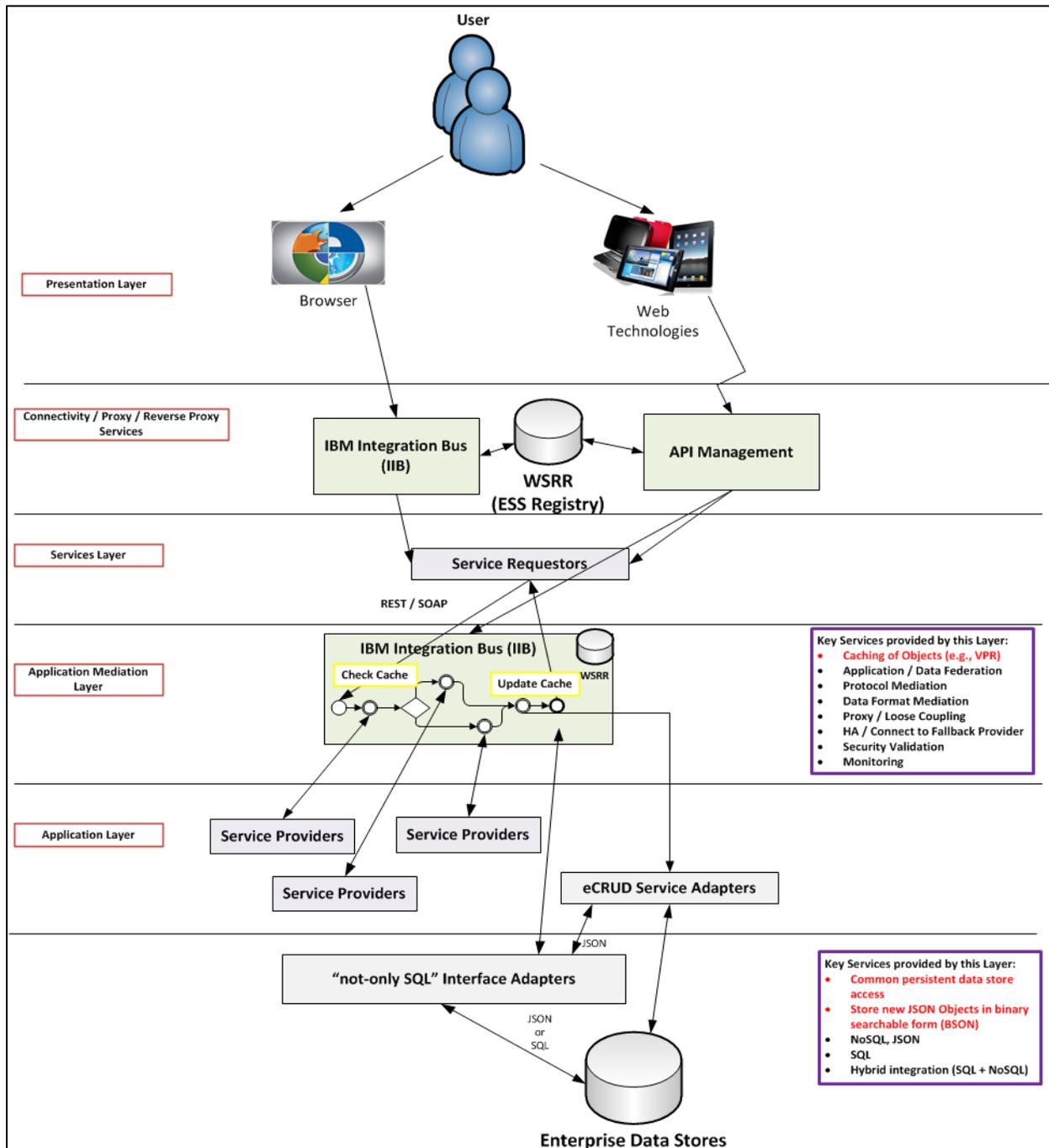


Figure 5. Internal Data Access for Composite Services with Caching

Caching considerations are as follows:

1. Caching of key (frequently re-used) data elements can occur at multiple layers of the architecture
 - o At the Connectivity / Proxy Services layer (e.g., in API Management)
 - o At the Service / Composite Service layer or Application Mediation layer (e.g., in IIB)
 - o At the data access layer (e.g., in VLER DAS or IIB)
2. Similarly, caching of key infrastructure services can occur at each layer of the architecture (e.g., Business Rules, WSRR endpoint look-ups, etc.)
3. Caching leverage either the native / "built-in" capabilities in the point products / eMI or leverage a shared caching platform (e.g., Extreme Scale / XC10)

