

**Identity and Access Management
Access Services 2.0 Increment 5
Single Sign-On – External
System Design Document**



Department of Veterans Affairs

March 2015

Version 1.1

Revision History

Date	Version	Description	Author
04/17/2015	1.1	Updated per anomalies	Insignia
03/27/2015	1.0	Updates for AcS Increment 5 and new template	[REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]

Artifact Rationale

The System Design Document (SDD) is a dual-use document that provides the conceptual design as well as the as-built design. This document will be updated as the product is built, to reflect the as-built product. Per the Project Management Accountability System (PMAS) Guide, the SDD as a conceptual design is required prior to the Milestone 1 Review. (Sections 1, 2, 3, 4, 5, 7, 9 need to be populated, as applicable.) The as-built design for each delivery must be incorporated prior to the Milestone 2 Review. (The entire document needs to be populated or updated, as applicable.)

Table of Contents

1. Introduction	1
1.1. Purpose of the SDD	1
1.2. Identification.....	2
1.3. Scope	2
1.4. Constraining Policies, Directives and Procedures	3
1.5. User Characteristics	5
1.5.1. Target Population.....	5
1.5.2. User Objectives	6
1.6. Relationship to Other Documents and Plans	6
1.7. Definitions, Acronyms, and Abbreviations	7
1.8. References.....	7
2. Background	8
2.1. Overview of the System.....	9
2.1.1. Federated Identity Management (FIM)	9
2.1.2. Credential Service Provider (CSP).....	9
2.1.3. Service Provider (SP) or Relying Party (RP)	10
2.1.4. Web Service Proxy	10
2.1.5. Protected Application or Resource	10
2.1.6. Federated Sign-On	10
2.1.7. Single Sign-On (SSO).....	10
2.1.8. Authentication	11
2.1.9. Authorization	11
2.1.10. Application Registration	11
2.1.11. Activation.....	11
2.1.12. OAuth 2.0	12
2.1.13. AccessVA.....	14
2.1.14. IAM Secure Token Service (STS) Web Service (WS).....	14
2.1.15. Target Portal Strategy	15
2.2. Overview of the Business Process.....	15
2.2.1. VAAFI	15
2.2.2. AccessVA.....	16
2.3. Business Benefits	17
2.4. Assumptions and Constraints	17
2.4.1. Design Assumptions.....	17
2.4.2. Design Constraints	17
2.4.3. Design Trade-offs.....	18
2.5. Overview of the Significant Requirements	20

2.5.1.	Overview of Significant Functional Requirements.....	20
2.5.2.	Overview of Functional Workload/Performance Requirements	1
2.5.2.1.	SSOe (Application Junction, Reassertion Provider, CSP/IdP Federation Partner)	1
2.5.2.2.	SSOe STS	2
2.5.2.3.	SSOe oAuth.....	2
2.5.2.4.	SSOe (WebService Client, Webservice Producer)	2
2.5.3.	Overview of Operational Requirements	3
2.5.4.	Overview of the Technical Requirements	3
2.5.5.	Overview of the Security or Privacy Requirements	5
2.5.6.	Overview of System Criticality and High Availability Requirements	6
2.5.7.	Single Sign-on Requirement	6
2.5.8.	Requirement for Use of Enterprise Portals	6
2.5.9.	Special Device Requirements	6
2.6.	Legacy System Retirement	6
3.	Conceptual Design	7
3.1.	Conceptual Application Design	7
3.1.1.	Application Context	7
3.1.1.1.	VAAFI Target Portal Strategy.....	7
3.1.1.2.	Access VA	7
3.1.1.3.	OAuth	9
3.1.2.	High-Level Application Design.....	10
3.1.3.	Application Locations	10
3.2.	Conceptual Data Design	11
3.2.1.	Project Conceptual Data Model.....	12
3.2.2.	Database Information.....	12
3.2.3.	User Interface Data Mapping	12
3.3.	Conceptual Infrastructure Design	12
3.3.1.	System Criticality and High Availability	13
3.3.1.1.	Contingency Capability and Backups.....	13
3.3.1.2.	Secure and Unsecure Access to AccessVA	13
3.3.1.3.	OAuth High Availability – Disaster Recovery (HADR)	14
3.3.2.	Special Technology.....	15
3.3.3.	Technology Locations	15
3.3.4.	Conceptual Infrastructure Diagram	16
3.3.4.1.	Location of Environments and External Interfaces	16
3.3.4.2.	Conceptual Production String Diagram	17
3.3.4.3.	VA Authentication Federation Service Provider.....	17
3.3.4.4.	VAAFI PKI CSP (SPEC192.7.11.15).....	20
3.3.4.5.	Web Service Proxy	22
3.3.5.	VAAFI Concepts	25
3.3.5.1.	Reverse Proxy	25

3.3.6.	Authentication Roles	25
3.3.6.1.	Registration and Activation	25
3.3.6.2.	User Audit and Reporting.....	26
4.	System Architecture.....	27
4.1.	High-Level Architecture.....	27
4.1.1.	System Boundaries.....	27
4.1.2.	Architectural Principles	27
4.1.3.	High-Level Network Architecture	28
4.1.3.1.	Network Layout.....	28
4.1.3.2.	VA Intranet Zone.....	30
4.1.3.3.	VAAFI Internal Zone	30
4.1.3.4.	VAAFI Demilitarized Zone (DMZ).....	31
4.1.3.5.	Internet Zone	31
4.2.	Hardware Architecture.....	32
4.2.1.	F5 Big IP Load Balancers	32
4.2.1.1.	Internal Interfaces	32
4.2.1.2.	External Interfaces.....	35
4.2.1.3.	User Interface	35
4.2.2.	DataPower.....	36
4.2.3.	ESX VMware Farm.....	36
4.2.4.	Cisco Firewalls and Switches	36
4.3.	Software Architecture.....	37
4.3.1.	AccessVA.....	37
4.3.2.	System Software Inventory	38
4.3.3.	Front-End	39
4.3.4.	Mid-Tier	39
4.3.5.	Back-End.....	40
4.4.	Network Architecture.....	40
4.5.	Service Oriented Architecture/ESS	40
4.5.1.	Protected Reverse Proxy.....	40
4.5.2.	Web Service Proxy Producer.....	40
4.5.3.	Web Service Proxy Consumer.....	40
4.5.4.	Reassertion of SAML	41
Credential Service Provider.....		41
4.5.5.....		41
4.5.6.	OAuth	41
4.5.7.	STS	41
4.6.	Enterprise Architecture	41
4.7.	Sequence Diagrams.....	41
4.7.1.	Starting from AccessVA (Level of Assurance 1 or VA PIV Credential)42	42
4.7.2.	Starting from AccessVA (Level 2+ Credential)	44

4.7.3.	Starting from the CSP	45
4.7.4.	Starting at the Agency Application.....	46
4.7.5.	Starting with a Bookmark through VAAFI to an AA	48
4.7.6.	Continuous Communication	49
4.7.7.	Insufficient Assurance Level.....	50
4.7.8.	Invalid.....	51
4.7.9.	Session Reset.....	51
4.7.10.	Protected Business Partner Web Service	53
4.7.11.	OAuth Flows	54
4.7.11.1.	OAuth Authorization Code Grant Flow	54
4.7.11.2.	OAuth Implicit Grant Flow.....	55
4.7.11.3.	OAuth Resource Owner Password Credentials Flow.....	56
4.7.11.4.	Client Credentials Grant	57
4.7.11.5.	Session Data Injection Flow	57
4.7.11.6.	Token Validation Sequence Flows	61
4.7.11.7.	Access Token Validation	63
4.7.12.	STS Sequence Diagram	64
5.	Data Design.....	66
5.1.	DBMS Files	66
5.1.1.	OAuth Physical Data Model.....	66
5.1.2.	OAuth Tables and Fields	66
5.2.	Non-DBMS Files	75
5.2.1.	VAAFI Data.....	75
5.2.2.	AccessVA Data	91
5.2.3.	PKI Registration Data.....	91
5.2.4.	IAM STS WS Data	91
5.2.4.1.	Request Security Token (RST) Message	91
5.2.4.2.	Request Security Token Response (RSTR) Message.....	92
5.2.4.3.	SAML2 Token.....	92
5.2.4.4.	JSON Token	92
5.2.4.5.	Kerberos Token	92
5.3.	Data View	93
5.3.1.	OAuth Data.....	93
5.3.1.1.	JSON Bearer Token.....	93
6.	Detailed Design.....	94
6.1.	Hardware Detailed Design	94
6.1.1.	IBM XI52 DataPower Appliance.....	94
6.2.	Software Detailed Design	94
6.2.1.	IBM Tivoli Directory Server – User Repository	96
6.2.1.1.	Software	96
6.2.1.2.	Database Design	96
6.2.1.3.	Interfaces.....	96

6.2.1.4.	External Interfaces	97
6.2.1.5.	User Interface	98
6.2.2.	IBM Tivoli Access Manager	98
6.2.2.1.	Software	98
6.2.2.2.	Dependencies.....	99
6.2.2.3.	Database Design	99
6.2.2.4.	Interfaces.....	99
6.2.2.5.	User Interface	100
6.2.2.6.	Access to Applications	101
6.2.3.	IBM WebSphere Deployment Manager	101
6.2.3.1.	Software	101
6.2.3.2.	Database Design	102
6.2.3.3.	Internal Interfaces	102
6.2.3.4.	External Interfaces	102
6.2.3.5.	User Interface	102
6.2.3.6.	Cluster Naming	103
6.2.3.7.	Node Naming	103
6.2.3.8.	FIM Domain Naming	103
6.2.4.	IBM Tivoli Session Management Server (SMS).....	104
6.2.4.1.	Software	104
6.2.4.2.	Dependencies.....	104
6.2.4.3.	Database Design	105
6.2.4.4.	Internal Interfaces	105
6.2.4.5.	External Interfaces	105
6.2.4.6.	User Interface	105
6.2.5.	IBM Tivoli Session Management Server (SMS) Catalog Server	106
6.2.5.1.	Software	107
6.2.5.2.	Dependencies.....	107
6.2.5.3.	Database Design	107
6.2.5.4.	Internal Interfaces	107
6.2.5.5.	External Interfaces	108
6.2.5.6.	User Interface	108
6.2.6.	IBM Tivoli Federated Identity Manager (ITFIM)	108
6.2.6.1.	Software	108
6.2.6.2.	Runtime Component Prerequisites	109
6.2.6.3.	Database Design	109
6.2.6.4.	Internal Interfaces	109
6.2.6.5.	External Interfaces	109
6.2.6.6.	User Interface	109
6.2.6.7.	Hardware Architecture	110
6.2.6.8.	VA_Hash and TransactionID JAR	111
6.2.6.9.	Federation Partner XSLT Files.....	111
6.2.6.9.1.	DSLogon Partner XSLT	111
6.2.6.9.2.	PKICSP Partner XSLT	112
6.2.6.9.3.	Symantec Partner XSLT	113
6.2.6.10.	Attribute Retrieval Service	113
6.2.6.11.	Portal Strategy Implementation	113
6.2.6.12.	Error Pages	120
6.2.6.13.	Logging	125

6.2.7. IBM Tivoli Access Manager WebSEAL – Reverse Proxy	125
6.2.7.1. Software	125
6.2.7.2. Dependencies.....	125
6.2.7.3. Database Design	125
6.2.7.4. External Interfaces.....	127
6.2.7.4.1. User Interface	127
6.2.7.4.2. Shared Object Space for WebSEAL.....	127
6.2.7.5. WebSEAL Instance Naming.....	128
6.2.7.6. Logging.....	128
6.2.8. PKI CSP Detailed Design	128
6.2.8.1. PKI CSP Authentication	130
6.2.8.1.1. Cluster Naming	131
6.2.8.1.2. ITFIM Domain Naming	131
6.2.8.2. PKI CSP Registration.....	131
6.2.8.3. Registration Servers	132
6.2.8.3.1. Software	132
6.2.8.3.2. Dependencies.....	132
6.2.8.3.3. Database Design	132
6.2.8.3.4. Security.....	133
6.2.8.4. IBM HTTP Server (IHS) Servers	133
6.2.8.4.1. Software	133
6.2.8.4.2. Dependencies.....	133
6.2.8.4.3. Database Design	133
6.2.9. IBM WebSphere DataPower Appliances	133
6.2.9.1. Software	135
6.2.9.2. Dependencies.....	135
6.2.9.3. Interfaces.....	136
6.2.10. AccessVA Detailed Design	136
6.2.10.1. Mobile Responsive (Bootstrap Framework)	137
6.2.10.2. Spring Framework.....	138
6.2.10.3. JavaServerPages (JSP) + Tiles.....	138
6.2.10.4. CSP Register Sequence	140
6.2.10.5. CSP Login Sequence.....	141
6.2.10.6. CSP Logout Sequence.....	142
6.2.10.7. AccessVA Widget.....	142
6.2.10.8. Third Party Onboarding Confirmation Page Sequence – Provisioning Not Available	143
6.2.10.9. E-Sig Attestation Page	146
6.2.11. OAuth Detail Design.....	149
6.2.11.1. Client Configuration Provider.....	150
6.2.11.2. Token Cache Provider.....	151
6.2.11.3. Trusted Client Manager.....	152
6.2.11.4. OAuth Web Application	153
6.2.11.5. User Session Data Storage.....	157
6.2.11.5.1. Authorization Grant Process	158
6.2.11.5.2. Implicit Grant Flow	162
6.2.11.5.3. OAuth PEP Flow.....	164
6.2.12. IAM STS WS Detailed Design	166
6.2.12.1. IAM STS WS Multi-Protocol Gateway (MPGW).....	166

6.2.12.2.	IAM STS WS Application Firewall (AFW)	166
6.2.12.3.	IAM STS WS Proxy (WSP).....	166
6.2.12.4.	VAAFI STS Web Service Process Flow.....	167
6.3.	Network Detailed Design	170
6.4.	Service Oriented Architecture/ESS Detailed Design.....	170
6.4.1.	Service Description for <Consumed Service Name>	170
6.4.2.	Service Design for <Provided Service Name>	170
6.4.2.1.	Introduction.....	170
6.4.2.1.1.	Purpose and Scope of Service.....	171
6.4.2.1.2.	Links to Other Documents.....	171
6.4.2.2.	Service Details.....	171
6.4.2.2.1.	Service Identification.....	171
6.4.2.2.2.	Service Versions.....	171
6.4.2.2.3.	Summary of Design and Platform Details.....	171
6.4.2.2.3.1.	SOA Pattern(s) Implemented.....	171
6.4.2.2.3.2.	COTS Platform Vendor Names and Versions for Hosting Platform ...	171
6.4.2.3.	Dependencies.....	171
6.4.2.4.	Service Design Details.....	171
6.4.2.4.1.	Interface Technical Specs.....	171
6.4.2.4.1.1.	Service Invocation Type	171
6.4.2.4.1.2.	Service Interface Type.....	171
6.4.2.4.1.3.	Service Name.....	171
6.4.2.4.1.4.	Interface	171
6.4.2.4.1.5.	End Points.....	172
6.4.2.4.1.6.	Operations or Methods	172
6.4.2.4.1.7.	Message Schemas	172
6.4.2.4.2.	Information Model	172
6.4.2.4.2.1.	Class Diagram and Description of Entities Involved.....	172
6.4.2.4.2.2.	Mappings from ELDM to Standards Based Schemas	172
6.4.2.4.3.	Behavior Model (AKA Use Case Realization).....	172
6.4.2.4.3.1.	Use Cases (Use Case Model)	172
6.4.2.4.3.2.	Interaction Diagrams	172
6.4.2.5.	Gap Analysis	172
6.4.2.5.1.	Variances from Enterprise Target Architecture.....	172
6.4.2.5.2.	Variances from SLDs.....	172
6.4.2.5.3.	Variances from Standards and Policies.....	172
6.4.2.5.4.	Justification for Exceptions and Mitigation.....	172
7.	External Interface Design.....	173
7.1.	Interface Architecture	173
7.2.	Interface Detailed Design	173
8.	Human-Machine Interface	180
8.1.	Interface Design Rules	180
8.1.1.	Section 508 Compliance	180
8.1.2.	Other Design Guidelines	181
8.2.	Inputs	181
8.3.	Outputs	181

8.4. Navigation Hierarchy	181
8.4.1. AccessVA Switchboard Functionality	182
8.4.1.1. AccessVA Home Page – Unauthenticated	183
8.4.1.2. Unauthenticated – Application Selected: My HealtheVet Example	183
8.4.1.3. Authenticated: DS Logon Example	185
8.4.2. AccessVA Widget.....	186
8.4.3. About Page	187
8.4.3.1. Credential Service Providers (CSPs)	187
8.4.3.1.1. DS Logon.....	188
8.4.3.1.2. DOD CAC	188
8.4.3.1.3. VA PIV	188
8.4.3.1.4. Norton Symantec LOA 2	188
8.4.3.1.5. Norton Symantec LOA 3	189
8.4.3.1.6. Connect.Gov-Basic.....	189
8.4.3.1.7. Connect.Gov-Advanced	189
8.4.3.1.8. Anonymous.....	190
8.4.3.2. VA Web Sites & Applications	190
8.4.3.2.1. ROES	190
8.4.3.2.2. SEP	191
8.4.3.2.3. My HealtheVet	191
8.4.3.2.4. VOA.....	192
8.4.4. AccessVA Login Button	193
8.4.4.1. AccessVA User Type Selector Pop-Up Widget	194
8.4.4.2. AccessVA CSP Selector User Type Pop-Up Widget.....	195
8.4.4.3. AccessVA CSP Selector Pop-Up Widget	196
8.4.5. Help & Support	197
8.4.5.1. Frequently Asked Questions (FAQs).....	197
8.4.5.2. Contact Us.....	200
8.4.6. Sign-In Partners Page	201
8.4.7. Application Preselected: MHV Example.....	203
8.4.8. Contact Us	203
8.4.9. Site Down.....	205
8.4.10. Third Party Onboarding Confirmation	205
8.4.10.1. Successful Confirmation.....	208
8.4.10.2. Unsuccessful Confirmation	209
8.4.10.3. Cancel Confirmation.....	210
8.4.11. Global Deny	211
8.4.12. VA External Link Notification Dialog	212
8.4.13. Error pages	213
8.4.13.1. VAAFI Insufficient Assurance Level Error.....	213
8.4.13.2. VAAFI Error 404.....	214
8.4.13.3. VAAFI Error 403.....	214
8.4.13.4. VAAFI Error 500.....	214
8.4.13.5. VAAFI Error 401.....	214
8.4.14. VAAFI No PKI Error	215
8.4.15. Timeout Pop-up	216

8.5.	PKI CSP Registration	217
8.5.1.	PKI Registration Screen	217
8.5.2.	401 Error Screen.....	218
8.5.3.	403 Error Screen.....	218
8.5.4.	500 Error Screen.....	219
8.5.5.	404 Error Screen.....	219
8.5.6.	No PKI Error	219
8.6.	FIM and WebSEAL Error Pages	220
8.7.	OAuth Navigation Hierarchy	221
8.7.1.	OAuth Consent Management Screen	221
8.7.2.	OAuth Response Page.....	222
8.7.3.	OAuth Error Page	223
8.7.4.	Client Registration Pages	224
8.7.4.1.	Client Registration - Main.....	224
8.7.4.2.	Client Registration – New Client Registration.....	225
8.7.5.	Device Registration Page	226
8.7.6.	Token Consent Management Page	227
9.	Security and Privacy	229
9.1.	Security	229
9.2.	Privacy	229
Appendix A. Additional Information		231
A.1	RTM	231
A.2	Packaging and Installation	231
A.3	Design Metrics.....	231
A.4	Acronym List and Glossary.....	231
A.5	Required Technical Documents.....	231
A.6	STSUniversalUser Document Schema.....	231
A.7	Attach Documents	233

1. Introduction

The VA Authentication Federation Infrastructure (VAAFI) mission is to promote and provide authentication and related services to the Department of Veterans Affairs (VA) systems and resources to better serve Veterans and those who support Veterans. VAAFI achieves this through a simplified logon and cost savings to the VA system providers.

VAAFI is different from many typical VA applications in that it is a framework built from commercial off-the-shelf (COTS) products as opposed to a custom-coded application. VAAFI provides authentication and related services to many partner applications. As the number of partners grows, the “framework” remains stable, and the configuration changes are made for the partners. Refer to the individual Integration Control Documents (ICDs) to better understand the interaction between VAAFI and its partners.

VA has also identified a need to improve externally facing Single Sign-On (SSO) support for Veterans and their representatives. Previously, Veterans would log on to multiple systems and re-authenticate to each system. Each system may or may not rely on a central identity store, and different systems may share different identity stores. Once authenticated, the Veteran may be subjected to submitting a release information form (Right of Access Request [RAR]) before gaining access to some systems.

AccessVA along with VAAFI, Enterprise Veterans Self Service (EVSS), Veterans Online Application (VOA), Stakeholder Enterprise Portal (SEP), MyHealtheVet (MHV), and other initiatives have made great strides in consolidating the VA web applications behind a single set of credentials that create a trusted SSO network.

AccessVA envisions a unified user interface for authenticating Veterans and the VA's community of interest. With SSO capability, the users will authenticate once and submit their required forms once. Upon confirmation, the results of those submissions will be securely accessible to downstream applications through web services. The purpose of this document is to outline the conceptual, logical, and physical architectures of the AccessVA 2.0.

1.1. Purpose of the SDD

The purpose of this System Design Document (SDD) is to detail the VAAFI architecture. This document differs from a traditional SDD, as VAAFI will consist of a COTS software integration and implementation, as opposed to software development. This document will describe VAAFI and its relationship and interaction with internal VA information systems and security processes as well as other systems and partners.

This document describes the VAAFI components, their relationships, and their context within Federated Identity Management (FIM). Section A, Additional Information identifies the required software for deployment. All software components of the system run on VMware ESX virtual machines running Redhat Enterprise Linux 5.9 (64 bit). The VMware ESX hosts and supporting hardware are outside of the scope of this document and maintained in production by a Platform as a Service (PaaS) contract in the Terremark Culpeper, VA, and Miami, FL, locations. The intended audience for this document includes the technical personnel working directly with VAAFI as either development or systems administration staff and personnel reviewing the VAAFI design.

This document also describes the design and implementation of an interactive gateway for authenticating Veterans and “communities of interest” with the Single Sign-On – External (SSOe) capability and is the technical response to realize the business requirements put forth by the Identity and Access Management (IAM) Business Program Management Office (BPMO) and the Access Services 2.0 Requirements Specification Document (RSD). This document is restricted to the current requirements and the approach to provide gateway functionality to stakeholders and users including Veterans, Active Duty Members, Business Partners, and Service Providers. This SDD identifies the capabilities included in the AcS Increment 5 release delivery.

1.2. Identification

The System UPI: 029-00-01-25-01-5104-00

Throughout, this document will refer to the systems as the VA Authentication Federation Infrastructure or VAAFI and AccessVA.

1.3. Scope

The scope establishes the boundaries of the design document and describes features outside of the scope (for example, services that other entities provide or that other documents describe).

Table 1: Scope Inclusions

Includes
VAAFI Service Provider (SP)
VAAFI Public Key Infrastructure (PKI) Credential Service Provider (CSP)
VAAFI Web Service Proxy (including Syslog Servers)
VAAFI Secure Token Service (STS)
VAAFI OAuth Protocol
AccessVA infrastructure and software components

Table 2: Scope Exclusion

Excludes
Monitoring System provided by the operations team (covered in the Production Operations Manual)
Logging and Reporting Systems provided by the operations team (covered in the Production Operations Manual)
Backup Requirements (covered in the Production Operations Manual)
Installation and configuration instructions for software (covered in the Installation and Configuration Guide)
Details of individual partner configurations (covered in the Interface Control Documents [ICDs])

1.4. Constraining Policies, Directives and Procedures

This design complies with the following policies, directives, and procedures (as applicable). The specific requirement and sub-requirement numbers are highlighted in the individual service-specific SDDs (where appropriate).

Table 3: Applicable Policies, Directives, and Procedures

#	Issuing Agency	Policy, Directive, or Procedure	Purpose
1	VA	VA 6500 Handbook	<ul style="list-style-type: none">• Directive Information Security Program.• Defining overall Security Framework for VA.
2	VA	VA 6501 Directive	<ul style="list-style-type: none">• VA Identity Verification In-Person Proofing (IPP) Process.• Defining overall Identity Proofing Methodology for VA IAM.
3	VA	VA 6300 Directive	<ul style="list-style-type: none">• Directive Records and Information Management.• Defines information management framework for VA Access Services.
4	NIST	SP 800-53-4	<ul style="list-style-type: none">• Special Publication – Recommended Security Controls for Federal Information Systems and Organizations.• Defines the required security controls for IT systems under the Federal Information Security Management Act (FISMA).
5	NIST	SP 800-63-2	<ul style="list-style-type: none">• Special Publication – Electronic Authentication Guideline.• Defines levels of assurance in user identities presented to IT systems over open networks.• Defines the data and procedural requirements for VA Access Services.
6	NIST	FIPS-201-2	<ul style="list-style-type: none">• Federal Information Processing Standards Publication – PIV of Federal Employees and Contractors.• Provides Identity Proofing, credentialing and chain of trust requirements and processes.• Defines the method for secure administrative interaction and control.
7	NIST	FIPS-140-2	<ul style="list-style-type: none">• Federal Information Processing Standards Publication (FIPS) – Security Requirements for Cryptographic Modules.• Defines the cryptographic standards and requirements.

#	Issuing Agency	Policy, Directive, or Procedure	Purpose
8	NIST	SP 800-122	<ul style="list-style-type: none"> • Guide to Protecting the Confidentiality of Personally Identifiable Information (PII). • Provides technical procedures for protecting PII in information systems. • Defines the information which can be used to distinguish or trace an individual's identity.
9	US Congress	Section 508 Amendment to the Rehabilitation Act of 1973	<ul style="list-style-type: none"> • Section 508 Electronic and information technology requirements for Federal departments and agencies. • Accessibility, development, procurement maintenance, or use of electronic and information technology. • Defines the “Human-Machine Interface” accessibility requirements.
10	OMB	M-04-04	<ul style="list-style-type: none"> • Memorandum to the Heads of All Department and Agencies – E-Authentication Guidance for Federal Agencies. • Defines the E-Authentication requirement.
11	OMB	M-11-11	<ul style="list-style-type: none"> • Requirements for Accepting Externally-Issued Identity Credentials. • FICAM architecture and procedures for federal agencies.
12	GSA	FICAM	<ul style="list-style-type: none"> • Federal Identity, Credentialing and Access Management (FICAM) Roadmap and Implementation Guidance. • Provides the common segment architecture and implementation guidance for federal ICAM programs.
13	White House	NSTIC	<ul style="list-style-type: none"> • National Strategy for Trusted Identities in Cyberspace (NSTIC) – Provides guidance for identity trust in cyberspace.
14	US Congress	FISMA	<ul style="list-style-type: none"> • FISMA of 2002, Public Law 107-347
15	US Congress	E-Government Act of 2002	<ul style="list-style-type: none"> • Federal Management and Promotion of Electronic Government Services. • Defines the requirements for electronic services.
16	US Congress	The Privacy Act of 1974	<ul style="list-style-type: none"> • § 552a. Records maintained on individuals. • Defines VA Access Services Privacy assessment and control requirements.