More details regarding caching with respect to VA SOA support infrastructure services can be found in the following document:



Elaboration on where caching occurs in the

4.3 Auditing Considerations

The IBM Integration Bus (IIB) can provide several capabilities to monitor and audit services where needed. These capabilities are at the IT performance level of the services and at the application level. A functional view is provided to view the performance and operations of the services to further view performance statistics. This provides operational personnel with access to a facility to drill down to an appropriate level of detail, scoping statistics to an execution group, a thread, or even an individual node. At the application level IIB allows facilities to view the performance and usage of an application and by user. Different views can be created for applications performance monitoring, and for statistical analysis to meet monitoring and auditing needs.

4.3.1 Transaction Monitoring and Auditing

The events published by the IBM Integration Bus can be written to a transaction repository, creating an audit trail of the transactions that are processed by a broker. A transaction repository can be used for monitoring, auditing and replay of transactions. Bitstream data can be included so that failed transactions can be resubmitted. Programs can perform the following tasks to set up transaction monitoring and auditing:

1. Configure events for transactions - In most cases bitstream information is not sufficient to allow querying of the logged transactions. Key fields and other correlation data can be extracted from the message payload and placed into the event. The logging application or message flow can extract these fields and log them with the message bit stream.

2. Subscribe to the event topic and write events to a repository - Create a message flow, or any WebSphere Message Queue (MQ) application, that subscribes to the event topic and writes events to a relational database. The schema details depend on the requirements of the organization, for example the number of key fields and transaction IDs.

4.3.2 Business Process Monitoring

The events published by a broker can be monitored by WebSphere Business Monitor. Important fields in the message payload can be added to the events emitted by your message flows, allowing them to be monitored.

4.4 Service Governance Considerations

Service governance related capturing and cataloging service metadata is a key capability of WSRR in the eMI and VA Enterprise SOA Infrastructure. ASD Product Engineering's (PE) [ESS team](#) (aka Service Oriented Enterprise Center of Excellence) is working actively on this effort the IPO SOA team.

A key capability of WSRR (assuming that technical controls are in place to prohibit execution of unauthorized services) is the ability to capture desired SLAs for both service providers and service requesters (as part of the service metadata). Initially during runtime, the WSRR is used for governance and review of service contracts, and it is also used for SLA enforcement as desired.

4.5 Mobile Application Considerations

Mobile applications will adhere to the same technical attributes outlined in the previous sections. Additionally, mobile applications will leverage the VA Mobile Framework (VAMF), as documented in the **Mobile Application Reference Architecture (MARA)**. The following figure shows a conceptual overview of a user receiving authentication with enterprise identity service providers and then securely accessing the VA enterprise application store via VAMF.

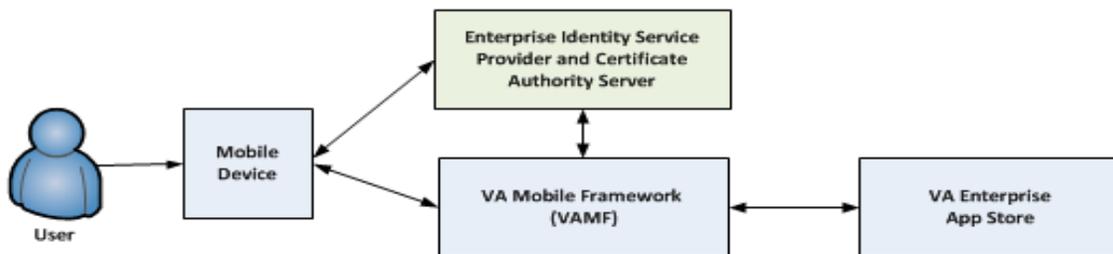


Figure 6. Conceptual Overview of Obtaining Mobile Applications via the VAMF and Enterprise App Store
Technical attributes pertaining to mobile applications are as follows:

1. Mobile devices and applications shall use end-to-end JSON message encoding and JSON data stores
2. Data encryption for mobile devices shall be FIPS 140-2 compliant using NIST certified AES 256 cryptography
3. The to-be mobile application architecture will leverage capabilities from eMI such as auditing and notification
4. Access control to mobile applications is provided by the App Store through either groups or individual identities

More details regarding mobile applications using the IBM Worklight Server (WLS) may be found in the following documents:



Draft WL

AppCenter_Feb2014.