#	Issuing Agency	Policy, Directive, or Procedure	Purpose
17	National Archives and Records Administration (NARA)	Federal Records Act	<ul style="list-style-type: none"> Establishes the framework for records management programs in Federal Agencies.
18	VA	VA D 0735	<ul style="list-style-type: none"> Homeland Security Presidential Directive 12 (HSPD-12) Program Defines Department-wide policy, roles, and responsibilities for the creation and maintenance of systems and processes to implement VA's HSPD-12 Program necessary to implement Homeland Security Presidential Directive 12 (HSPD-12) program.
19	OMB	M-05-24	<ul style="list-style-type: none"> Implementation of HSPD 12 – Policy for a Common Identification Standard for Federal Employees and Contractors.

1.5. User Characteristics

1.5.1. Target Population

The target user population for VAAFI is any system user accessing a secured VA application available on the public Internet, potentially the entire Veteran population, their dependents, VA's customers and business partners, and VA employees supporting these applications. This user population has the following options:

1. Continue to use their VA applications using multiple application specific credentials (when available).
2. Obtain a credential from an approved CSP and use the single credential to access participating VAAFI applications across the VA and multiple government agencies of the appropriate assurance level.

There are exceptions to the public Internet accessibility for administrative portals and other supporting infrastructure.

Table 3: AccessVA User Characteristics

User	User Characteristic (Function)
Veteran	To enter from outside the VA firewall and access online information for self and dependents
Active Duty Member	To enter from outside the VA firewall and access online information for self and dependents

User	User Characteristic (Function)
Business Partners	A Partner to the VA to provide benefits to the Veteran, e.g., Loan Guarantee Officer (LGY)
Service Providers	A Partner to the VA to assist the Veteran with offered benefits, e.g., Veteran healthcare provider

1.5.2. User Objectives

VAAFI's overarching goal is to provide authentication services and SSO services to support the VA's Veteran-facing applications and other Veteran Relationship Management (VRM) initiatives. Authentication is the foundation for most security services including access control, auditing, digital signatures, non-repudiation, and SSO. Therefore, it is arguably the single most important security service. Authentication services support VA's architecture goal of supporting distributed information systems. Authentication services provide the following benefits:

- Reduce costs by centralizing authentication services formerly distributed to each application;
- Simplify the user's logon experience;
- Provide services essential to centralized auditing; and
- Simplify VA's security management environment.

SSO is about providing a single identity to the user for identification purposes. It allows a user to access all allowed systems with the look and feel of a single "master key," greatly simplifying the user experience. SSO is an inherent capability of a Service Oriented Architecture (SOA)-based authentication service.

VAAFI's objective has been to deploy an SOA with a flexible, modular, and scalable design that can support authentication and security services for all VA applications and initiatives.

The objective of AccessVA is to provide a single front-end to access VA applications with a variety of VA, Department of Defense (DoD), and commercially issued trusted credentials. This will not only improve the security posture at VA (due to lower number of passwords to remember for the user and fewer complex secure passwords or passphrases), but will provide a better user experience.

1.6. Relationship to Other Documents and Plans

The following plans and other documents relate to this SDD:

- Requirements Specification Document (RSD) is developed from the system's original System Requirements Specification (SRS) along with the additional requirements that led to the changes to the system over the years since the original SRS was developed.
- Contingency Plan is developed according to the VA templates and National Institute of Standards and Technology (NIST) Special Publication (SP) 800-34, which describes the processes and personnel required to operate the system when the system's primary site is not functional.
- Production Operations Manual (POM) contains the information required to successfully operate and maintain the system.

- Installation and Configuration Guide contains detailed information about how the products are installed and configured.
- Interface Control Documents (ICDs) contain information about specific interfaces with external systems.
- System Security Plan contains information about the security controls implemented for VAAFI.

1.7. Definitions, Acronyms, and Abbreviations

The abbreviations and terms used in this document are defined in the [IAM Services Master Glossary](#).

1.8. References

The references uses in this document are located on the [TSPR](#).

2. Background

VAAFI started in 2004 with the goals of simplifying access to Veteran-facing web-enabled applications and reducing costs for VA web application development and sustainment. At that time, there was a Federal initiative to create a standardized federated authentication infrastructure. VA initiated a project to select software vendors and approaches and to identify VA applications for participation. VA selected the IBM Tivoli product and began implementation early in 2006. In October 2007, VAAFI went live with Operational Research Consultants' (ORC) credential and the My HealtheVet (MHV) application.

Since deployment, the infrastructure has had the following improvements, and additional applications and CSPs were implemented:

- Addition of the Remote Order Entry System (ROES) as a protected application (February 2008)
- Addition of a warm contingency site to VAAFI located in Hines, IL. (August 2008) The existence of a warm contingency site was suspended when VAAFI moved to Terremark in March 2013, but resumed with VAAFI AcS Increment 3.
- Addition of the electronic Contract Management System (eCMS) as a protected application (August 2008)
- Migration from a Microsoft Windows-based platform for running virtual servers to a more stable and scalable product, VMware ESX in Production (May 2009)
- Improved report and auditing functionality using IBM's Common Audit and Reporting Service (CARS) (May 2009)
- Improved monitoring and testing capability (May 2009)
- Acceptance of the Common Access Card (CAC) or Personal Identity Verification (PIV) cards as identity credentials (July 2009)
- Acceptance of DS Logon as an identity credential (September 2009)
- Potential to use a second factor in authenticating to protected resources (September 2009)
- Migration to VMware ESX at the Contingency Site (October 2009)
- Addition of eBenefits as a protected application (October 2009)
- Implementation of additional high availability and scalability features (January 2010)
- Addition of Web Service proxy capabilities (January 2010)
- Web Service session check (May 2010)
- SAML Reassertion to external partners (Tricare Online [TOL]), Symplicity, and Prudential) (August 2010)
- Retirement of the ORC Credential, with the possibility of replacement by one or more of several credentials including: Symantec, United States Automobile Association (USAA), or the VA CSP (September 2012)
- Upgrade of Tivoli Products to newer versions and removal of CARS integration and Error Handling Servlet (as part of the move to the Terremark Facility in Culpeper, VA (March 2013)
- Integration with the Norton Symantec credential (January 2013)

- Migration of EAuth to AccessVA with additional functionality added to AccessVA (April 2014)
- Implementation of Target Portal Strategy Traits (July 2014)
- Standing up the Terremark Miami Florida site as the primary site and keeping the Culpeper, Virginia, site as a warm failover site (September 2014)
- Implementation of Third-Party Onboarding using a soft-boarding approach for DS Logon and CAC users (December 2014)
- Addition of an AccessVA and e-Sig Widget (January 2015)

AccessVA facilitates externally facing SSO support for Veterans and their beneficiaries. This prevents the need for Veterans to log on to multiple systems and re-authenticate to each system with different credentials. Each system may or may not rely on a central identity store, and different systems may share different identity stores. Once the Veteran authenticates, he or she may be subjected to submitting a RAR before gaining access. AccessVA's design makes it scalable with dynamic HTML page generation to support the addition of new VA applications as well as newer CSP to support initiatives such as NSTIC.

2.1. Overview of the System

This section provides descriptions of the key concepts of a Federated Identity Management (FIM) infrastructure and its relation to VAAFI.

2.1.1. Federated Identity Management (FIM)

Identity management in general is the process of establishing, verifying, maintaining, and decommissioning user credentials for logging onto systems usually within a single enterprise. Throughout this entire life cycle, implementing security mechanisms for establishing, storing, and destroying identity data must be commensurate with the risk allowable within the system.

A FIM infrastructure allows individuals to use the same username, password, or other personal identification to sign on to the networks of more than one system. Partners in a FIM system depend on each other to authenticate their respective users and vouch for their access to services. For example, this allows a Veteran to order a prescription drug from a participating VA application, order prosthesis supplies from a second participating VA application, as well as conduct various transactions with other participating federal agencies such as the DoD without using different login usernames and passwords for each application.

FIM includes the set of business agreements, technical agreements, and policies that enable government agencies and their business partners to improve user experience and mitigate security risks for transactions.

2.1.2. Credential Service Provider (CSP)

A CSP is an entity that provides a credential and that authenticates the end users of the system. Authentication often occurs through comparison of a user identifier and password that the user provides to a secured database of user identifiers and passwords. Other cases use cryptographic mechanisms or physical tokens to make the process more secure. These user identifiers, passwords, cryptographic mechanisms, and tokens are called credentials. A CSP may provide these credentials themselves or delegate this responsibility to a Registration Authority. A CSP

must follow specific rules for the issuance, maintenance, and secure transmission of information related to user credentials. VAAFI bases these rules on NIST SP 800-63 and OMB M-04-04.

In VAAFI, a CSP sends a SAML assertion identifying the authenticated user and assurance level to a SAML receiver, in this case the VAAFI Service Provider (SP). Service Providers are also sometimes called Relying Parties (RP) because they rely on an identity a CSP sends.

VAAFI contains a CSP called the PKI CSP. This CSP uses a PIV or CAC as a credential, forcing the user to authenticate the card by entering the card personal identification number (PIN), then checking the validity of the card certificate.

2.1.3. Service Provider (SP) or Relying Party (RP)

An SP is the role in a federated architecture that receives an authenticated identity and its assurance level for the organization. That organization then either uses the identity or securely passes the identity to a protected application or resource.

In VAAFI, the VAAFI SP receives SAML and passes the user's identity and other attributes, such as the assurance level, to the protected resources in secure HTTP headers.

2.1.4. Web Service Proxy

A web service proxy acts as an intermediate between a caller application and the target web service. Web service proxying is a very common practice used for different reasons, like security or auditing. In VAAFI, the web service proxy reduces the burden on partners for maintaining certificates between consumers and producers, and can perform additional steps, such as checking for user sessions.

2.1.5. Protected Application or Resource

The protected application or resource is the ultimate consumer of the identity. This document refers to them as Agency Applications (AAs). In VAAFI, the AAs receive the identity in secure HTTP headers.

2.1.6. Federated Sign-On

Federated Sign-On is the ability of end users to authenticate to multiple resources or applications spread across different domains, different organizations, and even different federal agencies with only one authentication credential. Depending on the trust relationships established, and the associated security levels of the systems being accessed, the user could be asked to present the single credential again.

2.1.7. Single Sign-On (SSO)

SSO provides users with the ability to authenticate (log on) once and be able to access resources or applications that they are entitled to, across the enterprise without re-authenticating to each system. This is different from Federated Sign-On where an individual may need to authenticate more than once (although, with the significant convenience of having to remember only one set of credentials). True SSO requires the establishment of significant trust relationships and specific security requirements for passing the user seamlessly from applications of varying security levels without having to re-authenticate.

The VAAFI project will implement Federated Sign-On and, in most cases, SSO; however, SSO depends on cooperation of the application owners to post links that implement SSO.

2.1.8. Authentication

Authentication is the process of validating a presented identity. Several types of objects, including an end user, a specific device, or a specific computer process, can present an identity. The methodology for authenticating the identity of these diverse types of objects can be very different.

One of the CSPs authenticates the end users participating in VAAFI. Currently, these include the following:

- Defense Manpower Data Center (DMDC) DoD Defense Self-Service Logon (DS Logon) credential
- VAAFI PKI CSP, for using the DoD CAC and VA PIV card
- Norton Symantec credential
- Connect.Gov

2.1.9. Authorization

Authorization is the mechanism by which a system determines the level of access that a particular authenticated user should possess to secure resources controlled by the system policies. Although the CSPs authenticate the end users, AAs will continue to perform authorization.

While VAAFI software does have additional capabilities in the area of end user authorization/provisioning, these capabilities are not currently within the scope of the VAAFI project; however, this functionality could provide significant benefit and cost savings to other VA projects or initiatives.

2.1.10. Application Registration

Application registration is the process of end users requesting and obtaining initial access to an agency application. Once an end user has been granted access to an application, the “registration” process is complete. While the CSPs provide a credential to the end users, the VA application registration process for VA end users participating in VAAFI remains the responsibility of the AA owners.

2.1.11. Activation

Activation is the process that binds the identity of a VAAFI user with the same user’s identity stored in the AAs (i.e., John F. Smith from DS Logon is Johnny F. Smith in the application). Activation can occur in many different ways to meet the requirements of applications. The activation mechanisms can range from fully automated to a completely manual process. The common types of activation include the following:

- None: Activation is not required. For example, there may not be users stored in the target application.

- **Deferred:** This approach requires a manual activation. This could be a manual or systematic check, or an out-of-band communication requiring the user to perform some specified action. The expectation is that AAs will use this mechanism.
- **Prompted activation:** This involves mapping the asserted identity to a specific end user in the AA's user database. Performing this can be by the contents of the SAML assertion, PKI certificate, or additional information the AA receives prompting the end user to enter some form of shared data. This is currently the preferred mechanism of activation in VAAFI.
- **Automatic:** This involves mapping the asserted identity to a specific end user in the AA's user database, solely from the contents of the SAML assertion or PKI certificate as they are passed on in the HTTPS headers.

2.1.12. OAuth 2.0

OAuth 2.0 (henceforth referred to as OAuth) is an authorization framework that provides a mechanism for clients to access resources on behalf of a resource owner. It also enables a resource owner to, explicitly or implicitly, provide consent (permit/deny) for accessing resources, known as a three legged flow. By introducing an authorization layer, OAuth alleviates the need for resource owners to share their credentials with the clients. OAuth defines four roles as part of the protocol flow:

- **Resource Owner:** Entity capable of providing access to a protected resource. The resource owner can be system or a user.
- **Resource Server:** Entity responsible for protecting resources and authorizing access to them. Resource Servers play the role of the Policy Enforcement Point.
- **Client:** An application accessing resources on behalf of the resource owner. OAuth 2.0 specification defines two types of clients:
 - **Client:** Clients incapable of maintaining the confidentiality of their credentials (e.g., clients running on devices)
 - **Confidential Client:** Clients capable of maintaining the confidentiality of their credentials (e.g., server side web application). These clients are suitable for the Authorization Code grant.
- **Authorization Server:** Entity responsible for issuing authorization codes and access tokens.

OAuth defines a number of methodologies for clients to access resources on behalf of the resource owners. The diagram below illustrates a high-level flow between the OAuth 2.0 protocol entities.

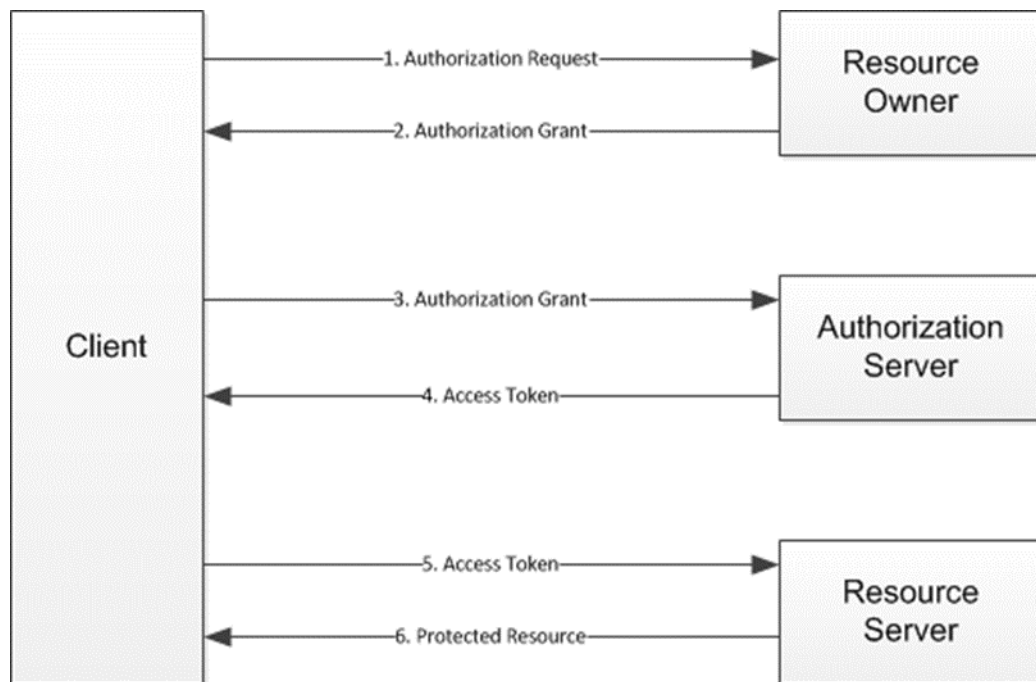


Figure 1: OAuth High-Level Flow

The OAuth 2.0 support in Tivoli Federated Identity Manager provides the following four different ways for an OAuth client to obtain an Authorization Grant:

- Authorization Code: The authorization code grant is used to obtain both access token and refresh tokens and is optimized for confidential clients.
- Implicit Grant
- Client Credentials Grant
- Resource Owner Password Credentials Grant

The OAuth 2.0 function of the Tivoli Federated Identity Manager can be configured through the following methods:

- Tivoli Federated Identity Manager console
- Command-line interface

Additionally, to conform to additional services being developed under the SSOi IAM OAuth solution an IBM DataPower OAuth implementation will be included as an Authorization Server Option. This enhancement will streamline Client Management at the DataPower level and reduce the number of custom administration screens and custom code in general for IAM OAuth services. The backend DB2 solution that has already been developed will remain in place for these enhanced services, and for the short term the TFIM administration screens will remain in place but the end goal will be to utilize the DataPower Administration screens for OAuth management in IAM (SSOe/SSOi).

Still Under Development:

- OAuth High-Level Flow for SSOe and SSOi combined
- Architectural Diagram of SSOe TFIM and DataPower Hybrid Solution

- Summary of DataPower enhanced OAuth Capability for SSOe
- Transition plan for any current OAuth enablements based on final solution
- JSON Process Flow and Token Types
- Sprint 2 for Design
- Sprint 4 for Implementation

2.1.13. AccessVA

Primarily, the AccessVA subsystem aims to be a single point of entry for the various applications within the VA Enterprise. AccessVA supports unauthenticated as well as authenticated users at various authentication levels of assurance. For unauthenticated users, AccessVA provides information about the ability to obtain credentials from the various CSPs and provides links to register and obtain credentials. For authenticated users, AccessVA provides access to various VA applications integrated to the AccessVA system. This subsystem is accessible by the standard AccessVA website or through a widget version that can be placed on a consumer's application.

Secondarily, AccessVA provides a web platform for end-user facing VAAFI features and web pages. These include Third Party Onboarding pages such as the Confirmation page, Success page and Failure page, and Error pages for the VAAFI SP and PKI CSP. The AccessVA Widget is a compact web page that presents the login sources for users rather than accessing the AccessVA page. This allows users to seamlessly access AccessVA within an iFrame from a different application.

2.1.14. IAM Secure Token Service (STS) Web Service (WS)

The IAM STS WS is the service that builds, signs, and issues security tokens according to the Web Service (WS-*) collection of standards, to include WS-Trust, WS-Policy, and WS-Security.

The IAM STS WS acts as an authorization broker between consumer (client or client application) and a producer. In lieu of generic or service account credentials, the consuming application utilizes security tokens or active SSO sessions to request additional security tokens (or claims) from the IAM STS WS. The consuming application may then use the security tokens to authorize with producing applications. The producing application uses these tokens to make authorization decisions, and return appropriate business data to the consuming application.

The service will be multiprotocol and multitoken, allowing the greatest amount of flexibility in the requesting and issuance of tokens.

Figure 2 is the high-level STS relationship diagram depicting a consumer and producer using an STS token service – To be updated during Sprint 6.

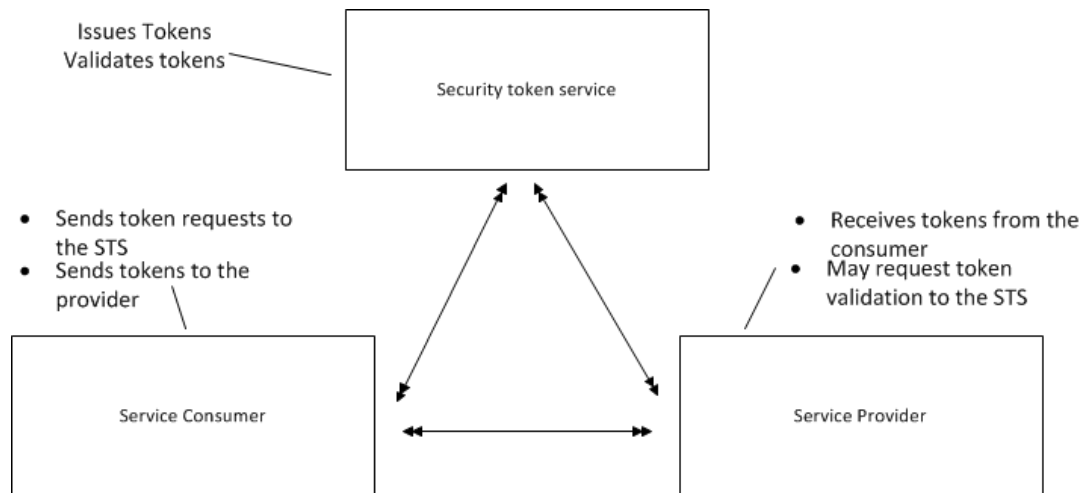


Figure 2: Authentication Using an STS

2.1.15. Target Portal Strategy

The Target Portal Strategy is an effort to send consistent data to VAAFI partners, break free of the data restrictions and CSPs, and become “CSP agnostic.”

Highlights of this architecture are:

- IAM provides all basic person data and certain primary IDs, including Integration Control Number (ICN), Person Identification (PID), File Number (BIRLS), My HealtheVet (MHV) individual entry number (IEN), Security Identification (SecID), and Electronic Data Interchange Person Identification (EDIPI), during authentication.
- SecID is retrieved from Provisioning Store based on CSP ID.
- Basic person data and Primary IDs are retrieved from Master Veteran Index (MVI) via VDS based on SecID.
- Portal receives HTTP headers containing standardized Authentication Traits via VAAFI.
- Portals, portlets, and portal-based applications may call MVI for additional traits and IDs (depending on their business needs).
- All portals are integrated with AccessVA.

2.2. Overview of the Business Process

2.2.1. VAAFI

VAAFI consists of five basic business flows as shown in the table below.

Table 4: Business Process

Business Process ID	Business Process Name	Type	Owner	Description
1	Authenticate to System	Existing	VA Application or System	Proxy user authentication to a partner application or system

Business Process ID	Business Process Name	Type	Owner	Description
2	Proxy Web Service	Existing	VA Application or System	Proxy web service producers and consumers
3	Enterprise Identity Resolution	Existing	VA Application or System	Search for user enterprise identity, combine enterprise view and CSP view, initiate registration confirmation when required
4	3rd Party Authorization	Existing	VA Application or System	Provides a mechanism for clients to access resources on behalf of a resource owner
5	Extended Authentication	Existing	VA Application or System	Builds, signs, and issues security tokens according to the WS-Trust, WS-Policy, and WS-Security

2.2.2. AccessVA

AccessVA consists of six business processes, as Table 5 shows.

Table 5: AccessVA Business Processes

Business Process ID	Business Process Name	Owner	Description
AccessVA-001	AccessVA Public Landing Page for Users	IAM Integrated Product Team (IPT)	Navigation to AccessVA, logging into DS Logon
AccessVA - 002	New Users Registers through Application	IAM IPT	Logging in through an application
AccessVA - 003	Register for Credential at AccessVA	IAM IPT	Navigation to AccessVA; registering for a CSP from a matrix of CSPs to applications displayed and accessing the DS Logon self-registration page
AccessVA - 004	Log on to AccessVA	IAM IPT	Navigation to AccessVA and log on
AccessVA - 005	User logs into AccessVA through Applications	IAM IPT	Navigation to applications to login to AccessVA
AccessVA - 006	Register in IAM Identity Stores	IAM IPT	Completion of confirmation page an submission to AcS services

2.3. Business Benefits

VAAFI provides numerous business benefits; the following list is by no means all inclusive:

1. Reduces the number of usernames and passwords required to access VA Web sites
2. Removes the need for VA web sites and systems to implement authentication and support usernames and passwords
3. Consolidates the number of partner web service producers and consumers VAAFI Partners need to manage

2.4. Assumptions and Constraints

This section describes the assumptions, and constraints that impacted the design of the system.

2.4.1. Design Assumptions

- VAAFI has a High System Baseline for Confidentiality, Integrity, and Availability.
- VAAFI initial system size objective is to accommodate millions of users.
- VAAFI will run on Terremark-provided VMware ESX platforms with sufficient resources.
- Tivoli Federated Identity Management Suite and other system utilities will run on RedHat Enterprise Linux 5.X and 6.X (64-bit).
- VAAFI will adhere to NIST SP 800-63-2: Electronic Authentication Guideline and OMB M0404: e-Authentication Guidance for Federal Agencies.
- Protected applications accept the authentication standards (and any associated risk) of the CSPs from whom they receive credentials (password complexity, etc.).
- Protected applications provide authorization of users for content appropriate for the users' credential assurance level based on NIST SP 800-63 and OMB M04-04.
- VAAFI supports SAML version 1.0, 1.1, and 2.0. and JSON Web Tokens.
- AccessVA will be deployed as a component of the Development, Software Quality Assurance (SQA), Pre-production, and Production Infrastructure that VAAFI is provided.

2.4.2. Design Constraints

This section details any constraints on the system design, such as schedules or costs, or technical constraints, such as the company's commitment to a specific development platform or programming language.

- Licensing: A limited number of IBM Tivoli Federated Identity Manager (ITFIM) licenses are available.
- System Resources: VAAFI and its AccessVA component run on the IAM-shared platform and are limited by the resources available and resources consumed by other IAM systems.
- Operating Systems: VAAFI must use VA-approved operating systems.

- Downtime with partners: During any transition, upgrade, or system maintenance, downtime needs to be limited as partners rely on VAAFI for authentication and related services “24/7/365.”
- Connectivity: Upgrade and migrations may require Enterprise Security Change Control Board (ESCCB) approvals before being able to establish connections

2.4.3. Design Trade-offs

VAAFI development incorporates several design trade-offs. This section details the following trade-offs:

- **Centralized vs. Decentralized Security Assertion Markup Language (SAML) Implementation**

In the decentralized approach, every application interfaces with the Credential Service Providers (CSPs), whereas in the relying party approach a centralized service provides the interface to the CSPs on behalf of all of the VA applications. A centralized implementation was chosen and thus VAAFI acts as the relying party for all VA applications.

- **SP Implementation: Infrastructure vs. Agent**

One of the VAAFI Project’s earliest decision points was the determination of the primary architecture methodology for VA applications to interface with the other VAAFI members.

One approach could have been to install agents on protected applications. This approach requires significant modification to the application and long-term management of all the trust, policy verification, and traffic authorizations directly by each individual application. In effect, each of the protected applications would need to catch SAML and be its own Service Provider.

An enterprise infrastructure approach allows a centralized internal infrastructure to perform as an authentication layer between the CSPs and the protected applications. This removes the responsibility of performing many of the trust, policy verification, and authentication actions from the participating applications. It also minimizes the level of software modification necessary for applications to participate. The infrastructure methodology also provides a standardized approach for applications to participate in VAAFI. The VAAFI project elected to move forward with the infrastructure approach.

- **PKI CSP: Direct Lightweight Directory Access Protocol (LDAP) Entry vs. SAML**

A decision was that VA partners would authenticate via SAML assertions instead of authenticating directly into Tivoli Access Manager (TAM), the SP. This promotes the federated model, allowing VAAFI to reassert these users to applications outside of VAAFI.

- **Encrypted Headers vs. SAML (internal applications)**

A decision was to pass the user authentication attributes via encrypted HTTP headers as opposed to passing the information along via the SAML assertion. This allows the application to integrate with VAAFI without needing to understand SAML and ensure ease of integration.

- **SAML vs. Encrypted Headers (external applications)**

A decision was to pass the user authentication attributes via SAML assertions as opposed to passing the information along via the encrypted headers. This ensures a greater level of security for applications residing outside the VA network and standards compliance.

- **AccessVA Flexibility vs. Integration Complexity**

No.	High-Level Requirement
1.	Configure and develop AccessVA functionality which includes: <ul style="list-style-type: none"> • AccessVA User Experience enhancements consisting of the following: <ul style="list-style-type: none"> ○ Unauthenticated Home Page ○ Processing Page ○ CSP Selection Page ○ Authenticated Page ○ Frequently Asked Questions (FAQ) Page ○ Help Page ○ Login Button ○ User Selector Pop-Up Widget • Develop enhancements to iFrame capability that supports embedding one HTML document within another • Develop session based (User Persona Context) role selection capability • Capability to exchange imprecise dates of birth information with Master Veteran Index (MVI) • Capability to add Address information to Virtual Directory Service (VDS) • Develop minimum Level of Assurance (LOA) 1 trait set
2.	Configure and develop a SSOe capability which includes: <ul style="list-style-type: none"> • Capability to exchange imprecise dates of birth information with Master Veteran Index (MVI) • Capability to add Virtual Directory Service (VDS) Address data to Portal Strategy header fields • Capability to add Role object to Portal Strategy header fields • Develop session based (User Persona Context) role selection capability • Develop minimum Level of Assurance (LOA) 1 trait set
3.	Configure and develop enhancements to the SSOe STS functionality (Phase 1) to include: <ul style="list-style-type: none"> • Exposing STS Service with REST interface • Accepting JSON Web Tokens and Bearer Tokens
4.	Configure and develop enhancements to enterprise capability that supports OAuth protocols and Mobile Access to include: <ul style="list-style-type: none"> • Mobile device registration. • OAuth Access Token Validation service

5.	Develop capability to provide PID and BIRLS ids
6.	Enhance SSOe activity to comply with performance specifications as detailed in RSD

The architecture of the AccessVA component is flexible and uses the Java 2 Platform, Enterprise Edition (J2EE) best practices. The AccessVA architecture is scalable with dynamic HTML pages to support the addition of new VA applications as well as new CSPs supporting initiatives such as the National Strategy for Trusted Identities in Cyberspace (NSTIC).

- **Flexibility vs. Integration Complexity**

The system has flexibility to add new CSPs; however, the complexity of attribute mapping and required attributes increases the complexity of integration.

2.5. Overview of the Significant Requirements

This section includes an overview of significant requirements. This version of the SDD meets the following high-level requirements:

2.5.1. Overview of Significant Functional Requirements

This section provides an overview of the major functional requirements for the system. This section does not include the full set of requirements or replace the functional requirements documents, but rather identifies the major functions to be performed and the few major requirements that drive the design that is described in the later sections. The emphasis is on identifying the impact that the requirements have on the design.

The principal AccessVA usage scenarios include the following:

1. The user accesses AccessVA from an application
2. The user accesses AccessVA first, selects an application, and signs on using the credential of choice and is directed immediately to the application if the user is known in VDS, otherwise the user is directed to the 3POB Confirmation screen to complete onboarding
3. The user is presented with a widget to authenticate through VAAFI

AccessVA will provide a dynamic user interface. For example, if an application only accepts credentials from two CSPs, then AccessVA will display only sign-on widgets for those two applications. If the application accepts three credentials, three widgets display on AccessVA, and so on.

The “User logs into Application through AccessVA” maps to the “AccessVA Sign-On” use case as follows:

1. Sign on to a CSP with a particular application as the target.
2. Upon Successful authentication, redirect to the application.

Table 6: Functional Requirements

ID	Specific Requirement/Synopsis	Requirement
VAAFI-1	Standard Framework	VAAFI shall provide authentication services for VA applications through a standard framework.
VAAFI-2	COTS Product	VAAFI shall use commercial off-the-shelf (COTS) Federated Identity Management (FIM) products.
VAAFI-3	Open Standards	VAAFI shall support a suite-based and open-standards approach to identity and access management.
VAAFI-4	Uniquely Identify End Users	VAAFI shall provide a standard method of uniquely identifying end users and presenting them to agency applications.
VAAFI-5	Session Timeouts	VAAFI shall be capable of configuring session timeout that is compatible with the individual agency applications.
VAAFI-6	Trusted CSPs	VAAFI shall determine that the communication is with a trusted CSP.
VAAFI-7	CAC and PIV Acceptance	VAAFI shall accept CAC and PIV cards for authentication
VAAFI-8	Error Handling	VAAFI shall appropriately handle malformed or missing information received from partners
VAAFI-9	Monthly Reporting	VAAFI shall provide data elements needed to provide a monthly report containing each authenticated session. Log the following: Timestamp, target app, csid, uid, assurancelevel, assertion_id/artifact
VAAFI-10	Web Content Standards	VAAFI shall conform to VA standards for web content.
VAAFI-11	FAQ Link	The portal shall have a VA FAQ link
VAAFI-12	Descriptive Errors	VAAFI shall provide a descriptive error message to the user when an error occurs.
VAAFI-13	Unavailable Application Message	VAAFI shall provide a user-friendly message in the event the application is unavailable/unresponsive.

ID	Specific Requirement/Synopsis	Requirement
VAAFI-14	Usage Scenarios	<p>VAAFI shall be able to accommodate any of the following usage scenarios:</p> <ul style="list-style-type: none"> • New User to both VAAFI and agency application • New User to VAAFI, existing user in agency application • Existing User to both VAAFI and agency application • Existing user to VAAFI, new user to agency application.
VAAFI-15	Multiple Credentialed Users	VAAFI shall be able to accommodate users with multiple credentials from different CSPs.
VAAFI-16	SSO Enablement	VAAFI shall be able to accommodate users going to multiple VA E-Authentication enabled applications. (Allow SSO)
VAAFI-17	CSP Selection	User's starting at VAAFI shall be redirected to the CSP selection page unless the target application specifies otherwise.
	SSOe Provides PID, BIRLS, SEC ID, and EDI PI IDs	<ul style="list-style-type: none"> • [FEAT461935] SSOe shall provide all instances of PID IDs to target applications during authentication. • [FEAT461936] SSOe shall provide all instances of BIRLS file number ID to target applications during authentication. • SSOe shall provide all instances of SEC ID to target applications during authentication. • SSOe shall provide all instances of EDI PI to target applications during authentication. <p>See table 'LOA 2+ Authentication Traits Business Rules' at the end of this section.</p>
AccessVA-1	Entry through AccessVA	User Enters through AccessVA
AccessVA-2	Registration through Application	New User Registers (at a Credential Service Provider) through Application
AccessVA-3	Registration through AccessVA	New User Registers (at a Credential Service Provider) through AccessVA
AccessVA-4	Login to CSP	Log on to AccessVA (Logs on the Credential Service Provider through AccessVA)
AccessVA-5	Login to Application	User logs into Application through AccessVA

ID	Specific Requirement/Synopsis	Requirement
WI 150259	AccessVA Unauthenticated Home Page [FEAT461883]	On the AccessVA Unauthenticated Home Page, the AccessVA logo shall be displayed on the top left of the screen.
WI 150260	AccessVA Unauthenticated Home Page [FEAT461884]	On the AccessVA Unauthenticated Home Page, the IAM logo shall be displayed on the top right of the screen.
WI150261	AccessVA Unauthenticated Home Page [FEAT461885]	On the AccessVA Unauthenticated Home Page, the wording, "Please select a VA Website," shall be displayed.
WI 150268	AccessVA Unauthenticated Home Page [FEAT461892]	On the AccessVA Unauthenticated Home Page, when the user selects a partner website, they are taken to the AccessVA CSP selection page.
WI 150269	AccessVA Processing Page [FEAT461894]	On the AccessVA Processing Page, AccessVA shall display a processing notification page during third-party onboarding (3POB)
WI 150270	AccessVA Processing Page [FEAT461895]	On the AccessVA Processing Page, AccessVA's 3POB Processing notification page shall inform the user that their request is processing and may take up to the 30 seconds.
WI 150271	AccessVA CSP Selection Page [FEAT461897]	On the AccessVA CSP Selection Page, AccessVA shall have a CSP selection page displaying the accepted CSPs for the selected target application.
WI 150272	AccessVA CSP Selection Page [FEAT461898]	On the AccessVA CSP Selection Page, the AccessVA CSP selection page shall display a logo for each accepted CSP.
WI 162495	AccessVA CSP Selection Page [FEAT461899]	On the AccessVA CSP Selection Page, the AccessVA CSP Selection page shall have a "select another VA Application" button.
WI 150274	AccessVA CSP Selection Page [FEAT461900]	On the AccessVA CSP Selection Page, the "select another VA Application" button on the AccessVA CSP Selection page shall return the user to the AccessVA Unauthenticated page.
WI 162498	AccessVA CSP Selection Page [FEAT499971]	On the AccessVA CSP Selection Page, the logo for the application selected shall be displayed in the top left corner
WI 162499	AccessVA CSP Selection Page [FEAT499972]	On the AccessVA CSP Selection Page, the AccessVA CSP selection page CSP logos shall dynamically disappear to allow additional space for the text box, when the browser size is reduced.

ID	Specific Requirement/Synopsis	Requirement
WI 150275	AccessVA Authenticated Page [FEAT461902]	On the AccessVA Authenticated Page, the AccessVA Authenticated page shall be redesigned to be more consistent with the style of the redesigned AccessVA Unauthenticated page.
WI 150276	AccessVA Authenticated Page [FEAT461903]	On the AccessVA Authenticated Page, the AccessVA Authenticated page shall be updated to feature the log out capability.
WI 162504	Addition of an AccessVA Sign-In Partners Page [FEAT499973]	A new menu selection for Sign-In Partners shall be displayed in the AccessVA menu bar
WI 162506	Addition of an AccessVA Sign-In Partners Page [FEAT499974]	The Sign-In Partners page shall display “Sign-In Partners:” at the top of the page
WI 162509	Addition of an AccessVA Sign-In Partners Page [FEAT499975]	The Sign-In Partners page shall display a button with the logo for each Sign-In Partner available via AccessVA
WI 162513	Addition of an AccessVA Sign-In Partners Page [FEAT499976]	Upon selection of one of a Sign-In Partner button, information about that Sign-In Partner will be displayed under the wording, “About Sign-In Partner”
WI 162519	Addition of an AccessVA Sign-In Partners Page [FEAT499977]	“About Sign-In Partner” shall display the Sign-In Partner logo, and information to include; 1) Who Qualifies for the Sign-In Partner; 2) Which applications are accessible after logging in with the selected Sign-In Partner; 3) and links to logon, or register for the selected Sign-In Partner
WI 162523	Addition of an AccessVA Sign-In Partners Page [FEAT499978]	Upon logon or registering from the AccessVA Sign-In Partner page, the user will be returned to the AccessVA Authenticated Page.
WI 162528	AccessVA Widget [FEAT499979]	The AccessVA widget AccessVA logo shall be updated to the most current AccessVA logo.
WI 162534	AccessVA Widget [FEAT499981]	The AccessVA widget CSP logos shall be increased in size.
WI 150277	AccessVA Login Button [FEAT461906]	An AccessVA login button shall appear on the VA.gov. website.
WI 150278	AccessVA User Selector Pop-Up Widget [FEAT461908]	The AccessVA User Selector Pop-Up Widget shall display on the VA.gov website after the user selects the AccessVA login button.
WI 150279	AccessVA User Selector Pop-Up Widget [FEAT461909]	The AccessVA User Selector Pop-Up Widget shall darken the va.gov website underneath it.

ID	Specific Requirement/Synopsis	Requirement
WI 150280	AccessVA User Selector Pop-Up Widget [FEAT461910]	The AccessVA User Selector Pop-Up Widget User Type buttons shall display asking the user to identify themselves as by one of the following user types: Veteran, Family Member, Service Member, or VSO.
WI 150281	AccessVA User Selector Pop-Up Widget [FEAT461911]	The AccessVA User Selector Pop-Up Widget shall display buttons for the AccessVA Partner websites relevant to the selected user type when the user type button is selected.
WI 150282	AccessVA User Selector Pop-Up Widget [FEAT461912]	When the user selects a partner website from the AccessVA User Selector Pop-Up Widget, the AccessVA CSP Selection Pop-Up Widget shall appear.
WI 163757	AccessVA Unauthenticated Home Page [FEAT506032]	As an AccessVA user accessing the Unauthenticated Home Page, I want to see the IAM and AccessVA logos in the same banner, to meet VA application standards.
	AccessVA Unauthenticated Home Page [FEAT506033]	As an AccessVA user accessing the Unauthenticated Home Page, I want to see the Home Page banner in a 2 color gradient starting in blue (#1677bd) and ending in white (#FFFFFF), to meet VA application standards.
	AccessVA Unauthenticated Home Page [FEAT506034]	As an AccessVA user accessing the Unauthenticated Home Page, I want to see the text "Securing your Access to the VA", to meet VA application standards.
	AccessVA Unauthenticated Home Page [FEAT506035]	As an AccessVA user accessing the Unauthenticated Home Page, I want to see a top and bottom thin outline in red (#E31B23) on the Home Page banner, to meet VA application standards.
	AccessVA Unauthenticated Home Page [FEAT506036]	As an AccessVA user accessing the Unauthenticated Home Page, I want to see an introduction paragraph "Welcome to AccessVA, your solution for accessing VA's many web resources. AccessVA provides login capability to website and applications as part of VA's Identity and Access Management (IAM) Enterprise" on the Home Page, to meet VA application standards.
	AccessVA Unauthenticated Home Page [FEAT506037]	As an AccessVA user accessing the Unauthenticated Home Page, I want to see buttons with application logos for the AccessVA Partner web sites integrated with AccessVA on the Home Page, to meet VA application standards.

Table 7: Target Portal Strategy Requirements

ID	Specific Requirement/Synopsis	Requirement
WI 150283	Support Imprecise Date of Birth from MVI [FEAT 473915]	SSOe shall accept an imprecise date of birth from MVI.
WI 150284	SSOe Accepts Address from VDS [FEAT 473917]	SSOe shall accept an address from VDS during authentication
WI 150529	SSOe Accepts Address and Phone # from VDS	SSOe shall support a 3-line formatted street address.
WI 150532	SSOe Accepts Address and Phone # from VDS	SSOe shall accept a phone number from VDS during authentication.

Table 8: Third-Party Credential Onboarding Requirements

ID	Specific Requirement/Synopsis	Requirement
WI 162540	Third Party Credential Onboarding [FEAT473917]	<p>AccessVA shall collect street address in 3-line format.</p> <p>Address:</p> <ul style="list-style-type: none"> • Street Line 1, String: 3 – 35 • Street Line 2, String: 3 – 30 • Street Line 3, String: 3 – 30

Table 9: JSON/STS

ID	Specific Requirement/Synopsis	Requirement
VAAFI STS-1	SPEC192.7.11.1	Exchange user session data when presented a valid SSO user token.
VAAFI STS-2	SPEC192.7.11.2	Support authenticate web clients TLS client-auth
VAAFI STS-3	SPEC192.7.11.3	Support authenticate web clients WS-Security X509
VAAFI STS-4	SPEC192.7.11.4	Response message shall support SAML
VAAFI STS-5	SPEC192.7.11.5	Response message shall support XML encryption for message and attributes individually
VAAFI STS-6	SPEC192.7.11.6	Support XML digital signature
VAAFI STS-7	SPEC192.7.11.7	Support WS-Trust protocol

ID	Specific Requirement/Synopsis	Requirement
VAAFI STS-8	SPEC192.7.11.8	Support WS-Policy protocol
VAAFI STS-9	SPEC192.7.11.9	Support WS-Security
VAAFI STS-10	SPEC192.7.11.10	Support attribute retrieval from Provisioning (VDS)
WI 150285	SSOe Accepts JSON Web Token and Bearer Token [FEAT 461919]	SSOe shall provide a JSON web token within the Open Authorization Standard (OAuth) framework.
WI 150286	SSOe Accepts JSON Web Token and Bearer Token [FEAT 461920]	SSOe shall provide a JSON web token within the Security Token Service (STS) framework.
WI 150287, 162542	SSOe Accepts JSON Web Token and Bearer Token [FEAT 461921]	SSOe shall be able to digitally sign and encrypt JSON web tokens.
WI 150288, 162546	SSOe Accepts JSON Web Token and Bearer Token [FEAT 461922]	SSOe shall be able to verify digitally signed JSON web tokens and decrypt encrypted JSON web tokens.
WI 150289	SSOe Accepts JSON Web Token and Bearer Token [FEAT 461923]	SSOe shall accept JSON Web tokens
WI 150290	SSOe Accepts JSON Web Token and Bearer Token [FEAT 461924]	SSOe shall accept JSON Bearer tokens.
WI 150291	SSOe Expose STS Service with REST Interface [FEAT 461926]	SSOe shall expose the STS service with a REST interface.
WI 150292	SSOe Expose STS Service with REST Interface [FEAT 461297]	SSOe shall secure the REST STS service with mutual Transport Layer Security (TLS).
WI 150534	SSOe Expose STS Service with REST Interface [FEAT 473911]	SSOe shall exchange the user session data when presented with a valid REST SSO user token.

Table 10: OAuth

ID	Specific Requirement/Synopsis	Requirement
VAAFI OAuth-1	Support the mobile application authorization through the use of OAuth 2.0	Shall support the authentication of SSOe user with cookie as authentication token.
VAAFI OAuth-2	Support the mobile application authorization through the use of OAuth 2.0	Shall support the authentication of the client with URL parameter as Authorization token
VAAFI OAuth-3	Support the mobile application authorization through the use of OAuth 2.0	Shall support the authentication of SSOe user with PKI
VAAFI OAuth -4	Support the mobile application authorization through the use of OAuth 2.0	Shall support the mobile client registration
VAAFI OAuth -5	Support the mobile application authorization through the use of OAuth 2.0	Shall support the token query and cache
VAAFI OAuth -6	Support the mobile application authorization through the use of OAuth 2.0	Shall support the management and enforcement of OAuth policies
VAAFI OAuth -7	Support the mobile application authorization through the use of OAuth 2.0	Shall support fine-grained revocation
VAAFI OAuth -8	Support the mobile application authorization through the use of OAuth 2.0	Shall support limiting the number of access or refresh tokens
VAAFI OAuth -9	Support the mobile application authorization through the use of OAuth 2.0	Shall support the self-registration of clients
WI 150293	OAuth Validation Service [FEAT 461929]	SSOe shall provide an access token validation service.
WI 150294	OAuth Validation Service [FEAT 461930]	SSOe shall provide a validation service for the STS SAML token.

ID	Specific Requirement/Synopsis	Requirement
WI 150295	oAuth Validation Service [FEAT 461931]	SSOe shall provide a validation service for the JSON web token.
WI 150296	oAuth Validation Service [FEAT 461932]	SSOe s validation services shall be supported by SOAP and REST interfaces.
WI 150297	oAuth Validation Service [FEAT 461933]	SSOe s access token validation service shall provide both Extensible Markup Language (XML) and JSON responses.
WI 150535	oAuth Validation Service [FEAT 473912]	SSOe shall secure the OAuth service with mutual Transport Layer Security (TLS).
WI 150536	oAuth Validation Service [FEAT 473913]	SSOe shall exchange the user session data when presented with a valid OAuth user access token.
WI 150537	oAuth Validation Service [FEAT 473914]	SSOe s OAuth validation service REST response messages shall support the JSON format

SSOe Headers: The LOA 2+ authentication traits have been updated for this increment.

Table 11: LOA 2+ Authentication Traits Business Rules

Authentication Trait (HTTP Header)	Required /Optional	Source Primary (Secondary)	Source Attribute/Trait Name	Action If Trait Not Provided	Additional Business Rules
va_eauth_ICN	R	MVI	ICN	Pass User to Third-Party Credential Onboarding	N/A
va_eauth_PID	O	MVI	Person ID (Corporate DB)	Pass Null Value	If 2 or more PIDs are returned, then provide all in header as a comma separated string
va_eauth_filenummer	O	MVI	File Number (BIRLS)	Pass Null Value	If 2 or more BIRLS file numbers are returned, then provide all in header as a comma separated string
va_eauth_secID	R	Prov	secID	Pass User to Third-Party Credential Onboarding	If 2 or more secID are returned, then provide all in header as a comma separated string
va_eauth_dodedipnid	O	MVI	va_eauth_dod edipnid	<ol style="list-style-type: none"> If CSP is DS LOGON <ol style="list-style-type: none"> and EDI PI returned from MVI is different, then pass value from CSP and EDI PI not in MVI, then pass EDI PI from CSP If CSP is not DSLOGON, and EDI PI is not in MVI, then pass null 	If 2 or more EDI PI are returned, then provide all in header as a comma separated string
va_eauth_csid	R	CSP	va_eauth_csid	Do Not Authenticate	N/A

Authentication Trait (HTTP Header)	Required /Optional	Source Primary (Secondary)	Source Attribute/Trait Name	Action If Trait Not Provided	Additional Business Rules
va_eauth_uid	R	CSP	va_eauth_uid	Do Not Authenticate	N/A
va_eauth_hash	R	CSP	va_eauth_hash	Do Not Authenticate	N/A
va_eauth_assurancelevel	R	CSP	va_eauth_assurancelevel	Do Not Authenticate	N/A
va_eauth_authenticationmethod	O	CSP	va_eauth_authenticationmethod	Pass Null Value	N/A
va_eauth_authenticationauthority	O	CSP	va_eauth_authenticationauthority	Pass Null Value	N/A
va_eauth_commonname	O	CSP	va_eauth_commonname	Pass Null Value	N/A
va_eauth_email	O	CSP (Prov)	email	Pass Null Value	<ul style="list-style-type: none"> If email is provided by CSP, then pass CSP If email is not provided by CSP, then pass Prov value
va_eauth_LastName	R	MVI	va_eauth_LastName	Do Not Authenticate	N/A
va_eauth_FirstName	O	MVI	va_eauth_FirstName	Pass Null Value	N/A
va_eauth_MiddleName	O	MVI	MiddleName	Pass Null Value	N/A

Authentication Trait (HTTP Header)	Required /Optional	Source Primary (Secondary)	Source Attribute/Trait Name	Action If Trait Not Provided	Additional Business Rules
va_eauth_PNID	O	MVI (CSP)	va_eauth_PNID	Pass Null Value	<ul style="list-style-type: none"> If SSN is provided by MVI, then pass MVI If SSN is not provided by MVI, then pass CSP
va_eauth_Prefix	O	MVI	Prefix	Pass Null Value	N/A
va_eauth_Suffix	O	MVI	Suffix	Pass Null Value	N/A
va_eauth_Gender	O	MVI	Gender	Pass Null Value	N/A
va_eauth_DOB	O	MVI	Date of Birth (DOB)	Pass Null Value	N/A
va_eauth_transactionID	R	SSOe	Transaction ID for session user	Do Not Authenticate	N/A
va_eauth_IssueInstant	R	CSP	The time of authentication to the CSP	Do Not Authenticate	N/A
va_eauth_csp_object	O	CSP	Derived JSON package of CSP's additional traits*	Pass Null Value	N/A

Authentication Trait (HTTP Header)	Required /Optional	Source Primary (Secondary)	Source Attribute/Trait Name	Action If Trait Not Provided	Additional Business Rules
va_eauth_mhvien	O	MVI	MHV IEN (MHV)	Pass NOT_FOUND	If two or more MHV IENs are returned, then put all in header as a comma separated string
va_eauth_street	O	MVI (CSP)	va_eauth_street	Pass Null Value	N/A
va_eauth_street_2	O	MVI (CSP)	va_eauth_street	Pass Null Value	N/A
va_eauth_street_3	O	MVI (CSP)	va_eauth_street	Pass Null Value	N/A
va_eauth_city	O	MVI (CSP)	va_eauth_city	Pass Null Value	N/A
va_eauth_state	O	MVI (CSP)	va_eauth_state	Pass Null Value	N/A
va_eauth_country	O	MVI (CSP)	va_eauth_country	Pass Null Value	N/A
va_eauth_postalcode	O	MVI (CSP)	va_eauth_postalcode	Pass Null Value	N/A
CSP Data Only	R	SSOe		N/A	N/A
IAM Service Down	R	SSOe		N/A	N/A

*Data elements from the va_eauth_csp_object must not be used to override the elements provided in the common authentication traits.