WL and FIPS.doc

5 SECURITY CONSIDERATIONS

User authentication inside VA is aligned to security and risk management guidance provided by the Nation Institute for Standards and Technology (NIST) and applied in accordance with OMB M04-04. To determine which authentication protocol is appropriate, project managers are required to conduct a 5-step process.

1. **Conduct a risk assessment of the application/system** - using guidance published in NIST SP 800-30 or additional guidance published by VA Office of Information Security (OIS).
2. **Map identified risks to the appropriate level of assurance (LOA)** - LOA descriptions and requirements can be found within NIST SP 800-63, while guidance for mapping risks to these assurance levels is provided by OMB M-04-04
3. **Select the appropriate authentication protocol and appropriate standards for the LOA of your application/system** - Approved authentication protocols inside VA are:
 - Direct Client Authentication to the application using PKI over TLS,
 - Identity and Access Management enterprise services using IAM Single Sign-On Internal (SSOi), or
 - Active Directory authentication using Kerberos tickets.

Information and requirements for implementing these authentication protocols at the appropriate LOA are contained in the **User Authentication Design Pattern** published by VA's Office of Technology Strategies, ASD.

4. **OIS will then validate that the implemented system has met the required assurance level** - using guidance provided in NIST SP 800-53 revision 4
5. **Periodic reassessments of the system** - will take place to determine technology refresh requirements per NIST 800-37 revision 1

Detailed information on user authentication and integration with existing VA authentication mechanisms is contained in the **User Authentication Enterprise Design Pattern** and **NIST 800-63**. Additional security requirements for applications and systems are contained in **VA Handbook 6500** published by the Office of Information Security and **NIST 800-53**. The following figure shows a high-level concept of internal user authentication that will be incorporated into the **User Authentication Enterprise Design Pattern**.

Additionally, the Office of Technology Strategies has completed a review of current PIV Card implementation and implications for network security. More information about this review can be found in the following document:



ASD - Office of
Technology Strategie

APPENDIX A: RACI CHART

The primary points of contact for the eMI and VLER DAS are as follows:

eMI: Dr. Patrick Pearcy (IPO) (Patrick.Pearcy@va.gov)

VLER DAS: Steven Green (OIT PD) (Steven.Green@va.gov) and Lien Dinh (ASD PE) (Lien.Dinh@va.gov)

The following table shows a list of specific products referenced in this document and their representative organizations:

Products	VLER DAS	VA eMI
WSRR (ESS Registry)	I	R/A
BTS, LENS, and VRS	R/A	I
Enterprise Identity Service Provider (via IAM)	I	I
Enterprise eCRUD Service	R/A	I
IIB	I	R/A
WLS	I	R/A
WAS	I	R/A
DMS	I	R/A
VistA SOA Services	I	I
CTS	I	I
MVI	I	I

APPENDIX B: ACRONYMS

Acronym	Description
API	Application Programming Interface
BTS	Business Transaction Service
COTS	Commercial Off-the-shelf
GOTS	Government Off-the-shelf
FIPS	Federal Information Processing Standards
HATEOAS	Hypermedia as the Engine of Application State
HL7	Health Level Seven
HTTP	Hypertext Transport Protocol
IIB	IBM Integration Bus
JSON	JavaScript Object Notation
LENS	Life Event Notification System
LDAP	Lightweight Directory Access Protocol
MUMPS	Massachusetts General Hospital Utility Multi-Programming System
NPE	Non-person Entity
OSEHRA	Open Source Electronic Health Record Agent
PE	Person Entity
PKI	Public Key Infrastructure
RBAC	Role Based Access Control
REST	Representational State Transfer
SAML	Secure Assertion Markup Language
SLA	Service Level Agreement
SOAP	Simple Object Access Protocol
SLA	Service Level Agreement

Acronym	Description
SSL	Secure Socket Language
TLS	Transport Layer Security
VistA	Veterans Information Systems and Technology Architecture
VSA	VistA Service Assembler
WAS	WebSphere Application Server
WLS	WorkLight Server
WSRR	WebSphere Registry and Repository
VRS	VLER Read Service
XML	Extensible Markup Language