Notes:

- Required means that if this data element is not available from the Primary Source, do not authenticate the user
- Optional means that if this data element is not available from the Primary Source take alternative action
- Pass User to Third-Party Credential Onboarding
 - Pass Null Value in header
 - Pass Value from secondary source
 - Other as specified in the business rules table

Refer to Section 8 for human interface information.

2.5.2. Overview of Functional Workload/Performance Requirements

Functional workload volumes describe in business terms the amount of work to be performed. This section does not attempt to include any technical design decisions. Table 12 identifies the significant functional workload and functional performance requirements.

Table 12: Workload and Performance Requirements

ID	Requirement
P-1	The failure of any one element or module of VAAFI does not result in all VAAFI services being disabled.
P-2	VAAFI supports up to 4 million authentications per month.
P-3	VAAFI shall support up to 7500 concurrent users.
P-4	AccessVA shall support threshold average page load Essential time under light load conditions (10 requests/minute) <= 5 seconds.
P-5	AccessVA shall support threshold average page load Essential time under normal load conditions (100 requests/minute) <= 8 seconds.
P-6	AccessVA shall support threshold average page load Essential time under peak load conditions (1000 requests/minute) <= 10 seconds.

AccessVA and SSOe for this increment shall support the following:

2.5.2.1. SSOe (Application Junction, Reassertion Provider, CSP/IdP Federation Partner)

Table 13: Usage Profile (User Authentication Events)

Mean Daily volume	80000
Projected Growth	20000/year
Peak Daily volume	120000
Projected Growth	30000/year
Peak Hourly volume	10000

Days of operation	Sunday-Saturday
Hours of operation	24/7
Peak Hours	8am-10p.m.Eastern
Maximum Response Time	10 seconds

2.5.2.2. SSOe STS

Table 14: Usage Profile (Token Requests)

Projected Growth	20000/year
Peak Daily volume	0
Projected Growth	50000/year
Peak Hourly volume	0
Days of operation	Sunday-Saturday
Hours of operation	24/7
Peak Hours	8am-10p.m.Eastern
Maximum Response Time	10 seconds
Projected Growth	20000/year

2.5.2.3. SSOe oAuth

Table 15: Usage Profile (Token Requests)

Mean Daily volume	0
Projected Growth	20000/year
Peak Daily volume	0
Projected Growth	50000/year
Peak Hourly volume	0
Days of operation	Sunday-Saturday
Hours of operation	24/7
Peak Hours	8am-10p.m.Eastern
Maximum Response Time	10 seconds

2.5.2.4. SSOe (WebService Client, Webservice Producer)

Table 16: Usage Profile (Webservice Calls)

Mean Daily volume	6000
Projected Growth	1500/year
Peak Daily volume	6800

Projected Growth	1700/year
Peak Hourly volume	950
Days of operation	Sunday-Saturday
Hours of operation	24/7
Peak Hours	8am-10p.m.Eastern
Maximum Response Time	10 seconds

2.5.3. Overview of Operational Requirements

Table 17: Operational Requirements

ID	Requirement
O-1	VAAFI shall successfully complete the VA certification and accreditation process (C&A).
O-2	VAAFI shall allow for automated backup from the hosting facility.
O-3	VAAFI shall maintain local load balancers within each site for load balancing and fail over.
O-4	VAAFI shall achieve 99.99% availability during scheduled up time as defined by the VA. (8.76 hrs. per year of unscheduled downtime are allowed.)

2.5.4. Overview of the Technical Requirements

This section provides the major technical requirements that drive the conceptual design that later sections in the document describe. Please refer to the Master RTM below for Technical Requirements.



Table 18: Technical Requirements

ID	Requirement
T-1	VAAFI shall be able to consume SAML assertions from a participating credential service provider.
T-2	VAAFI shall implement SAML 1.0 and 2.0 specifications as per partner requirements
T-3	VAAFI shall be able to consume PKI credentials. Two supported variants are: <ul style="list-style-type: none"> • DoD CAC • VA PIV

ID	Requirement
T-4	VAAFI shall generate/create and assign a unique E-GUID (Globally Unique Identifier) or HASH for all system users.
T-5	VAAFI shall forward users to the agency application with the application specific assertion attributes and the E-GUID or HASH in the http headers.
T-9	VAAFI shall appropriately handle authentication errors, such as unknown CSP, hand-off failures. The system may use the following error codes: Error Code Usage 10 AA Unavailable 20 CS Unavailable 30 AA Invalid 40 CS Invalid 50 CS assurance level does not meet AA's assurance level 60 CS refused to issue Identity Assertion 70 Hand-off error 90 Unknown Exception (Application Down for Maintenance)
T-10	VAAFI shall log artifacts and every element in the assertion
T-11	VAAFI shall have the ability to correlate local session identifiers with associated authenticated transactions
T-12	VAAFI shall not overwrite log files and should support a process to archive logs for three years
T-13	VAAFI Portal Pages shall be designed per the VA web development standards.
T-14	All VAAFI User Interface pages shall be section 508 compliant.
T-15	Failure of any one element or module of VAAFI shall not result in the E-Authentication service being disabled.
T-16	If the Assurance Level attribute in the Assertion has a value less than is required by the VA Application, VAAFI shall display a page indicating that the assurance level is insufficient for the end user to access the application.
T-17	VAAFI shall maintain multiple Federated Identity Manager modules providing HTTP/S and SOAP services for the purpose of validating assertions.
T-18	VAAFI shall maintain multiple web service proxies that enforce policy decisions and allow agency applications protected access to web services provided by external entities
T-19	The AccessVA design is compliant with the VA Enterprise Architecture.
T-20	VAAFI shall be able to consume Connect.Gov credentials

As this document describes in later sections, VAAFI is built using the ITFIM Suite of applications. VA selected this product in early 2006 after a study of competing products. Tivoli Federated Identity Manager and Tivoli Directory Server are approved for the TRM.

2.5.5. Overview of the Security or Privacy Requirements

Table 19 outlines the security and privacy requirements:

Table 19: Security Requirements

ID	Requirement
S-1	VAAFI shall be compliant with the supplemental guidance to OMB A-130 (to include Appendix III) given in the OMB memorandum E-Authentication Guidance For Federal Agencies, Joshua B. Bolten, M-04-04, December 16, 2003, http://www.whitehouse.gov/omb/memoranda/fy04/m04-04.pdf .
S-2	VAAFI shall be compliant with the technical requirements for authentication guidance given in NIST Special Publication 800-63-2: Electronic Authentication Guideline.
S-3	VAAFI shall be in compliance with National Institute of Standards and Technology (NIST) SP 800-53 as required by the Federal Information Security Management Act (FISMA) for a system rated High Impact
S-4	VAAFI shall connect to partner systems using mutual Secure Sockets Layer (SSL) meeting Federal Information Processing Standards Publication (FIPS PUB)140-2: Security Requirements for Cryptographic Modules, compliance standards
S-5	VAAFI shall encrypt system logs on backup tapes per the System Security Plan
S-6	VAAFI shall be fault-tolerant.
S-7	VAAFI shall support disaster recovery at a primary and hot secondary location.
S-8	VAAFI shall not use any personally identifiable information received during the authentication or authorization process for any other purpose unless the CSP provides written consent to the RP and the end user provides explicit permission to the CSP.
S-9	Confidential information being stored, processed, transported or disposed of shall be marked as confidential by the data/information owner, and the receiver shall protect it as such unless otherwise specified by these operation rules
S-10	AccessVA will implement the necessary technical controls as outlined in NIST 800-53 version 4.
S-11	SPEC192.7.11 Supports authenticating web service clients using TLS client-auth and WS-Security X509
S-12	SPEC192.7.11 Supports WS-Trust, WS-Policy, and WS-Security
S-13	SPEC192.7.11 STS response message will support XML encryption for message and attributes individually, and XML digital signature.

VAAFI was issued an Authorization to Operate (ATO) in September 2007 and this ATO was renewed in July 2011.

2.5.6. Overview of System Criticality and High Availability Requirements

The Contingency Plan provides a Business Impact Analysis (BIA), which describes the business requirements for criticality and high availability requirements.

2.5.7. Single Sign-on Requirement

Not Applicable

2.5.8. Requirement for Use of Enterprise Portals

Not Applicable

2.5.9. Special Device Requirements

VAAFI makes use of several WebSphere DataPower SOA appliances for proxying web services for partners. See Section 3.3.4.5 below for more information.

The AccessVA component has no identified special device requirements.

2.6. Legacy System Retirement

Currently, no VAAFI legacy system retirements are within the scope of this document.

Table 20: Proposed Legacy Retirements

Legacy System or Legacy System Component	System Retired or Workload Reduced	Quantify the Workload Reduction
N/A	N/A	N/A

3. Conceptual Design

This section details the conceptual design of VAAFI.

3.1. Conceptual Application Design

The previous phases of the VAAFI project, which began in 2004, surveyed, tested, and selected a software product for the implementation of VAAFI. At that time, all federal agencies were mandated to use software products from an approved list. VA selected the ITFIM suite of products.

Due to VAAFI being a COTS product implementation, most of this section does not apply.

3.1.1. Application Context

3.1.1.1. VAAFI Target Portal Strategy

The VAAFI Target Portal Strategy will interact with other IAM services per the following diagram:

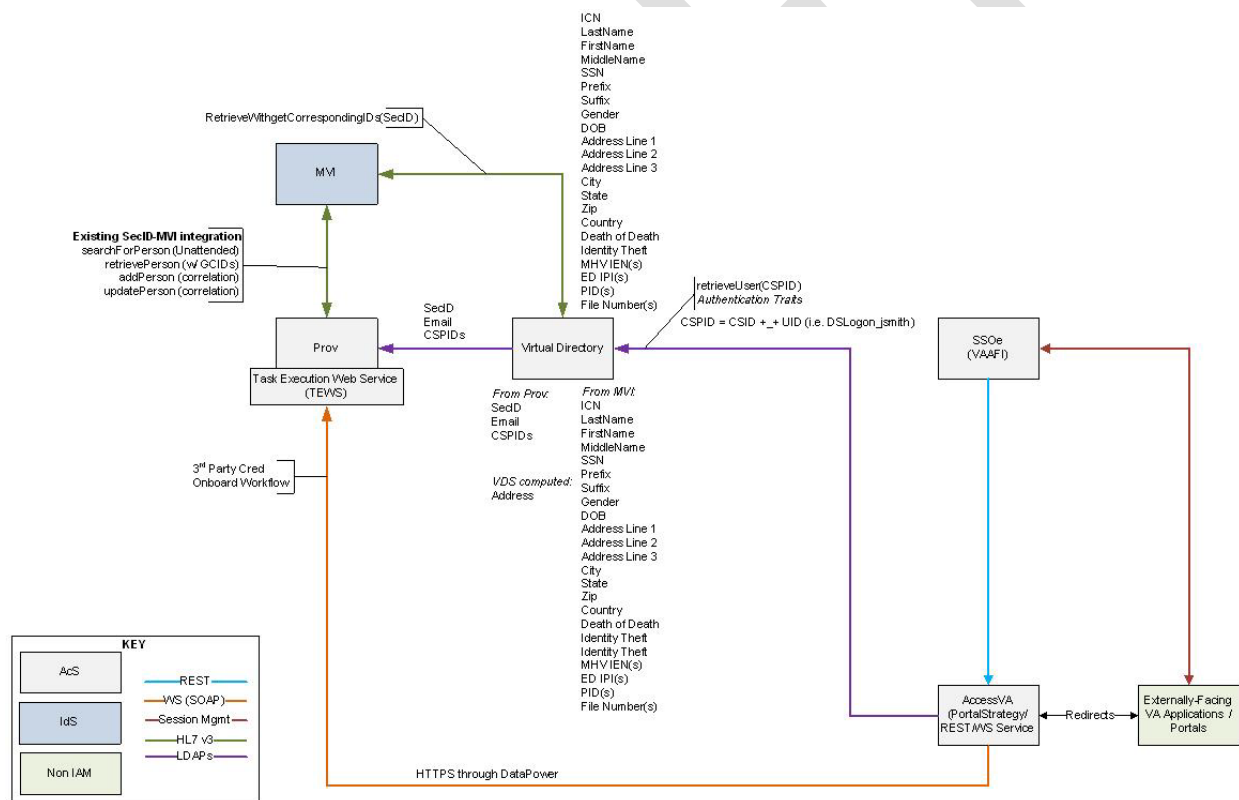


Figure 3: MVI-VDS Integration Supporting Target Portal Strategy

3.1.1.2. Access VA

This section is not applicable to the majority of the VAAFI system. However, it is applicable to the AccessVA component. The following diagram shows the application context for AccessVA. AccessVA is a front end (switchboard) to the various services such as the federation services,

security services, and the application integration and SSO services provided by the VAAFI Infrastructure. Figure 4, below, provides an example.

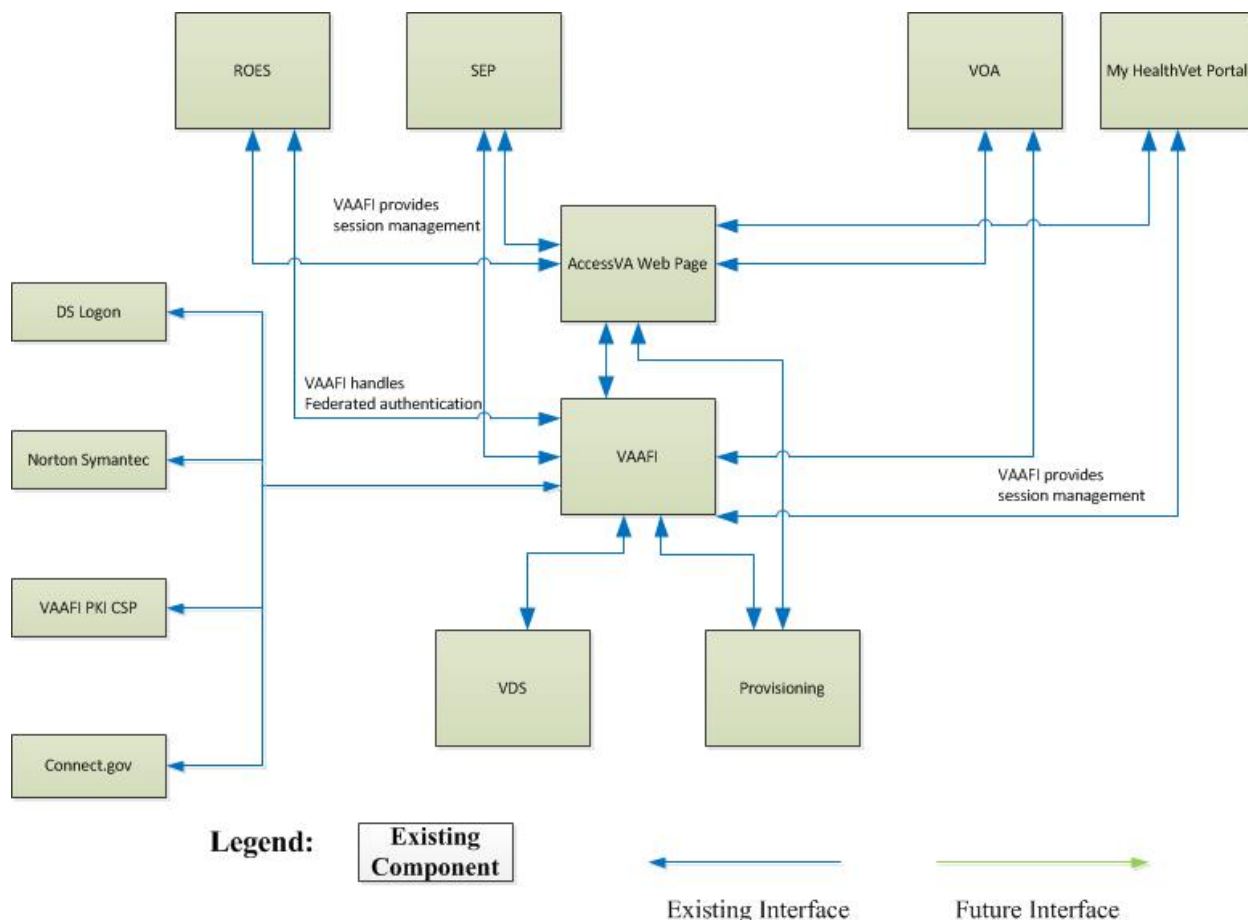


Figure 4: AccessVA Application Context – Example Flow

The AccessVA application context diagram illustrates the various foundational infrastructure components that the AccessVA system uses. A CSP communicates with the VAAFI infrastructure through the security standards SAML to provide federated SSO functionality. The VAAFI infrastructure provides a variety of security services to integrate the credential from the CSP to the various VAAFI protected applications.

3.1.1.3. OAuth

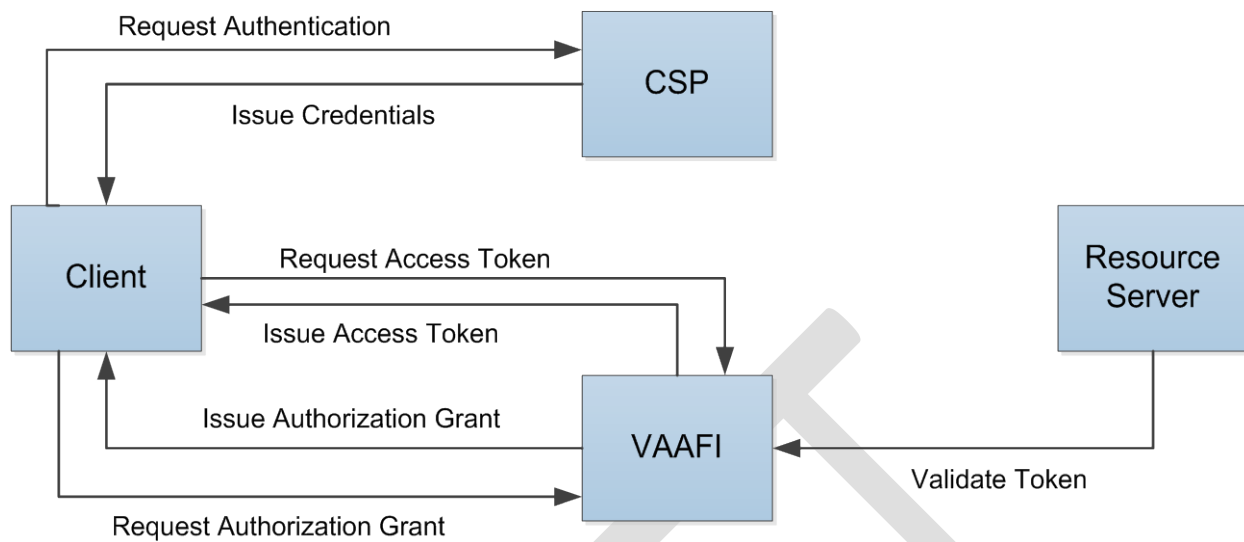


Figure 5: OAuth

These customizations leverage TFIM's pluggable architecture to provide enhancements to the COTS implementation. The customizations include:

- Client Configuration Provider: Provides the ability to externalize OAuth client configurations in order to provide self-registration of OAuth clients.
- Token Cache Provider: Permits externalized storage and query of OAuth grants and tokens.
- Trusted Client Manager: Permits externalized storage and query of resource owner authorization decisions.

A Relational Database is used as an external storage to support the customizations listed above. Additionally, the solution includes a web-based application to provide the following functionalities:

- Client registration;
- Device Registration; and
- Consent Management.

The following diagram provides a high-level application design of OAuth:

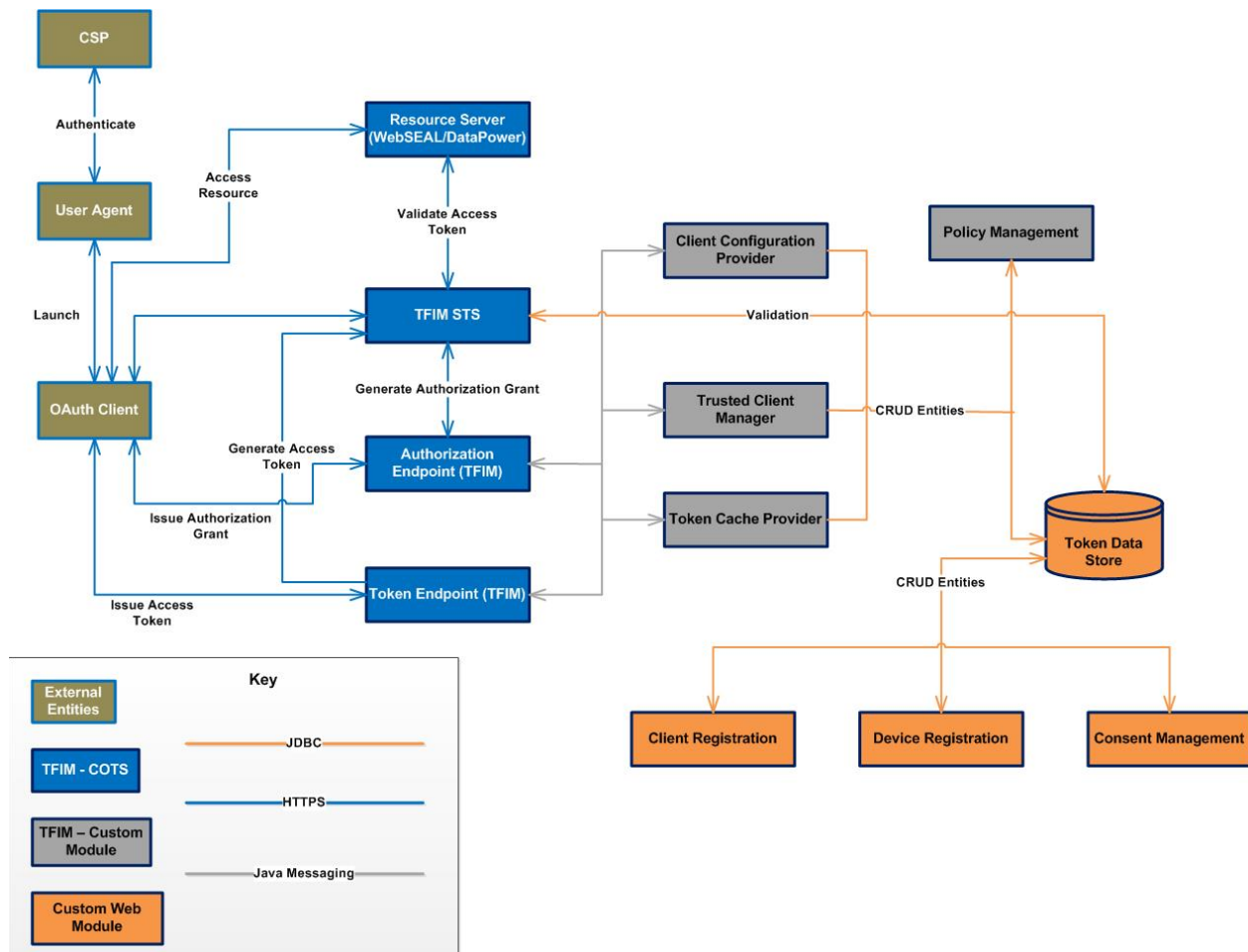


Figure 6: OAuth High-Level Design

3.1.2. High-Level Application Design

Refer to Figure 4. Details will be provided at the end of Sprint 6.

3.1.3. Application Locations

Table 21: Application Locations

Application Component	Description	Location at Which Component is Run	Type
VAAFI	Primary	Terremark, Florida	All
VAAFI	Secondary	Terremark, Virginia	All

Table 22: Application Users

Application Component	Location	User
eAuth Users	Worldwide	Veterans and their designees
AccessVA	Worldwide	Anonymous users

3.2. Conceptual Data Design

VAAFI only passes individual user information and does not store it. Although a database instance exists within VAAFI, the database is simply back-end storage for the directory server. Within the directory server, virtual users are stored along with configuration information for the IBM products; again, no user information is stored in the directory.

AccessVA does not use a database or a data store. The data is stored in the various systems that interact with AccessVA such as CSPs, VAAFI, and other infrastructure components.

AccessVA has an XML data matrix that stores the relationships between CSP, protected applications, and levels of assurance.

The VAAFI OAuth implementation uses an external data store (DB) to store OAuth artifacts. These artifacts include OAuth Trusted Clients, OAuth Tokens (Authorization Code and Access Token), and Registered OAuth Clients. The following specifies the conceptual data model of the entities leveraged by the OAuth solution:

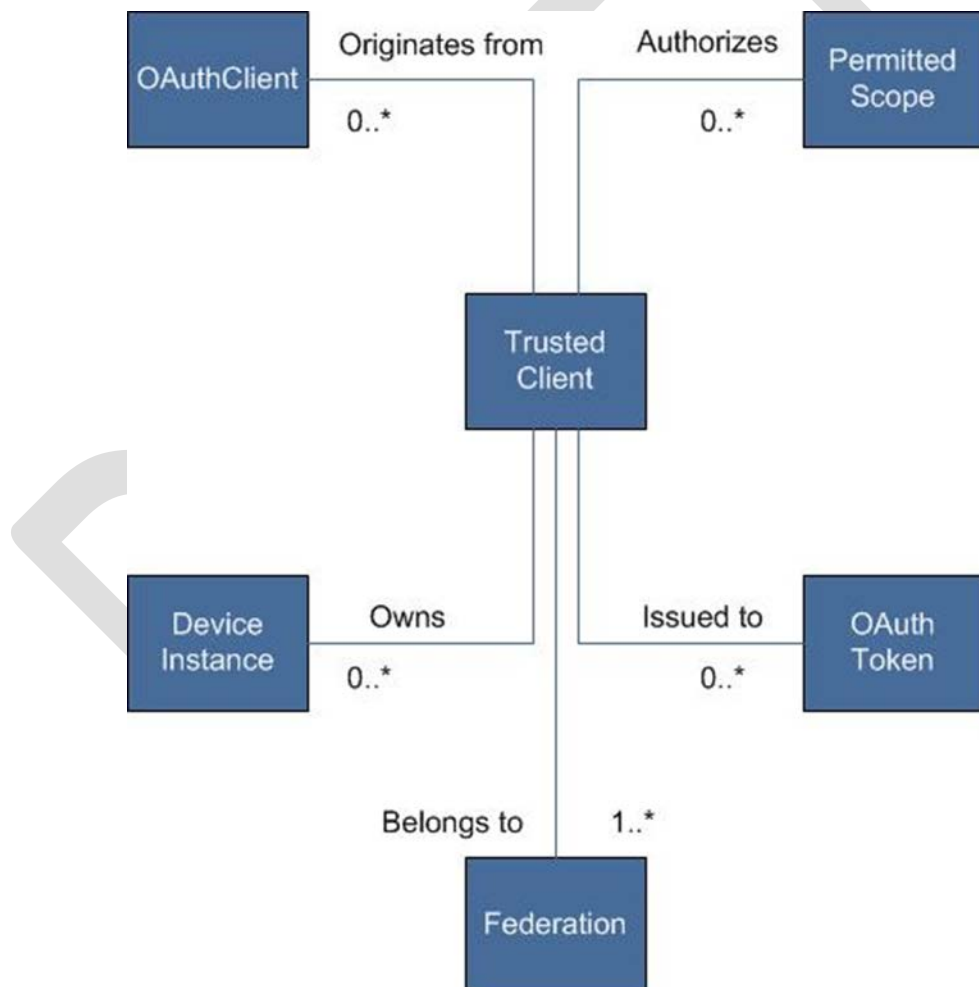


Figure 7: OAuth DB Data Model

Internally, the TFIM components use an XML document, called the Security Token Service Universal User (STUniversalUser) document, to represent the data and pass between modules.

The type definition of this document is in the schema form and is in Section A.6 as additional information.

3.2.1. Project Conceptual Data Model

This section is not applicable.

3.2.2. Database Information

Table 23: Database Inventory

Database Name	Description	Type	Steward
DB2 (OAuth Schema)	OAuth uses the DB2 database for storing entities	RDBMS	N/A

3.2.3. User Interface Data Mapping

This section is not applicable.

3.3. Conceptual Infrastructure Design

VAAFI is an implementation of the ITFIM System implemented with a flexible, modular, and scalable architecture. It supports Veterans and VA business partners conducting business transactions and exchanging information with all qualifying VA applications using a VAAFI credential. With this infrastructure, the end users will be able to use a single authentication credential (currently either a login ID and password, or a PIV or CAC) to access multiple applications. The primary location for the system is at the Terremark facility in Florida. The alternate site for VAAFI operations is the Terremark facility in Virginia.

The system includes several subsystems on top of a general support system managed by Corporate Data Center Operations (CDCO) consisting of ESX servers, Storage Area Network (SAN), and networking components. The subsystems of VAAFI include the following:

- SP;
- PKI CSP;
- Web Service Proxy; and
- AccessVA.

Older versions of VAAFI had a Common Audit and Reporting Service (CARS) subsystem. This was a VAAFI-specific service. CARS was replaced when VAAFI moved to Terremark, Splunk, a very powerful generic tool replaced CARS. Splunk aggregates all the server logs, not just the Tivoli logs. Each server in VAAFI has a Splunk agent that forwards logs to a Splunk server that processes the logs. Now, an IAM-specific service call Common Audit and Reporting (CAR) uses a Computer Associates (CA) product to aggregate and integrate the logging throughout IAM. A CAR agent has been added to the VAAFI servers (and the Splunk agent will continue to provide logs to Splunk). VAAFI will continue to use Splunk to generate the VAAFI Monthly Usage Report, which contains enhancements to include more data about back-end applications and services usage. This document does not describe Splunk's design, because the infrastructure team provides Splunk as a service for all Terremark enclaves. A configuration file on the server

determines what files are sent to Splunk. Similarly, CAR is a service that a separate IAM project provides. Aside from the CAR agent installation on VAAFI servers, the VAAFI design has no insights into the design or workings of CAR. Log file management is handled through scripts and is described in the Production Operations Manual.

The AccessVA component of VAAFI uses two additional Apache web servers and three Oracle WebLogic application servers, one of which serves as a deployment manager.

3.3.1. System Criticality and High Availability

3.3.1.1. Contingency Capability and Backups

A secondary site contingency capability resides in a Terremark data center in Culpeper, VA. It is a “Warm” site because all its servers are running and ready to process authentications or web service requests, although no Production traffic flows to the site during normal operation. The VA Gateway group has a health check set on the five VAAFI endpoints (eauth.va.gov, services.eauth.va.gov, pki.eauth.va.gov, register.eauth.va.gov and access.va.gov) at the primary site in Miami, FL. If those health checks fail, VA Gateway automatically routes traffic to the Culpeper site. Per this SDD, the instance of VAAFI at both sites is identical; therefore, this document does not distinguish between the sites.

In addition to load balancing, fault tolerance, and secondary site, VAAFI has backups that Terremark performs and tests. Backups essentially are the following (see the POM for a fuller description):

- Full backups are taken of virtual machines on a weekly basis.
- Backups of virtual machines must be transported off site at least monthly.
- Backups of specific databases will occur daily between 2 a.m. and 5 a.m. The POM will provide the databases locations.
- Backups of DataPower logs, which include PII, must be encrypted. The POM will identify the log locations.

3.3.1.2. Secure and Unsecure Access to AccessVA

AccessVA supports secured (authenticated) and unsecured (unauthenticated) access. Access is provided over Secure Sockets Layer (SSL)/Transport Level Security (TLS). The AccessVA Apache virtual host settings redirect all HTTP traffic to HTTPS.

CSPs integrated with VAAFI provide AccessVA authentication. VAAFI provides the Single-Sign-On ability via WebSEAL integration to AccessVA. The WebSEAL junction name for the AccessVA application is accessva to match the web application context root configured for AccessVA.

An unauthenticated user accesses the unauthenticated content in AccessVA over the HTTPS port 443. When the user authenticates with a VAAFI CSP partner, VAAFI negotiates with that partner to establish the authenticated identity of the user. Then VAAFI routes the user back to AccessVA (over port 443) or a protected application using the combination of one of the VAAFI SSO domains and the configured WebSEAL junction for AccessVA or other protected applications. As long as the user accesses AccessVA via the VAAFI SSO domain (EAUTH, etc.), VAAFI will apply the SSO headers to the HTTPS outbound request and AccessVA will

examine the headers when an AccessVA controller handles the request. If AccessVA can determine that the supplied VAAFI headers indicate an authenticated user, then AccessVA will route the user to the authenticated content area within AccessVA.

3.3.1.3. OAuth High Availability – Disaster Recovery (HADR)

A description of the very high-level architecture follows.

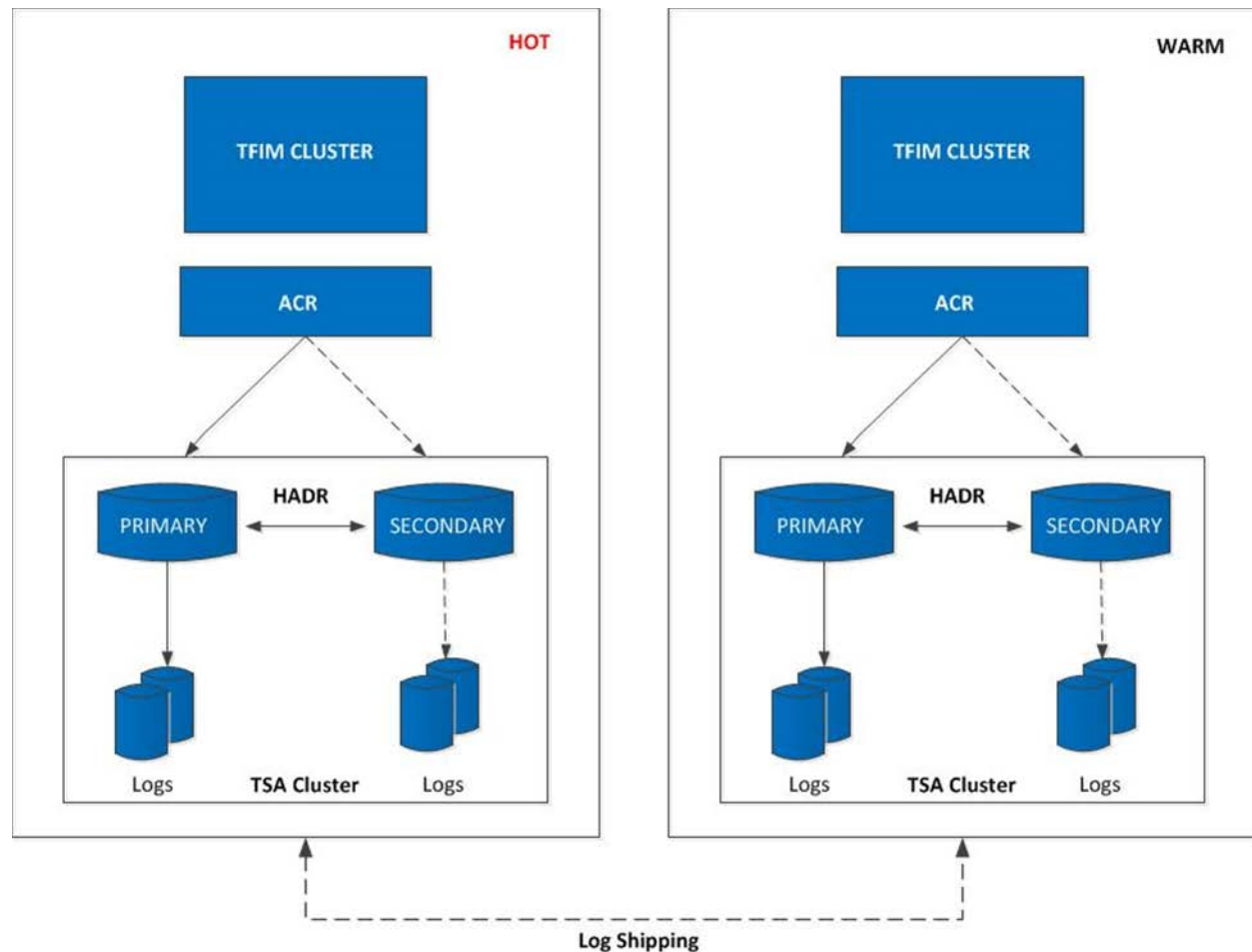


Figure 8: OAuth HADR Architecture

- DB2 HADR synchronizes the SECONDARY DB with the PRIMARY DB through log shipping.
- Tivoli System Automation cluster monitors the DB2 instances within a site and handles failover between PRIMARY and SECONDARY. When the PRIMARY fails, TSA swaps roles between the PRIMARY and SECONDARY.
- Automatic Client Reroute (ACR) directs DB requests to the PRIMARY DB.
- Synchronization of DBs between sites occurs through the log shipping mechanism as follows:
 - The archived logs from the HOT site's primary DB are sent to the WARM site's primary DB's active log path or to the overflow log path using a Secure File Transfer Protocol (SFTP).

- This push mechanism is automated through a DB2 user exit program according to a well-established schedule.
- A scheduled job initiates a roll-forward recovery to the end of the logs on the WARM site's primary DB. (HADR setup ensures these updates propagate to the secondary DB.) This processes the log records received from the HOT site resulting in the synchronization of the OAuth DB.

The frequency at which the log shipping process executes determines the amount of data loss during an unplanned failover to the WARM site. All transactions (client registrations, user consents, etc.) occurring during the time interval between the latest log shipping process and an unplanned failover will be lost if no manual intervention to synchronize the OAuth DBs across sites exists.

3.3.2. Special Technology

The table below identifies the special technology implemented as part of VAAFI.

Table 24: Special Technology Requirements

Special Technology	Description	Notional Location	TRM Status
WebSphere DataPower XI52	The DataPower device provides the needed WebService capabilities to VAAFI	All	Yes

3.3.3. Technology Locations

This section is not applicable.

Table 25: Technology Location Details

Technology Component Production 1	Location	Usage
Workstations	N/A	N/A
Special Hardware	N/A	N/A
Interface Processors	N/A	N/A
Legacy Mainframe	N/A	N/A
Legacy Application Server	N/A	N/A
Legacy Databases	N/A	N/A
Other	N/A	N/A

3.3.4. Conceptual Infrastructure Diagram

Figure 9 shows VAAFI with many of its internal and external connections exposed. The following sections of this document will describe each infrastructure subsystem. Each section will describe these external connections, and show an internal breakdown of the components.

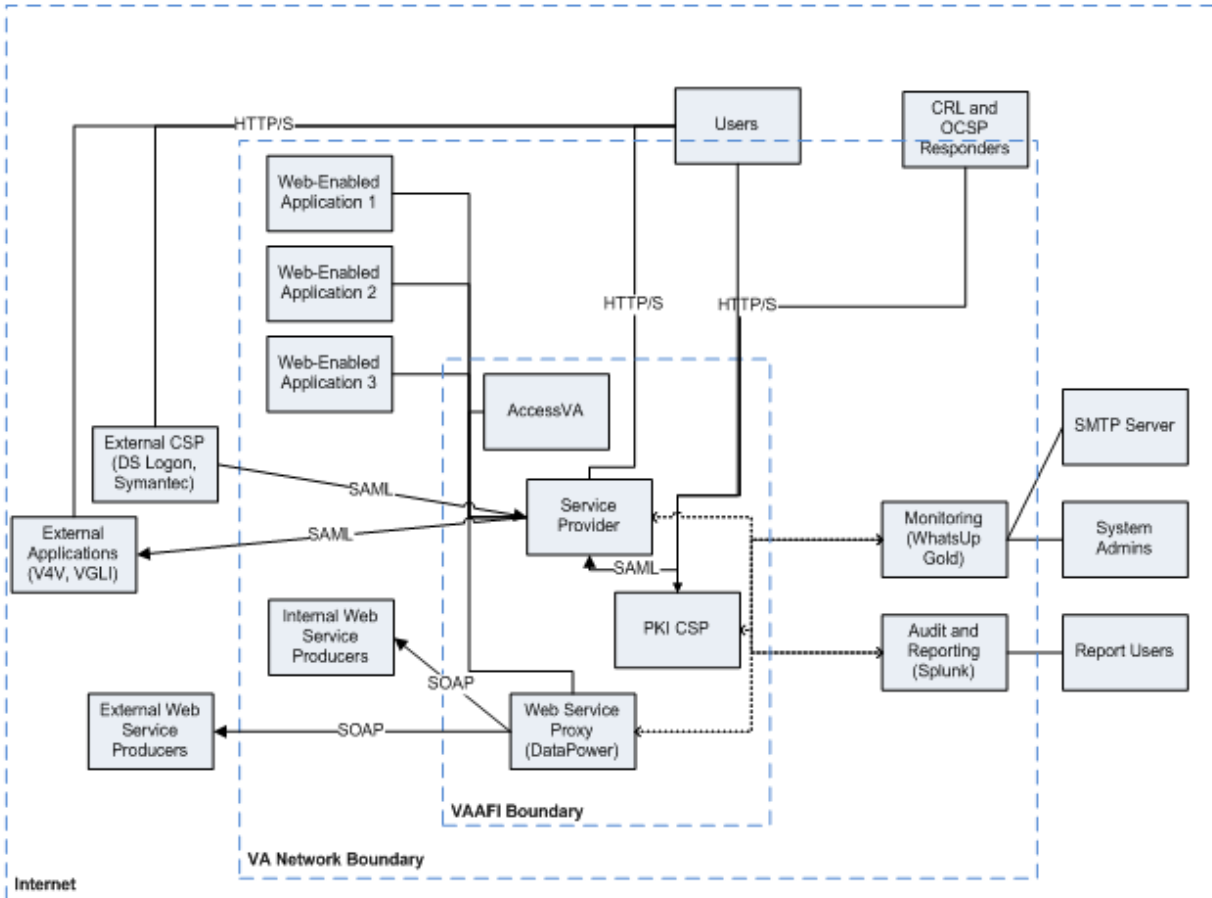


Figure 9: Conceptual Infrastructure Diagram

3.3.4.1. Location of Environments and External Interfaces

Details will be provided during Spring 6.

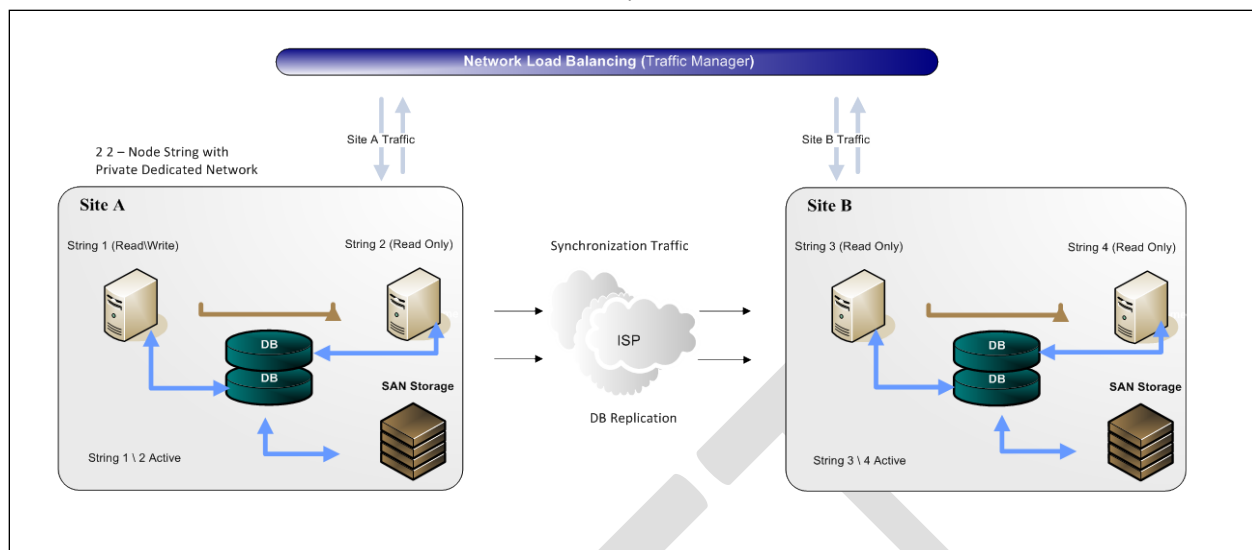


Figure 10: Network Load Balancing

3.3.4.2. Conceptual Production String Diagram

Details will be provided during Spring 6.

3.3.4.3. VA Authentication Federation Service Provider

The Service Provider provides authentication services to VA web-enabled applications through accepting SAML assertions from CSPs both internal to the infrastructure, such as the PKI CSP, and external, such as Norton Symantec and DS Logon. Users authenticate to a CSP, creating an identity assertion, and then passing a pointer to that assertion (called an artifact) along with the application they wish to access to the Service Provider. The Service Provider connects to the CSP, retrieves and validates the identity assertion, and then brokers the connection from the user to the application destination. The components of the VAAFI Service Provider include reverse proxy servers, a server dedicated to TAM, servers dedicated to ITFIM, and Lightweight Directory Access Protocol (LDAP) servers. ITFIM supports OAuth 2.0, a lightweight authorization technology that VA is pursuing for some of its mobile applications.

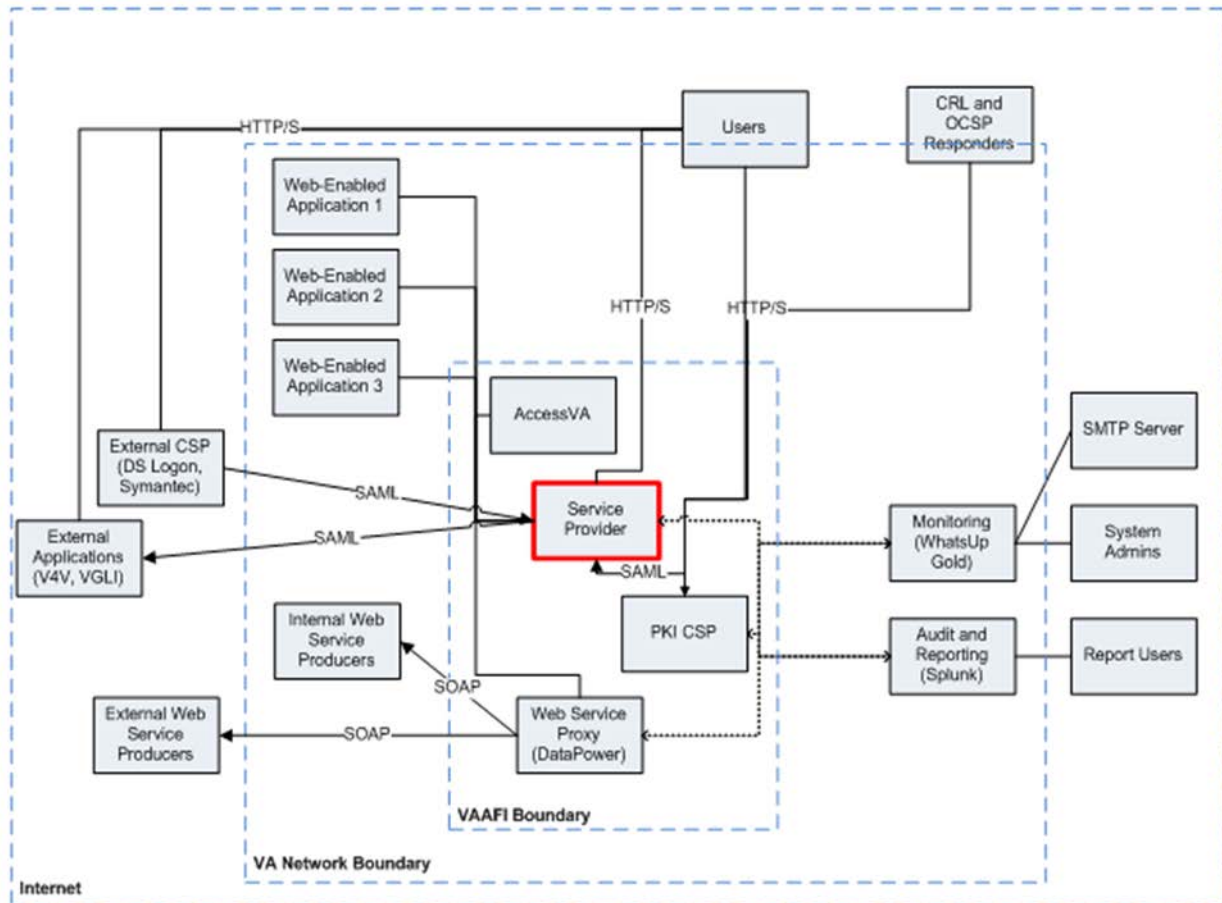


Figure 11: VA Authentication Federation Service Provider

A primary and secondary Load Balancer reside inside the first firewall and distribute traffic to the Tivoli WebSEAL Reverse Proxy servers. Tivoli Session Manager Server (SMS) manages the sessions sent to WebSEAL to ensure a session receives the correct treatment regardless of the availability of a particular WebSEAL. WebSEAL sends the request through the second firewall to TAM. Requests sent to an application must have carefully constructed headers adhering to the SAML specification. TAM, ITFIM, and the LDAP server parse these headers and, based on preconfigured information, determine if access to the application is allowed. The system diagram in Figure 12 displays each of the components of the system.

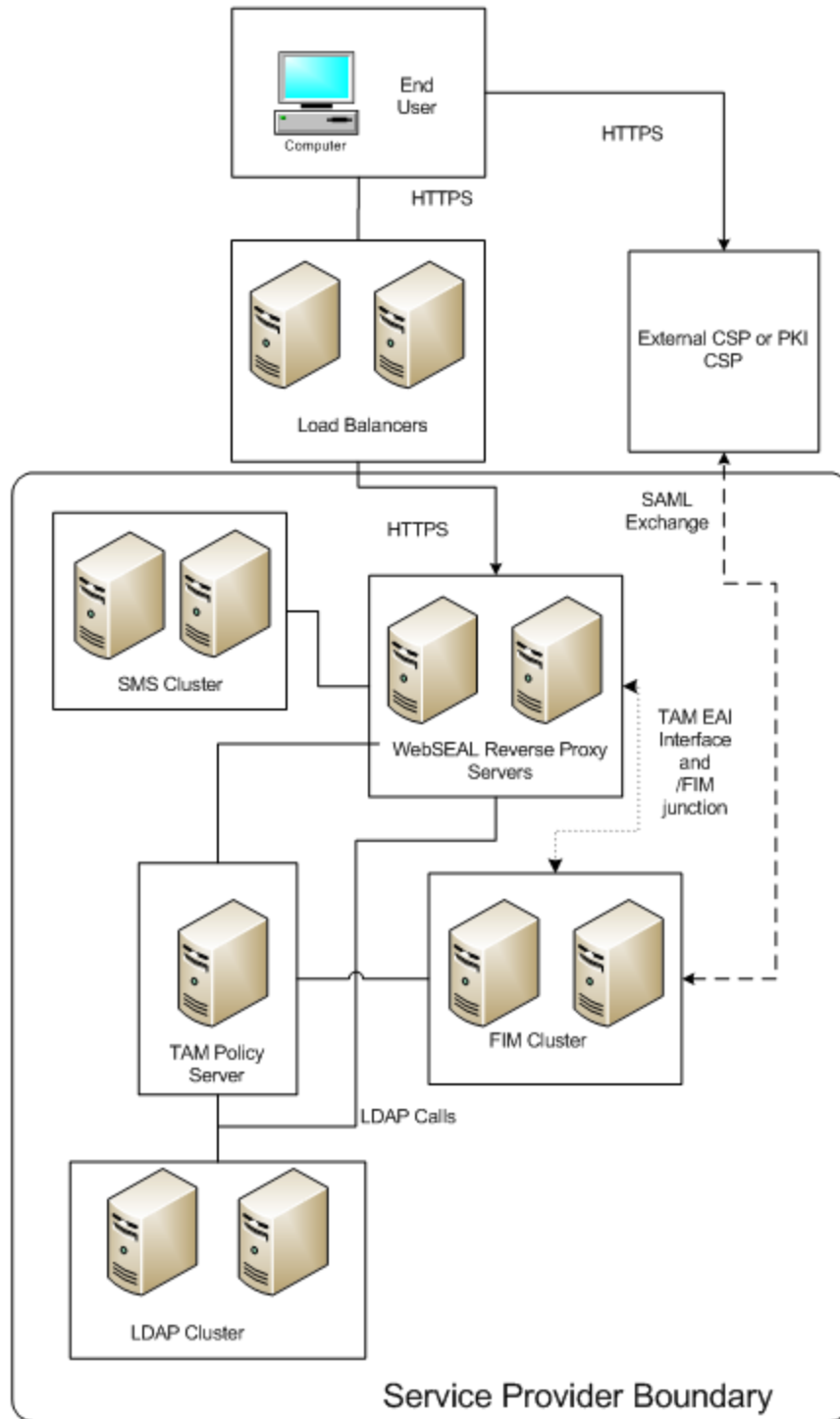


Figure 12: System Diagram

NOTE: This architecture only determines whether a credential has been properly validated and traffic has been redirected securely. The credential provider determines that the proper login ID and password is entered, and the application determines if the end user has rights to use the application and, if so, the level of access rights that user should be allowed.

3.3.4.4. VAAFI PKI CSP (SPEC192.7.11.15)

The VAAFI PKI CSP is a subsystem of VAAFI similar to the VAAFI SP. Like the SP, the PKI CSP conforms to the SAML 1.0, 1.1, and 2.0 standards. The VAAFI PKI CSP and the VAAFI SP both employ the same IBM Tivoli infrastructure. The VAAFI SP is a SAML receiver and the VAAFI CSP is a SAML provider. They work together as trusted partners in a federated model using a SAML 1.1 federation and the SAML 1.1 browser artifact profile. The VAAFI PKI CSP allows the use of CACs and PIVs as a federated credential, enabling VA protected applications to accept PKI credentials without PKI validation and verification development. Part of the validation process for CAC and PIV cards is to check that they have not been revoked. To do this, the system checks the Certificate Revocation Lists (CRL) or Online Certificate Status Protocol (OCSP) Responders for those credentials. The acceptance of CACs and PIVs allows VAAFI to support applications requiring a credential assurance level of 3. Currently, VAAFI supports CAC and only VA PIVs. ITFIM supports Holder-of-Key, the underpinning of sending SAML assertions with an assurance level of 4 per the SAML 2.0 standard, but requires a partner to implement and test.

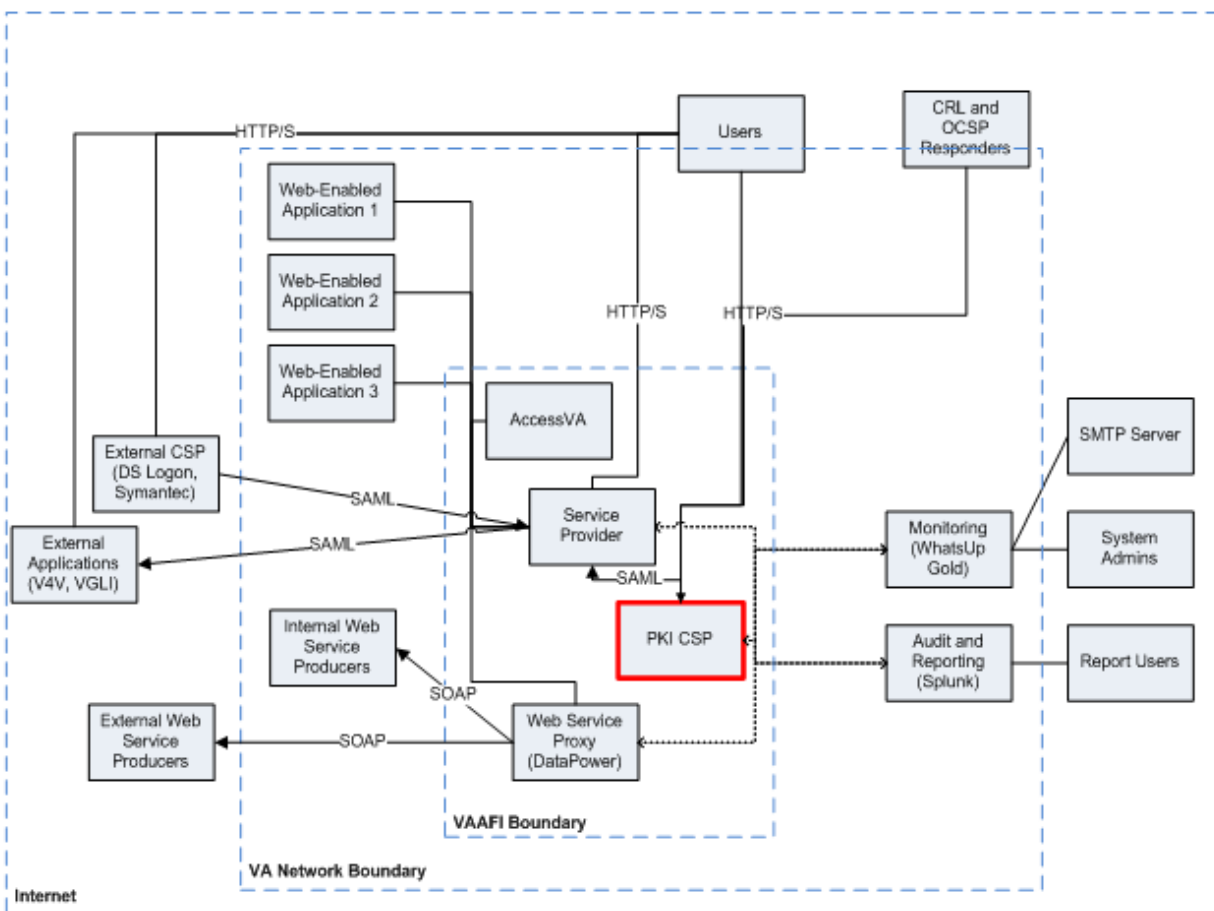


Figure 13: VAAFI PKI CSP

The PKI CSP consists of an implementation of the ITFIM very similar to the VAAFI Service Provider. Connectivity between components is similar, as is the redundancy between components. The PKI CSP has an additional part that the Service Provider does not need, a

registration application. Users connect through an IBM HTTP Server (IHS) to a WebSphere Application Server that runs an application that enrolls the users' Distinguished Name (DN) from their card into to the TAM user store. The diagram in Figure 14 loosely depicts the interaction between the components.

DRAFT

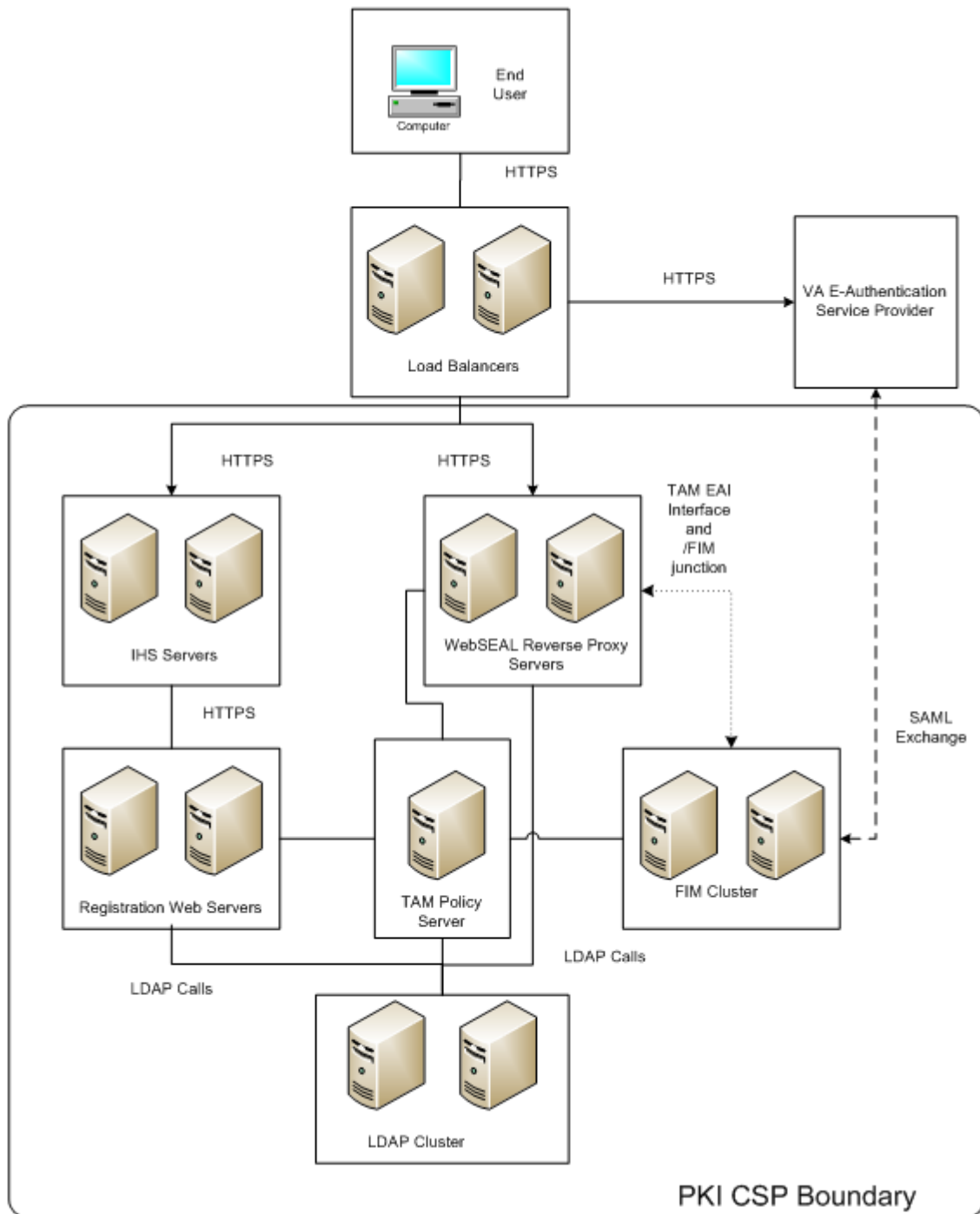


Figure 14: VAAFI PKI CSP – Component Interaction

3.3.4.5. Web Service Proxy

The Web Service Proxy offers various capabilities to the enterprise, such as endpoint relocation, composite web services, and web service operation hiding.

The Web Service Proxy also provides validation services for messages passing through. The inspector can validate messages based on standard XML and SOAP definitions or application-

specific schemas can do this. Web services using the Message Inspector pattern receive a certain level of automatic protection against the following XML threats, which require no additional configuration:

- XML Entity Expansion and Recursion Attacks
- XML Wellformedness-based Parser Attacks
- Memory Space Breach and Buffer Overflow Attacks
- Public Key Denial of Service (DoS) Attacks
- Resource Hijack Attacks
- Data Tampering Attacks
- Schema Poisoning Attacks

The options with additional configuration based on client need and threat include the following:

- Single Message XML Denial of Service (XDoS) Protection
 - XML Document Size Attacks
 - XML Document Width Attacks
 - XML Document Depth Attacks
 - Jumbo Payload Attacks
 - MegaTags – Jumbo Tag Names
- Multiple Message XML Denial of Service (MMXDoS) Protection
- SQL Injection Protection
- Protocol Threat Protection
- XML Virus (X-Virus) Protection
 - XML Virus Attacks
 - XML Encapsulation Attacks
 - Payload Hijack Attacks
 - Binary Injection Attacks
- Dictionary Attack Protection

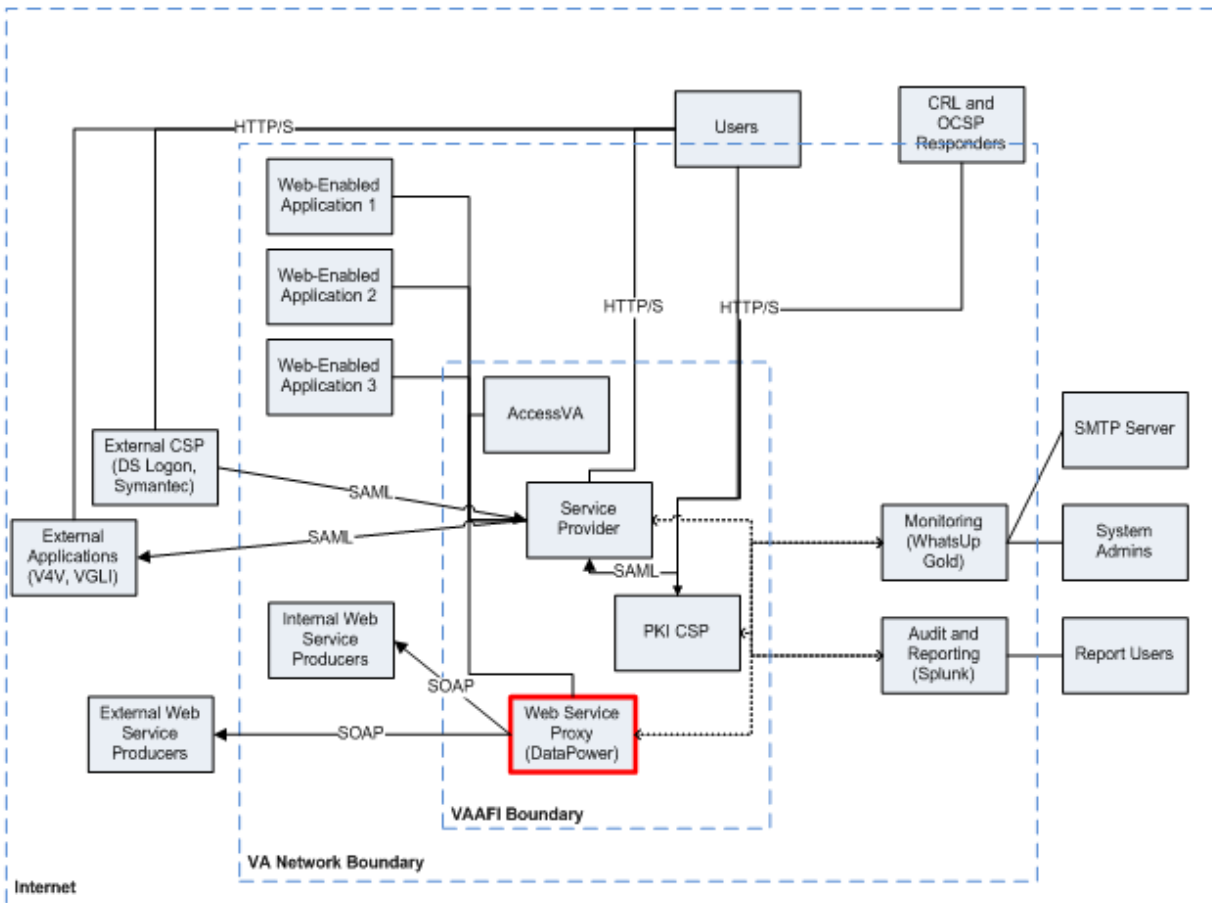


Figure 15: Web Service Proxy

The Web Service Proxy also offers security services at a higher layer (i.e., above the transport layer) than offered through the Secure Pipe pattern, although the two can be combined where necessary. Operating at the XML document and/or SOAP message level, this pattern offers digital signature and encryption capabilities for the document element or entire document. It also understands and can operate within the Organization for the Advancement of Structured Information Standards (OASIS) Web Services Security standards with SOAP-based security envelope and message body encryption and digital signature services. The infrastructure supports the following OASIS WS standards:

- WS-Security (Digital Signature and Encryption)
- WS-Policy
- WS-Addressing (Gatewayed and Pure)
- WS-ReliableMessaging
- WS-I Conformance Policies
 - Basic Profile 1.0 and 1.1
 - Attachment Profile 1.0
 - Basic Security Profile 1.0

Currently, partners use certificate authentication to authenticate to the services. Access to web services authorization is per operation and per web service.

3.3.5. VAAFI Concepts

This section discusses some major concepts applied to VAAFI.

3.3.5.1. Reverse Proxy

One of the primary benefits the ITFIM product offers is the ability for applications to participate in VAAFI without requiring a FIM type agent installed and maintained in each of the participating VA applications. This capability further minimizes the level of initial software modification and long-term maintenance VA applications require, in addition to the minimization offered by electing the infrastructure approach discussed earlier. Instead of using an agent, the ITFIM product uses a reverse proxy (WebSEAL).

A reverse proxy server is installed in a tightly controlled zone of the VAAFI network called the Demilitarized Zone (DMZ). Traffic coming from the Internet that uses VAAFI and whose destination is one of the AA servers flows through the reverse proxy server for the duration of the connection. The reverse proxy is the Policy Enforcement Point (PEP) and enforces access control decisions. VAAFI uses WebSEAL (the reverse proxy) along with Tivoli Access Manager to provide the Identity and Access Management policy enforcement.

After verifying that traffic is passing from a trusted CSP, the PEP proxies on behalf of users to the agency application. VAAFI passes to the application a hash value of concatenated attributes provided in the identity assertion. The hash value will be included in the WebSEAL proxy server credential. In an unsuccessful authentication, VAAFI redirects traffic with specific error messages.

3.3.6. Authentication Roles

VAAFI will perform the tasks of authenticating the source of the asserting party and ensure that the asserting CSP provides sufficient assurance levels. The VAAFI SP will not authenticate the end user; that is the role of the CSP. VAAFI SP will authenticate the traffic between the CSP, the application, and the end user. VAAFI will also extract attributes provided in the assertion. The appropriate ICD lists the attributes for a given CSP.

3.3.6.1. Registration and Activation

All applications will continue to maintain their own registration processes and user repository. VAAFI does not include a registration capability, but instead relies on the application registration processes. The methodology of activating new users of VAAFI, or “activation,” occurs by binding the system user through an AA defined process. VA applications participating in VAAFI will store the binding information in their existing user repositories. Each application must accept and consume the identity information to authenticate users and assign the appropriate level of access to each user. They do this in one of several ways based on the security pattern implemented for that specific partner. Activation approaches for each application vary and details are in the VAAFI Application Integration Guide.

3.3.6.2. User Audit and Reporting

The VAAFI implementation has a robust user audit and reporting capability. All audit events are formatted as XML and are completely configurable per enablement. Basic user reporting contains information related to single sign-on, single logout, name identifier management, and message security. Each type of event has additional sub-logging capability that can be implemented as needed. Utilizing the FIM Common Audit Service, the following standard audit elements are included:

- ContextDataElements
- SourceComponentIdelements
- Situation
- Outcome

As enablements are created, audit and reporting requirements are identified and logging is enabled on an “as needed” basis. This is to limit performance and system impact do to the increased processor and disk space required.

4. System Architecture

This section describes the system and subsystem(s) architecture for VAAFI and discusses the general architectural decisions that the VAAFI Project has made.

4.1. High-Level Architecture

This section of the document provides technical specifications for VAAFI.

4.1.1. System Boundaries

The VAAFI members' systems fall into two separate realms, representing the major components of the federation:

- External CSP Network, which is the Identity Provider (IdP); and
- VA Network, which hosts the AA, AcS/Provisioning Services, and VAAFI.

One entity (organization) can act in more than one role within VAAFI. In these cases, a member will be uniquely identified as being in a particular role. For example, VAAFI acts as a CSP as well as an RP. This can allow other Federation members to trust a VA-issued credential and allow VA end users to use the VA-issued credential with any federal government applications that are participating in the VAAFI.

Each federation member maintains its own infrastructure and resources and participates in VAAFI by sharing the usage of those resources with the other entities. Although each partner is solely responsible for maintaining its own physical structure, the logical links and boundaries define the maintenance and troubleshooting efforts for each VAAFI member.

4.1.2. Architectural Principles

Throughout the VAAFI design, the following architectural principles apply. These architectural principles stem from VAAFI partner requirements and best practices for designing and building highly scalable, secure, and resilient infrastructures.

- **Security:** VAAFI has been designed with a Defense in Depth approach. This ensures that rings of the infrastructure are protected with security controls at each layer and the controls at each layer are progressively more restrictive approaching the center. Each component in the architecture has a unique certificate to sign and encrypt communication between the other VAAFI components and external components.
- **Flexibility:** Given the likelihood of future changes in the handling of Veteran registration and activation to one or more VA applications, VAAFI does not preclude adoption of other methods or schemes to register and activate users as other systems emerge or are modified. VAAFI can and has incorporated new and diverse features to support authentication and security services as requirements develop. VAAFI can integrate into heterogeneous environments through a comprehensive set of Application Program Interfaces (APIs).
- **Scalability:** A critical aspect of the VAAFI Project is the concept of growth. The design of the initial operating capability was to meet the immediate and near-term sizing goals of the participating VA applications. What the end user acceptance rate will be is not yet known; however, as additional CSPs become available, increased end user awareness is

probable, along with corresponding increases in end user participation levels. One of the main objectives for the VAAFI Project is to design a system capable of expansion to support a very large population of users.

4.1.3. High-Level Network Architecture

Because of the complexity of a federated identity management deployment, this document will address the network architecture and software architecture.

4.1.3.1. Network Layout

The network zones and their transport classifications within the network include the following:

- Internet Zone (Uncontrolled);
- VA Network (Controlled);
- Demilitarized Zone (DMZ) (Controlled); and
- Internal Zone (Restricted).

The components within each of the zones provide specific services to the infrastructure, as Figure 16 depicts.

VAAFI SP Interconnections				
	VA Intranet	VAAFI Internal	VAAFI DMZ	Internet

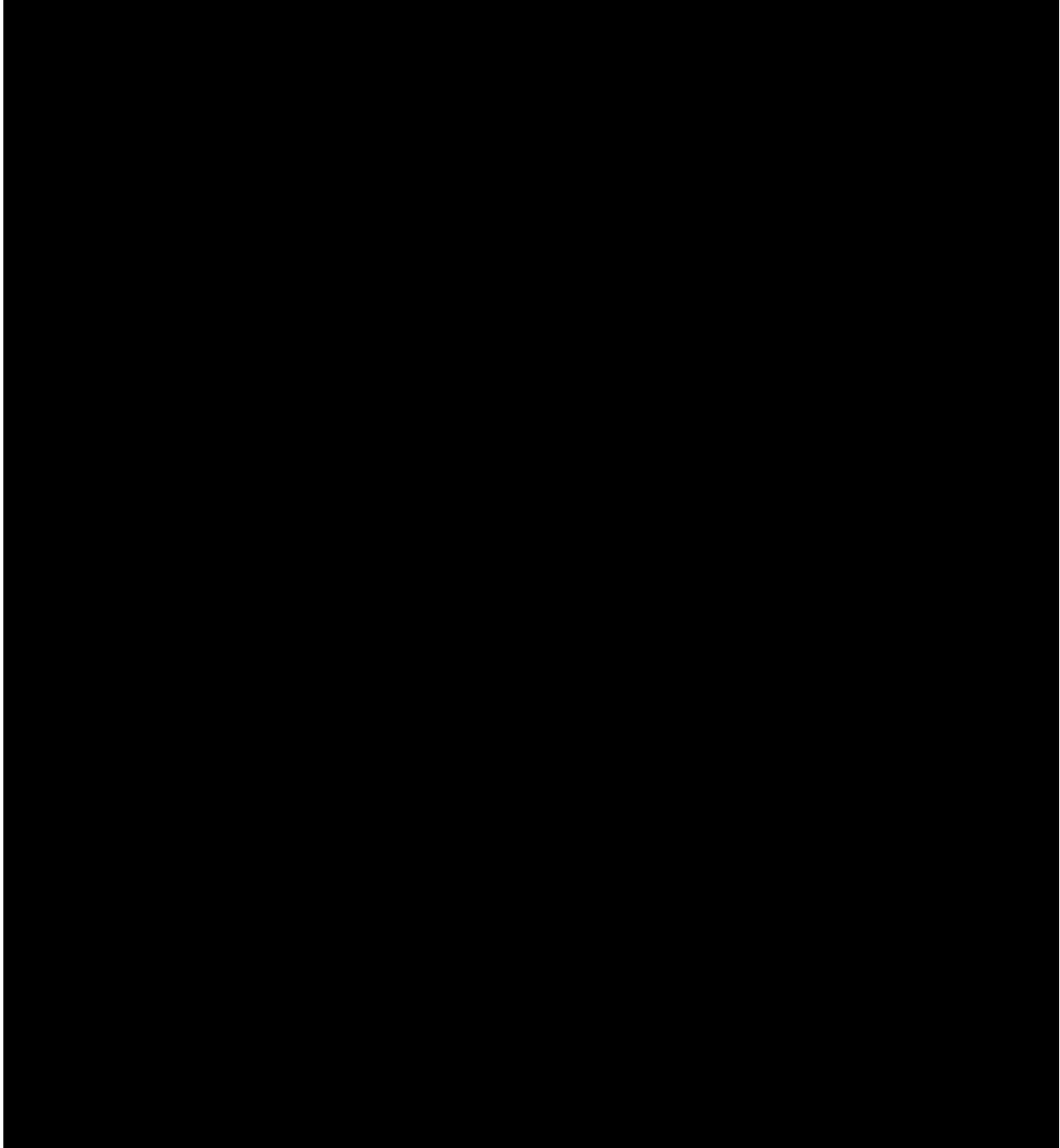


Figure 16: High-Level Network Architecture Diagram

4.1.3.2. VA Intranet Zone

The VAAFI solution is complex and involves many different types of resources. In requesting access or response outcome of a request for a particular application and/or data, many VAAFI resources will be transparent to the end user. These resources are owned and operated by VAAFI members including VA, participating CSPs, and other entities such as the DoD and Office of Government Policy (OGP). The security and maintenance of these protected resources or components falls under the umbrella of the owner organization.

Multiple devices or systems that VA owns and are participating in VAAFI are protected resources. Protected resources can also be web-based communication channels, using the HTTP/HTTPS protocol. The following are web resources that VA provides to the end user or the other members of VAAFI:

- **Agency Applications (AAs):** AAs are protected resources for VAAFI. VAAFI's configuration filters any request for a VAAFI-enabled AA and redirects the request to an appropriate CSP login page, where the user can authenticate or select a link to choose a CSP and authenticate. The WebSEAL subsystem configuration filters any HTTP/HTTPS request and processes the needed authentication and authorization mechanisms. Access to the protected application is only allowed after the request has been authenticated and authorized. HTTPS (HTTP over SSL/TLS) is the only protocol accepted for access to the Agency Applications. Similar to how WebSEAL provides client security for AAs, the DataPower device can secure client web service requests to AAs and support AA requirements for portal to portal tunnel and user-level security.
- **VA Partner Connections:** These connections are where VA has established connectivity to partners that memorandums of understanding (MOUs) govern. The DataPower devices are used to enforce security compliance with those MOUs.
- **IAM Services:** These connections are where IAM services such as Provisioning and the Virtual Directory Service reside. These services are called from the VAAFI SP per the Target Portal Strategy flow and from AccessVA per the Third Party Onboarding flow.

4.1.3.3. VAAFI Internal Zone

The Internal Zone is a restricted zone supporting operations that require very strict control. Communication transport is very closely managed between this zone and other boundary zones.

The following are VAAFI services in the Internal Zone:

- **Security services:**
 - LDAP User repository: IBM Tivoli Directory Server
 - Policy Manager: IBM Tivoli Access Manager for e-business
 - ITFIM
 - SysLogging
 - PKI CSP Registration Application
- **User Repository services:**
 - User repository: IBM Tivoli Directory Server V6.0

- Database server: IBM DB2 Universal Database, V8.2 Enterprise Server. Both of these services reside on the LDAP servers depicted in the diagram.
- Application Services: WebLogic Applications Servers for AccessVA

4.1.3.4. VAAFI Demilitarized Zone (DMZ)

The DMZ is a controlled zone that contains resources needed to provide controlled access. The DMZ zone is typically between two firewalls, and has managed and tightly controlled inbound as well as outbound traffic.

The classification of transport between a controlled and an uncontrolled zone is “public.” The classification of transport between a controlled and another controlled or restricted zone is “managed.”

The following are VAAFI network services in the DMZ:

- The Enterprise Cyber Security Infrastructure Project (ECSIP) Gateways provide the external firewall for the VA network and define the interface between the internet and DMZ zones. The ECSIP gateways perform packet filtering and rule-based firewall functions;
- Load balancers: F5 Big Internet Protocol (IP) device;
- Reverse proxy: IBM Tivoli Access Manager V6.1 for e-business (WebSEAL component);
- IBM WebSphere DataPower appliances for web services security and xml threat protection; and
- Apache HTTP Servers for AccessVA.

4.1.3.5. Internet Zone

The Internet is a global network (a network of networks) connecting millions of computers. Since no one entity controls the communication passing through the Internet Zone, all non-public information/traffic should be encrypted. Publicly available web sites and portals that provide VAAFI information and links are located here.

Non-VA Partner Connections are where the VA has established connectivity to partners that MOUs govern. The DataPower devices enforce security compliance with those MOUs.

4.2. Hardware Architecture

4.2.1. At the hardware level, VAAFI consists of servers built on VMware ESX 4.1 virtual machines, DataPower appliances, F5 Big IP Load Balancers, and Cisco firewalls and switches. The system firewalls divide this hardware into three network zones: the internal network, the DMZ, and the VA WAN. A further zone exists at the VA level where the Gateways provide access to the Internet, as Section 4.1.3 described. F5 Big IP Load Balancers

VAAFI uses F5 Big IP 6400 devices with 4GM GTM modules to load balance traffic for specific services within VAAFI. Specifically, the following addresses are cluster addresses within the VAAFI SP: eauth.va.gov and services.eauth.va.gov. VAAFI also uses the F5 devices to load balance the Syslog traffic from the DataPower devices to the syslog servers.

4.2.1.1. Internal Interfaces

The internal interface distributes inbound requests (at the external interface) to the registered cluster members within the local LAN. The F5 interfaces with the WebSEAL subsystem to load balance client requests to the front-end servers for eauth.va.gov traffic. The device load balances traffic across the Production WebSEAL servers. The current configuration is for the F5s to send a GET/HTTP 1.1 request every 15 seconds on port 443. For additional information regarding WebSEAL, see Section 6.2.6.13.

These devices also provide load balancing for the services.eauth.va.gov traffic; however, they load balance traffic to the two DataPower devices in Production and the two devices at the Contingency Site, totaling four DataPower devices. The DataPower devices receive a GET/monitor signal every 15 seconds.

In Table 26: F5 Terremark (Both Sites), “Least busy” indicates that the load balancer selects the node with the least connections to receive the next connection request.

Table 26: F5 Terremark (Both Sites)

Cluster	Sending To	# Members in Cluster	Heartbeat	Notes
eauth.va.gov	SP WebSEAL Servers	12		Least busy
pki.eauth.va.gov	PKI CSP WebSEAL Servers	2		Least busy

Cluster	Sending To	# Members in Cluster	Heartbeat	Notes
register.eauth.va.gov	PKI CSP IHS	2		Least busy
services.eauth.va.gov	DataPower Devices	2		Least busy
access.va.gov	AccessVA HTTP Servers	2		Least busy

Table 27: F5 Terremark (VA Only)

Cluster	Sending To	# Members in Cluster	Heartbeat	Notes
preprod.eauth.va.gov	SP WebSEAL Servers	12		Least busy
preprod.pki.eauth.va.gov	PKI CSP WebSEAL Servers	2		Least busy
preprod.register.eauth.va.gov	PKI CSP IHS	2		Least busy
preprod.services.eauth.va.gov	DataPower Devices	2		Least busy
preprod.access.va.gov	AccessVA HTTP Servers	2		Least busy

Table 28: F5 AITC SQA

Cluster	Sending To	# Members in Cluster	Heartbeat	Notes
sqa.eauth.va.gov	SP WebSEAL Servers	4		Least busy
sqa.pki.eauth.va.gov	PKI CSP WebSEAL Servers	2		Least busy
sqa.register.eauth.va.gov	PKI CSP IHS	2		Least busy
sqa.services.eauth.va.gov	DataPower Devices	2		Least busy
sqa.access.va.gov	AccessVA HTTP Servers	2		Least busy

Table 29: F5 AITC Lab

Cluster	Sending To	# Members in Cluster	Heartbeat	Notes
int.eauth.va.gov	SP WebSEAL Servers	4		Least busy
int.pki.eauth.va.gov	PKI CSP WebSEAL Servers	2		Least busy
int.register.eauth.va.gov	PKI CSP IHS	2		Least busy
int.services.eauth.va.gov	DataPower Devices	2		Least busy

Cluster	Sending To	# Members in Cluster	Heartbeat	Notes
pint.access.va.gov	AccessVA HTTP Servers	2		Least busy
pint.eauth.va.gov	SP WebSEAL Servers	1		Least busy
pint.pki.eauth.va.gov	PKI CSP WebSEAL Servers	1		Least busy
pint.register.eauth.va.gov	PKI CSP IHS	1		Least busy
pint.services.eauth.va.gov	DataPower Devices	1		Least busy
int.access.va.gov	AccessVA HTTP Servers	2		Least busy

4.2.1.2. External Interfaces

F5 has an IP-level based incoming interface configured to load balance any Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) client requests.

4.2.1.3. User Interface

The F5 Big IP device will be transparent to client requests. It includes a management console, accessible remotely via a network connection using HTTPS. Configuration can be via a Graphical User Interface (GUI) configuration application or a command line interface.

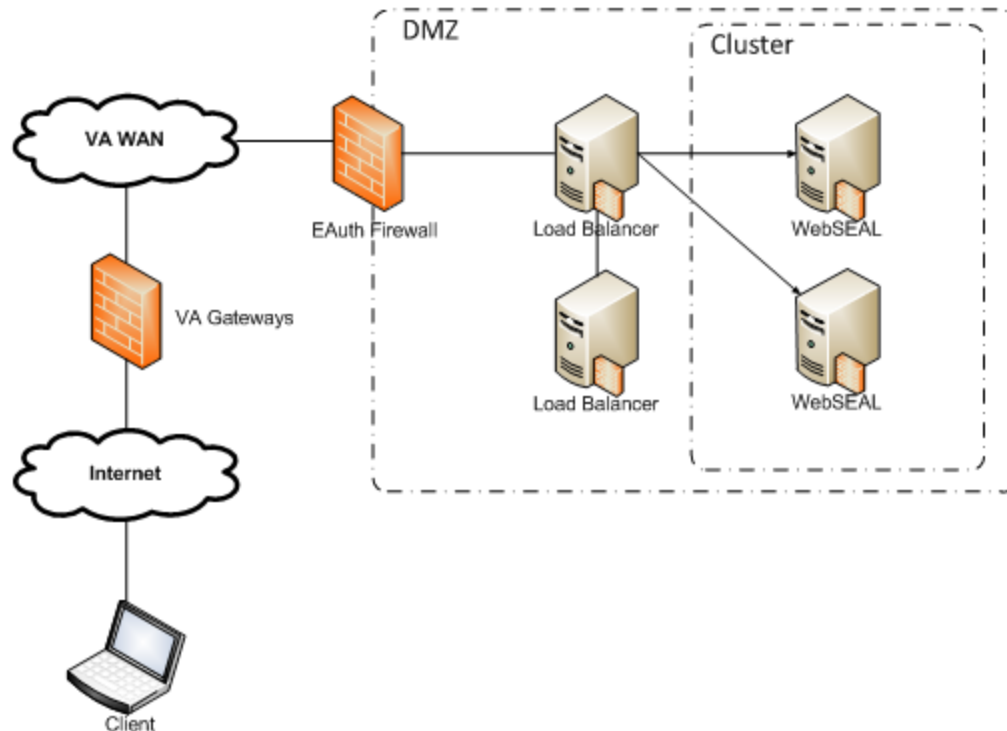


Figure 17: User Interface

The primary device functions as a load balancer and the other device remains in standby mode unless the primary device fails. The standby device uses a heartbeat to monitor the primary device's status. If the primary device fails, the standby device automatically switches to an active state, and assumes the responsibility of routing data packets.

4.2.2. DataPower

VAAFI uses the DataPower XI52 appliances to proxy web services within and outside of VA. Detailed information regarding these devices is in Section 6.3.

4.2.3. ESX VMware Farm

VAAFI resides on an ESX VMware farm. The VAAFI Installation and Configuration Guide and accompanying standard operating procedures (SOPs) assume VMware ESX version 4.1. The farm may be a single cluster or may be divided into an Internal farm and a DMZ farm per Terremark security requirements. There is connectivity between the components in the DMZ farm and Internal farm. The single cluster approach must have at least an initial 1 TB of disk space, 150 GB of RAM and X MHz of processor capacity. The Server Planning Worksheets provide the details regarding the virtual machine (VM) requirements for each environment.

4.2.4. Cisco Firewalls and Switches

Cisco firewalls and switches make up the physical hardware of the network. Section 4.1.3 contains details of the firewall configuration, along with diagrams of connectivity and ports. The detailed design information in Section 6 also describes port usage by individual component.

4.3. Software Architecture

The high-level software architecture consists of multiple tiers of components, as Figure 18 depicts. These tiers group into the following categories:

- A front-end user interface/portal tier;
- A middle tier that includes the proxy servers, policy servers, and other components; and
- A back-end tier that consists of user repositories and related components.

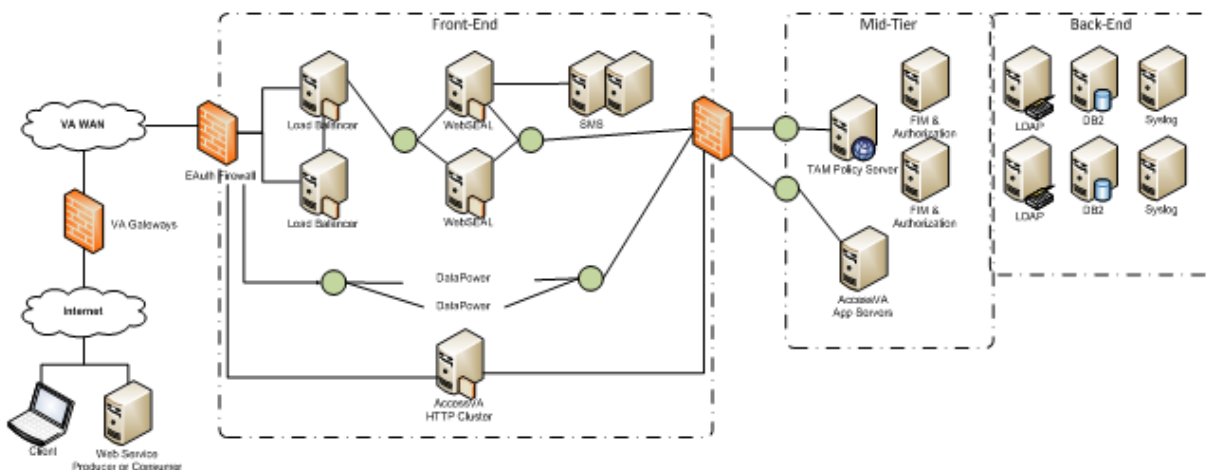


Figure 18: High-Level Software Architecture

The AccessVA software is servlet-based and consists of Spring MVC controllers and Mobile Responsive Bootstrap Framework that deploy on a Java 2 Platform, Enterprise Edition (J2EE) application server. The Model-View-Controller (MVC) layer has the ability to perform service calls to the AccessVA matrix layer and secure redirects to protected applications using the VAAFI infrastructure.

4.3.1. AccessVA

AccessVA is a web site that accepts both unauthenticated and authenticated traffic. VAAFI does not protect the unauthenticated traffic. The authenticated traffic uses a typical standard junction connection from VAAFI that protects AccessVA and passes it the standard Portal Strategy headers. AccessVA listens on Port 443 for unauthenticated traffic and supports no user attributes. VAAFI Protected traffic goes over port 444 using mutual SSL. The AccessVA Apache server configuration rejects any requests with headers on 443. AccessVA deploys as a single war file.

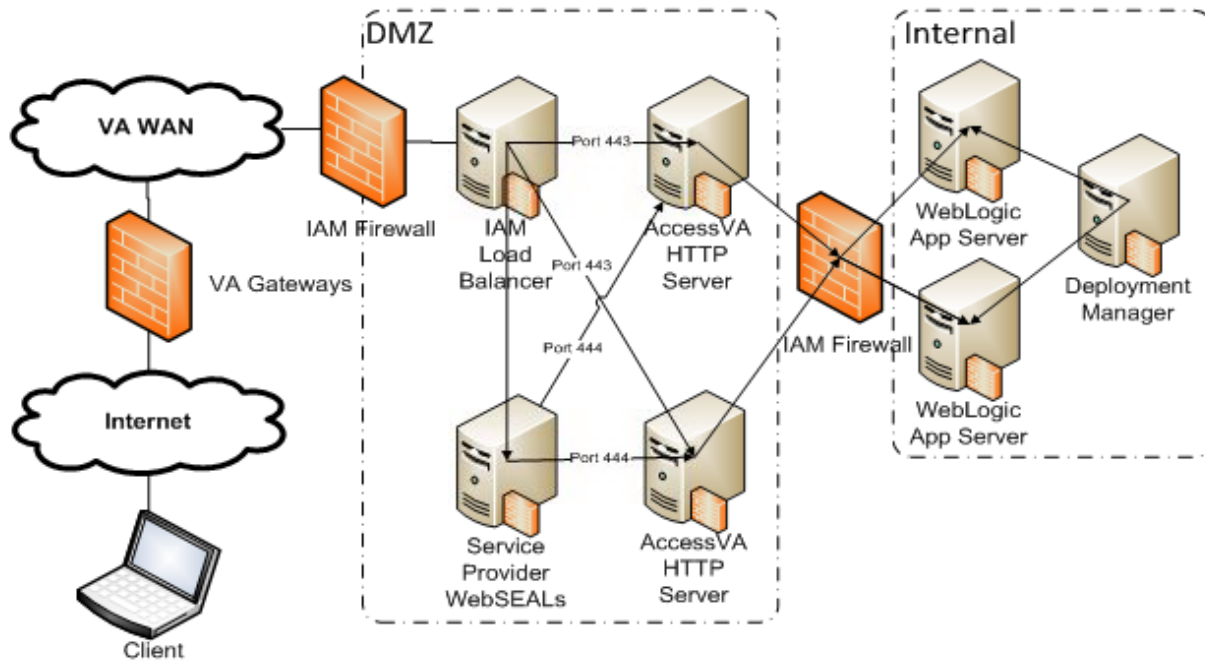


Figure 19: AccessVA

4.3.2. System Software Inventory

The table below lists the system software inventory.

Table 30: System Software Inventory

Vendor	Product	Version
IBM	Tivoli Directory Server	6.2
IBM	Tivoli Directory Client	6.2
IBM	DB2 Universal Database Enterprise Server Edition	9.5
IBM	GSKit	7.0.4.36
IBM	Tivoli Access Manager – Policy Server	6.1.1
IBM	Tivoli Access Manager – Authorization Server	6.1.1
IBM	Tivoli Access Manager – Runtime	6.1.1
IBM	Tivoli Access Manager – Java Runtime Environment	6.1.1
IBM	Tivoli Access Manager – Web Portal Manager	6.1.1
IBM	Tivoli Access Manager – SMS Command Line	6.1.1
IBM	Java Runtime Environment (JRE)	1.5.0
IBM	WebSphere Application Server Network Deployment	6.1.0.45
IBM	HTTP Server	6.1.0.45

Vendor	Product	Version
IBM	Session Management Server	6.1
IBM	WebSphere Extreme Scale	
IBM	Tivoli Federated Identity Manager	6.2.2.10+
IBM	Tivoli Access Manager WebSEAL	6.1.1
Oracle	Oracle WebLogic Server Enterprise Edition	12.12c
Apache		

4.3.3. Front-End

Front-end refers to the portion of the system closest to the end user. Front-end replication servers provide availability and fault tolerance.

- F5 Big IP Load Balancers
 - The system uses an active/stand-by failover scheme. This scheme uses the primary device as the active load balancer. The stand-by is intended for disaster recovery and primary device failure.
 - The primary F5 device load balances client requests to back-end resources, using the Round Robin Scheduling Algorithm.
- WebSEAL
 - The system ensures front-end scalability by load balancing the WebSEAL component. A replicated front-end WebSEAL server provides the site with load balancing during periods of high demand. The SMSs ensure that either WebSEAL can handle a session regardless of where the session started.
- DataPower
 - The system ensures front-end scalability by monitoring DataPower performance and scaling through the addition of more DataPower devices when necessary. A replicated front-end DataPower device, capable of handling the same load, provides the site with a standby device if the active (primary) should fail.
- Apache
 - The system ensures front-end scalability by load balancing the WebSEAL component. A replicated front-end WebSEAL server provides the site with load balancing during periods of high demand. The SMSs ensure that either WebSEAL can handle a session regardless of where the session started.

4.3.4. Mid-Tier

- IBM Tivoli Access Manager (TAM)
 - To attain an efficient failover and scalability schema, the primary IBM Tivoli Access Manager Policy Manager underlying directory is replicated to the WebSEAL servers. The TAM Policy Director is not necessary at runtime.

- The IBM TAM will be configured to continuously poll the local Lightweight Directory Access Protocol (LDAP) server. If a LDAP server does fail, IBM Tivoli Access Manager continuously polls the server to check for its return to active duty.
- IBM Tivoli Federated Identity Manager
 - Authorization service components are replicated across four servers to increase availability in a high-demand environment such as VAAFI environment. The Authorization servers are installed on the same servers as the Federated Identity Manager as per IBM's best practices.
- WebLogic
 - VA required J2EE application server set up in a cluster. This is a leading enterprise product.

4.3.5. Back-End

- IBM LDAP Directory
 - Tivoli Access Manager allows for primary and replica LDAP servers. The replica LDAP server can assume LDAP server operations if the primary LDAP server fails. The LDAP is built on the DB2 database infrastructure.
- Syslog Cluster
 - Splunk provides syslogging for the DataPower devices. Each environment has an assigned IP Address for system logging. The Culpeper environment forwards the DataPower logs for Culpeper to a server that caches the logs and forwards them to the Splunk server in Miami Florida.

4.4. Network Architecture

VAAFI uses TCP/IP protocol for all communications (see Figure 16, above).

AccessVA communicates with VAAFI-partnered CSPs and the VA applications such as ROES through secure communication protocols. VAAFI uses the SAML, which is relayed as HTTPS headers to consuming web applications such as AccessVA.

4.5. Service Oriented Architecture/ESS

4.5.1. Protected Reverse Proxy

Details will be provided at the end of Sprint 3.

4.5.2. Web Service Proxy Producer

A Web Proxy Producer is set up through the exchange of certificates between VAAFI and the consuming application. These certs are installed and referenced through a AAA file on the Data Powers. An endpoint is used to direct the the authentication to the consuming applications.

4.5.3. Web Service Proxy Consumer

A Web Proxy Consumer is set up through the exchange of certificates between VAAFI and the consuming application. These certs are installed and referenced through a AAA file on the Data Powers. An endpoint is used to direct the the authentication to the consuming applications.

4.5.4. Reassertion of SAML

A SAML reassertion is a secure method of exchanging data between a consuming application and VAAFI. The consuming application provides certificates as well as metadata for processing. VAAFI in return provides metadata that includes all header information to the application.

4.5.5. Credential Service Provider

A credential service provider is used by the user to pass information to VAAFI for authentication. A CSP is set up on the DataPowers.

4.5.6. OAuth

Details will be provided at the end of Sprint 6.

4.5.7. STS

Details will be provided at the end of sprint 6.

4.6. Enterprise Architecture

Please refer to the COTS Product Roadmap on the [AcS TSPR](#) site.

4.7. Sequence Diagrams

Because of the complexity of multiple traffic flows through VAAFI, the following sequence diagrams visually depict the sequences of actions that commonly occur, and which components complete the actions:

- Starting from AccessVA (Level of Assurance 1 or VA PIV Credential);
- Starting from AccessVA (Level 2+ non-VA PIV Credential)Starting from CSP (Level 2+ non-VA PIV Credential)Starting at AA (Level 2+ non-VA PIV Credential);
- Starting with a Bookmark through VAAFI to an AA (Level 2+ non-VA PIV Credential);
- Continuous Communication;
- Insufficient Assurance Level;
- Invalid CSP;
- Session Reset
- Protected Business Partner Web Service; and
- OAuth.

Each traffic flow use case diagram has details for SAML 1.1 Artifact Profile. The text also addresses Browser Post Profile. Each sequence diagram shows only one case where the credential assurance level is 1. However, similar cases exist for each that start at the CSP, at the AA, and with a bookmark. Each of these cases skips the call to VDS, per Target Portal Strategy requirements.

4.7.1. Starting from AccessVA (Level of Assurance 1 or VA PIV Credential)

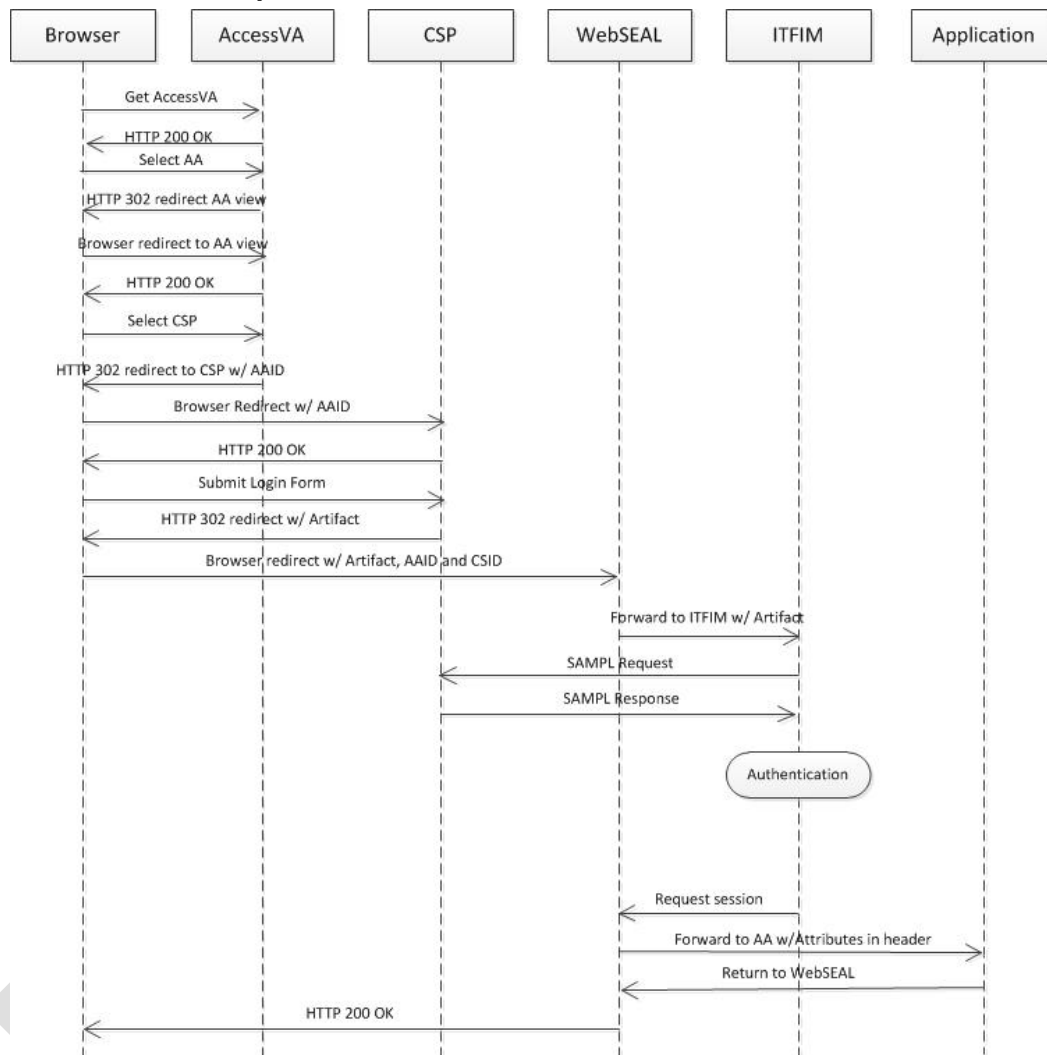


Figure 20: Starting from AccessVA (Level of Assurance 1 or VA PIV Credential)

1. User opens a browser to GET access.va.gov.
2. Browser displays AccessVA (HTTP 200 OK).
3. User selects AA from browser.
4. VA web site sends browser a 302 redirect to AA specific page.
5. Browser displays AA specific page.
6. User selects CSP.
7. VA web site sends browser a 302 redirect to CSP with AA Target in URL string.
8. User presented CSP login page (HTTP 200 OK).
9. User submits CSP credential.

10. Browser receives a 302 redirect with SAML artifact, AA Target, and CSid to WebSEAL when using SAML Artifact Profile. Using the Browser Post profile sends the SAML assertion rather than the artifact.
11. WebSEAL performs URL authorization and forwards to ITFIM with artifact or assertion.
12. ITFIM requests SAML assertion from CSP in the case of SAML Artifact profile. Using the Browser Post profile skips this action.
13. CSP responds with SAML assertion to ITFIM. Using the Browser Post Profile skips this action.
14. ITFIM validates the SAML response, authenticates the user, and recognizes the assurance level of 1 or VA PIV as the credential. (This sends the user through the legacy VAAFI flow rather than the Target Portal Strategy.)
15. ITFIM sends message to WebSEAL to create the user session.
16. WebSEAL forwards request to application with CSP-provided attributes in the header.
17. Application redirects page to WebSEAL.
18. WebSEAL redirects page to browser (HTTP 200 OK Display Page).

4.7.2. Starting from AccessVA (Level 2+ Credential)

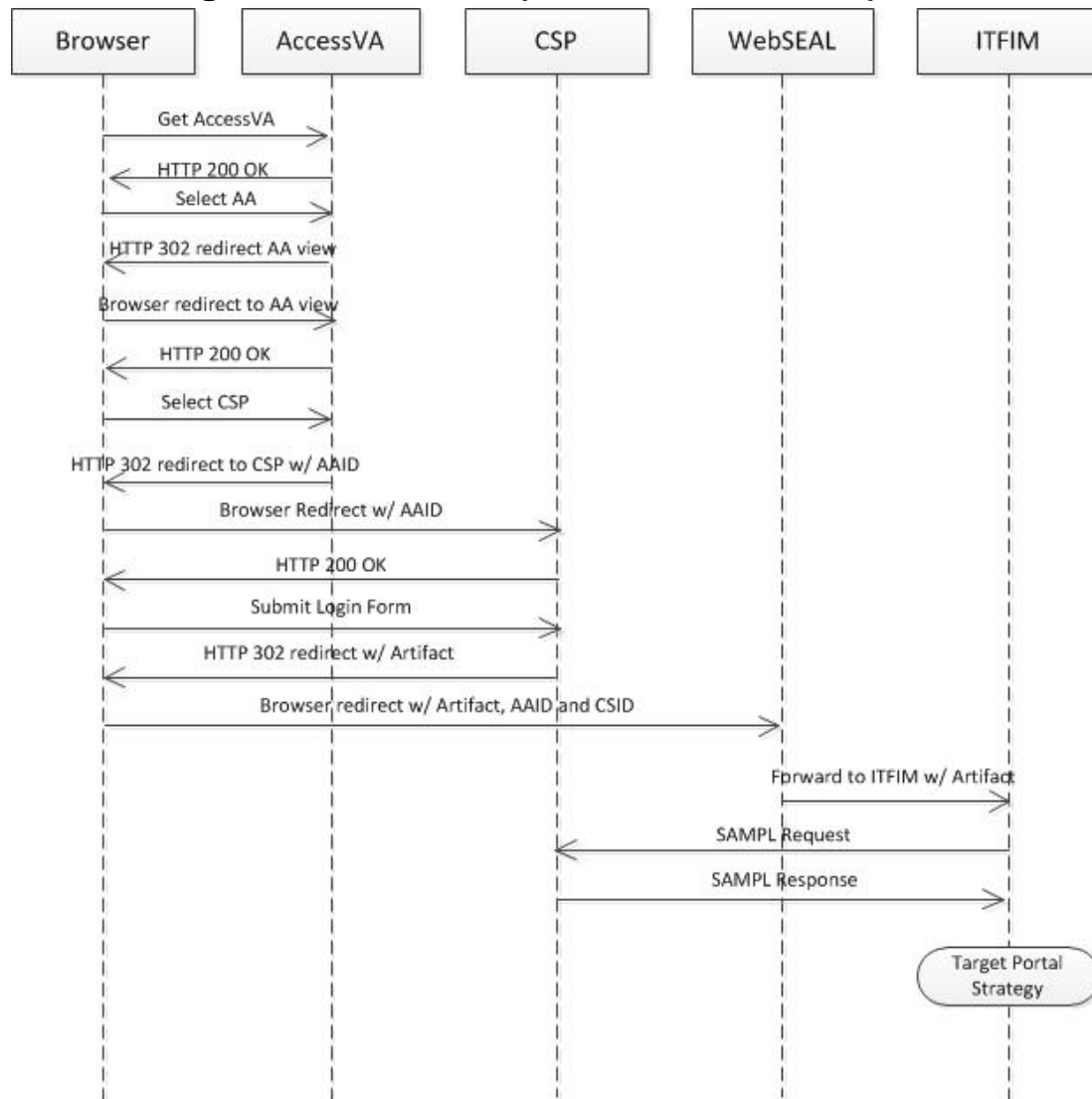


Figure 21: Starting from AccessVA (Level 2+ Credential)

1. User opens a browser to GET access.va.gov.
2. Browser displays AccessVA (HTTP 200 OK).
3. User selects AA from browser.
4. VA web site sends browser a 302 redirect to AA specific page.
5. Browser displays AA specific page.
6. User selects CSP.
7. VA web site sends browser a 302 redirect to CSP with AA Target in URL string.
8. User presented CSP login page (HTTP 200 OK).
9. User submits CSP credential.

10. Browser receives a 302 redirect with SAML artifact, AA Target, and CSid to WebSEAL when using SAML Artifact Profile. Using the Browser Post profile sends the SAML assertion rather than the artifact.
11. WebSEAL performs URL authorization and forwards to ITFIM with artifact or assertion.
12. ITFIM requests SAML assertion from CSP with the SAML Artifact profile. Using the Browser Post profile skips this action.
13. CSP responds with SAML assertion to ITFIM. Using the Browser Post Profile skips this action.
14. ITFIM validates the SAML Response and authenticates the user. From that point, the flow depends upon the response from the Portal Strategy Web Service. 6.2.6.11 describes this flow.

4.7.3. Starting from the CSP

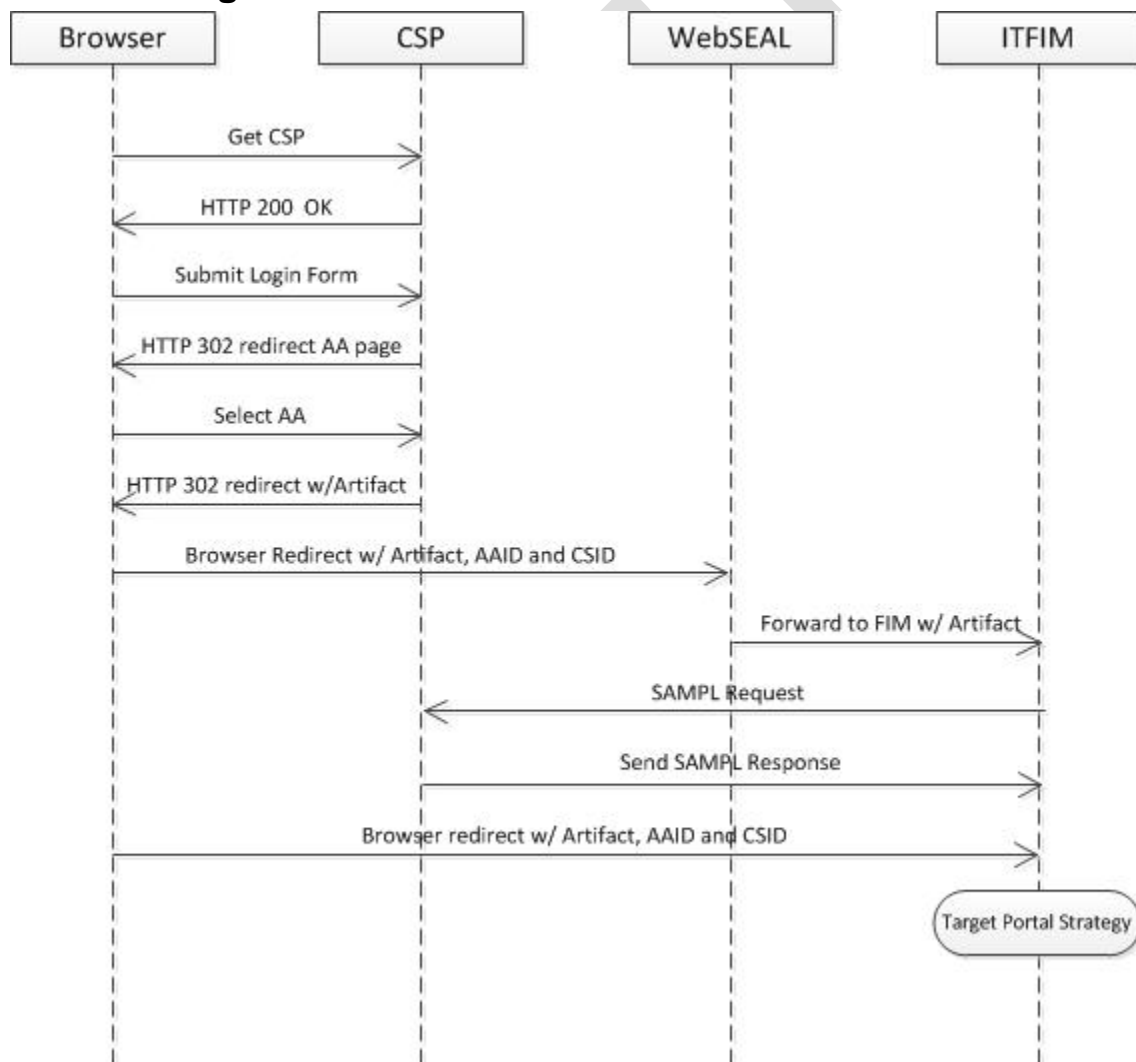


Figure 22: Starting from the CSP

1. User opens a browser to CSP Login page.

2. User presented CSP login page (HTTP 200 OK).
3. User submits CSP credential (Post Authenticate).
4. CSP displays Applications.
5. User selects AA from browser.
6. Browser receives a 302 redirect with SAML artifact, AA Target, and CSid to WebSEAL when using SAML Artifact Profile. Using the Browser Post profile sends the SAML assertion rather than the artifact.
7. WebSEAL performs URL authorization and forwards to ITFIM with artifact or assertion.
8. ITFIM requests SAML assertion from CSP in the case of SAML Artifact profile. Using the Browser Post profile skips this action.
9. CSP responds with SAML assertion to ITFIM. (Using the Browser Post profile skips this action). ITFIM validates the SAML response and authenticates the user. From this point, the flow depends upon the response from the Portal Strategy Web Service. Section 6.2.6.11 describes this flow.

4.7.4. Starting at the Agency Application

An end user may start the session from a link on a page that the AA (or some other entity) provides. These use cases could be similar to the use case for AccessVA where the user starts at the AA Specific View.

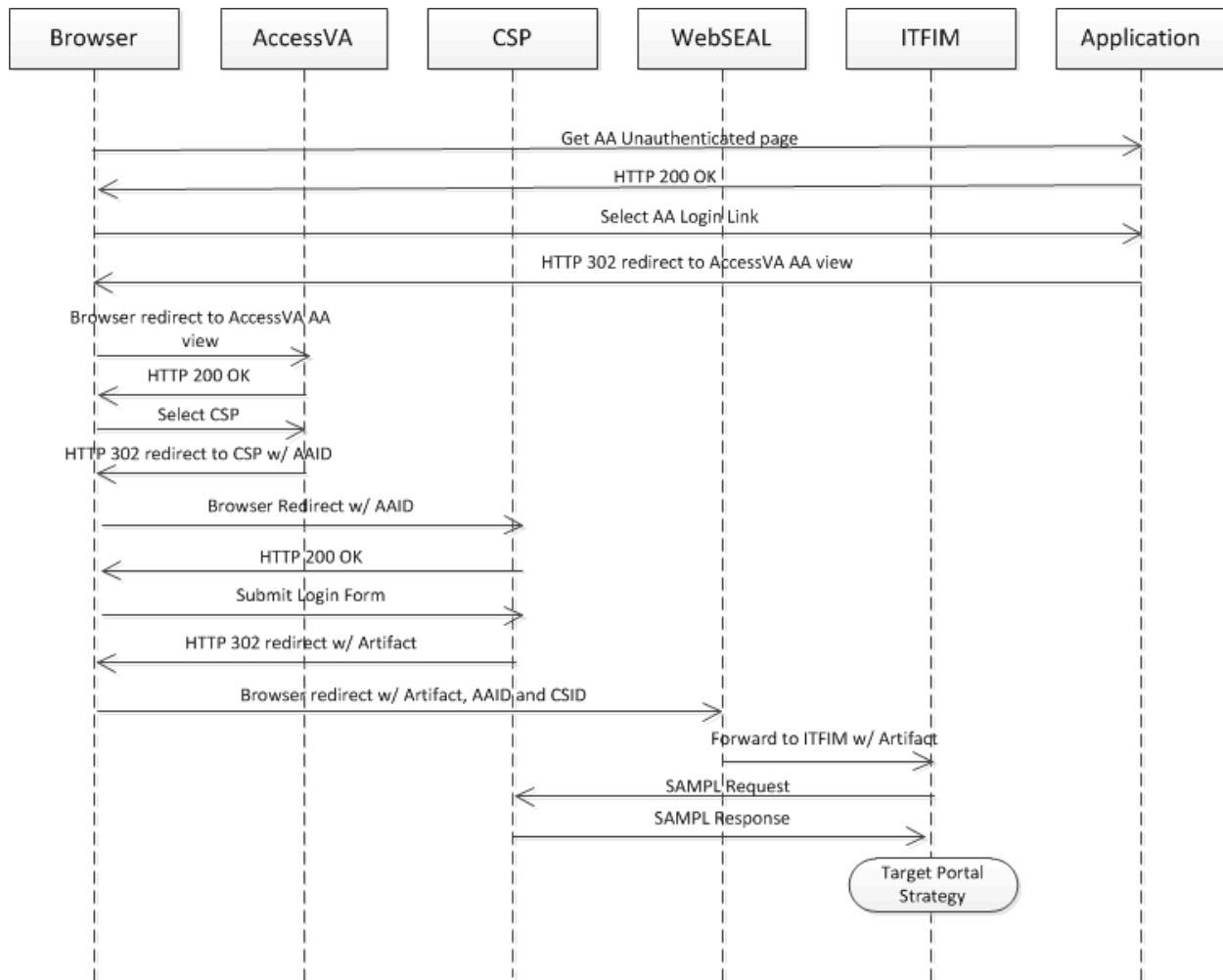


Figure 23: Starting at the Agency Application

1. User opens a browser to GET AA unauthenticated page.
2. Browser displays AA Unauthenticated page (HTTP 200 OK).
3. User selects Login link at AA web site from browser.
4. AA web site sends browser a 302 redirect to AccessVA AA specific view.
5. Browser displays AccessVA AA specific view.
6. User selects CSP.
7. VA web site sends browser a 302 redirect to CSP with AA Target in URL string.
8. User presented CSP login page (HTTP 200 OK).
9. User submits CSP credential.
10. Browser receives a 302 redirect with SAML artifact, AA Target, and CSid to WebSEAL when using SAML Artifact Profile. Using the Browser Post profile sends the SAML assertion rather than the artifact.
11. WebSEAL performs URL authorization and forwards to ITFIM with artifact or assertion.
12. ITFIM requests SAML assertion from CSP in the case of SAML Artifact profile. Using Browser Post profile skips this step.

13. CSP responds with SAML assertion to ITFIM. (Using Browser Post profile skips this step.)
14. ITFIM validates the SAML Response and authenticates the user. From this point, the flow depends upon the response from the Portal Strategy Web Service. Section 6.2.6.11 describes this flow.

4.7.5. Starting with a Bookmark through VAAFI to an AA

A new use case evolved during the project for a user who has previously authenticated to an AA bookmarks the landing page after reaching the AA. The following describes this sequence.

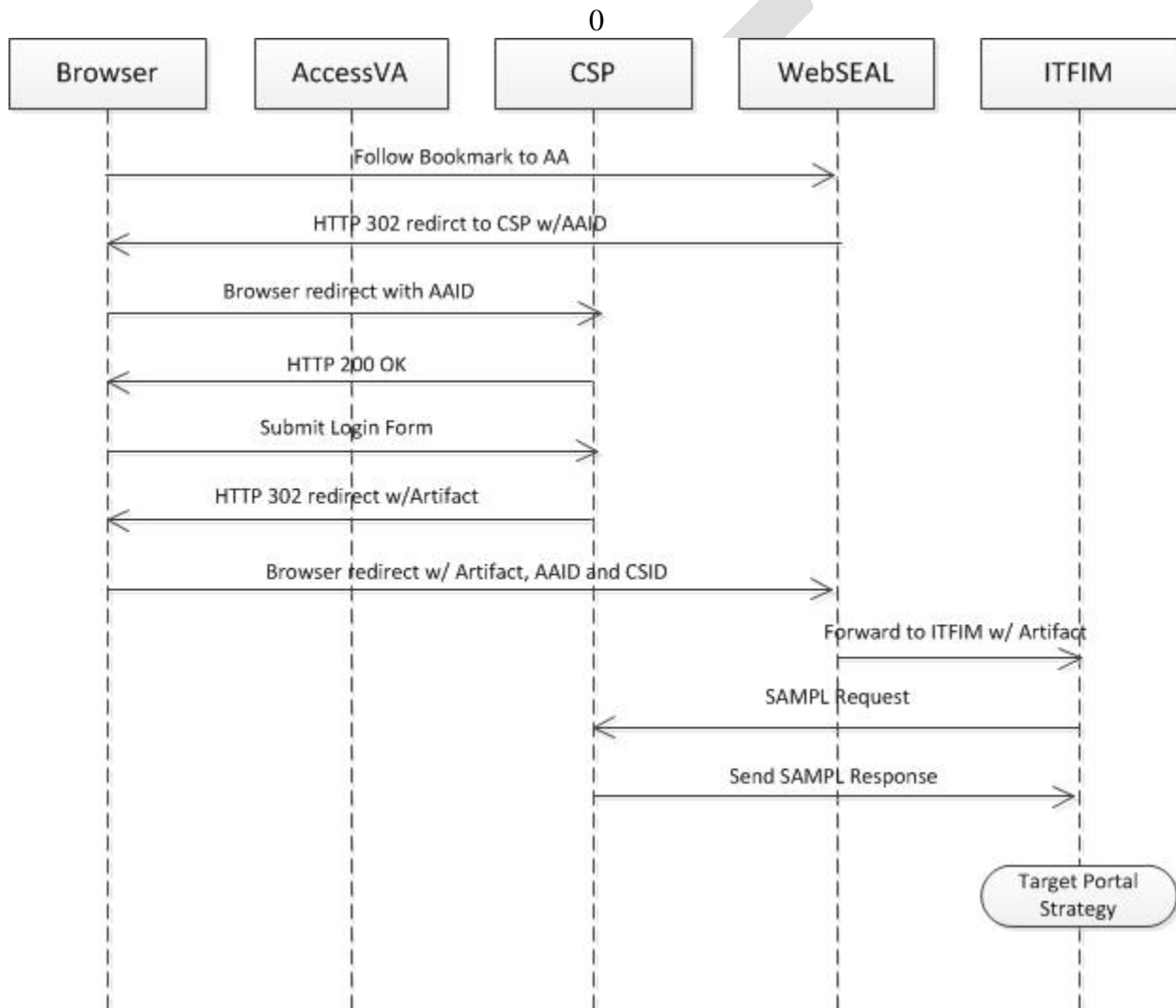


Figure 24: Starting with a Bookmark through VAAFI to an AA

1. User opens a browser and selects previously stored bookmark.
2. WebSEAL sends browser a 302 redirect to CSP with AA Target in URL string.
3. User presented CSP login page (HTTP 200 OK).
4. User submits CSP credential.

5. Browser receives a 302 redirect with SAML artifact, AA Target, and CSid to WebSEAL when using SAML Artifact Profile. Using the Browser Post profile sends the SAML assertion rather than the artifact.
6. WebSEAL performs URL authorization and forwards to ITFIM with artifact or assertion.
7. ITFIM requests SAML assertion from CSP in the case of SAML Artifact profile. Using the Browser Post profile skips this action.
8. CSP responds with SAML assertion to ITFIM (using the Browser Post Profile skips this action). ITFIM validates the SAML Response and authenticates the user. From this point, the flow depends upon the response from the Portal Strategy Web Service. Section 6.2.6.11 describes this flow.

4.7.6. Continuous Communication

Once any of the previous sequences have occurred, including any steps added by the Target Portal Strategy, communication continues to flow through WebSEAL.

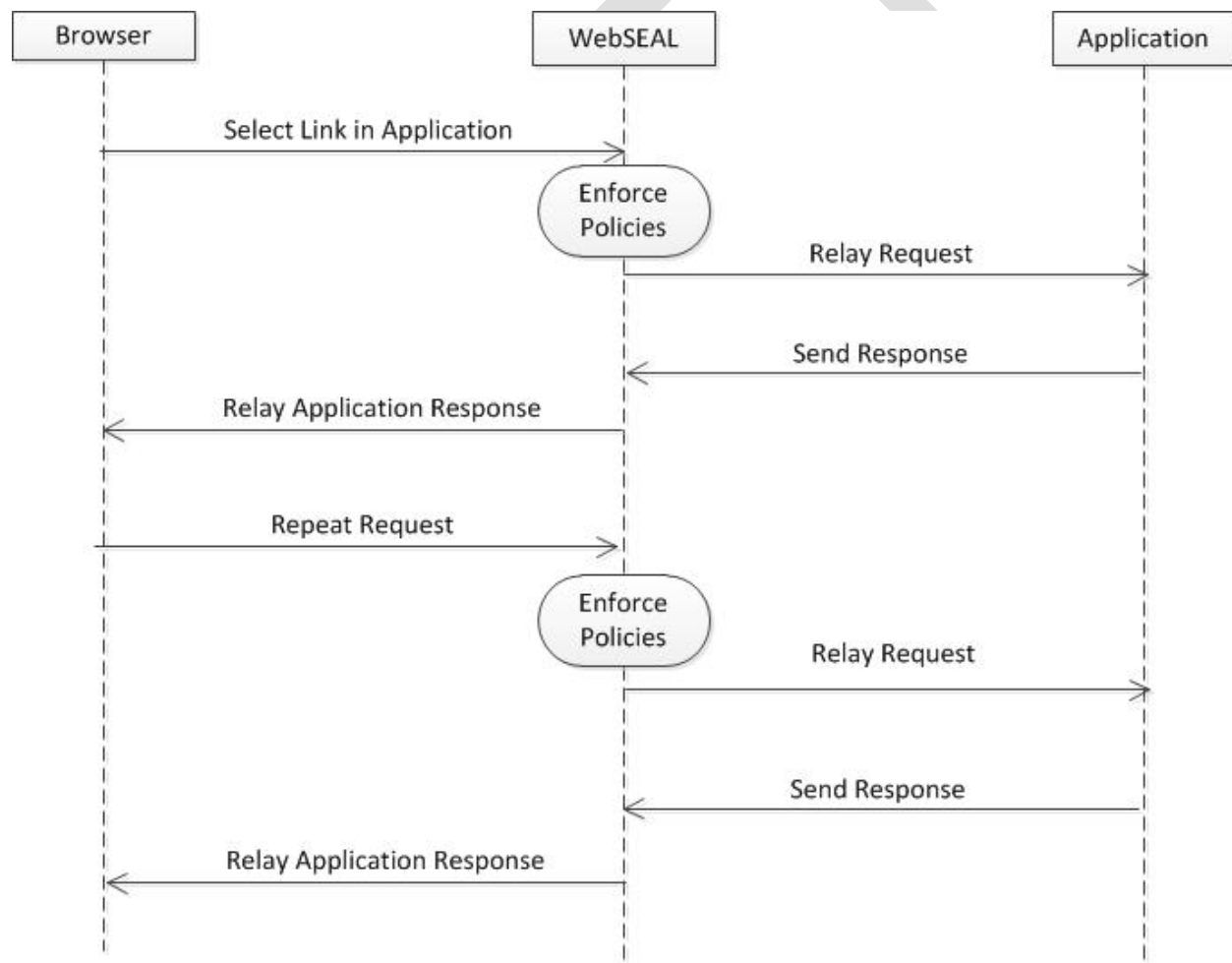


Figure 25: Continuous Communication

1. User is using the application and selects a link.

2. WebSEAL continues to function as a reverse proxy: it enforces policies and if the policies allow, relays the request from the user to the application.
3. When the application replies to the user's request, the reply is sent to WebSEAL.
4. WebSEAL relays the response from the application server back to the user.
5. This process repeats itself for every action the user takes in the application.
6. SAML 1.0 protocol does not describe a specified log out sequence of events at the end of a session. SAML 2.0 addresses this.

4.7.7. Insufficient Assurance Level

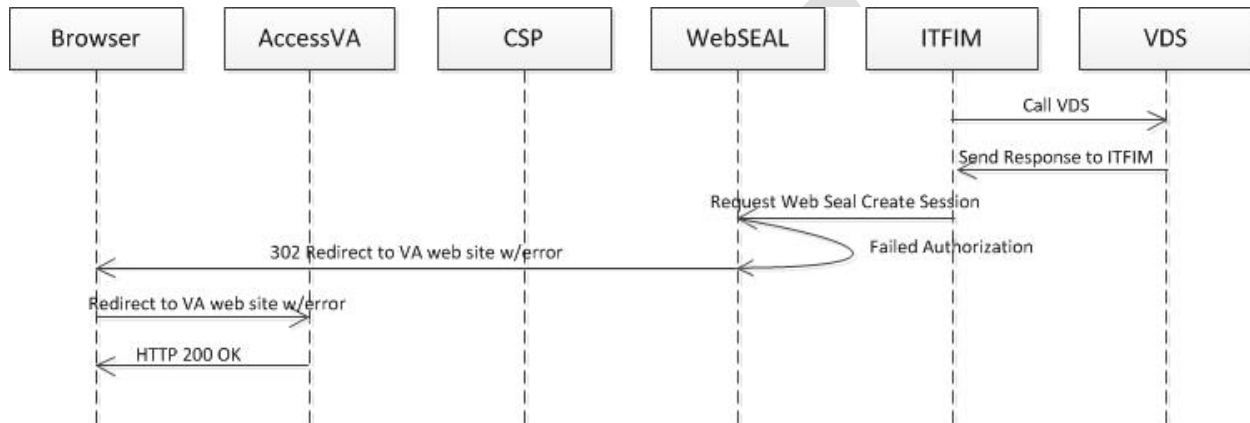


Figure 26: Insufficient Assurance Level

1. Target Portal Strategy directs the user to a target junction, but with a credential that does not meet the Assurance Level enforced by the junction's Authorization rule (see Section 6.2.2.6).
2. ITFIM sends message to WebSEAL to create the user session.
3. Application redirects page to WebSEAL and fails authorization.
4. WebSEAL sends the browser a 302 redirect to VA web site with error 50. Browser follows redirect to the VA web site error 50 page.
5. Browser displays the VA web site error 50 page (HTTP 200 OK).

4.7.8. Invalid

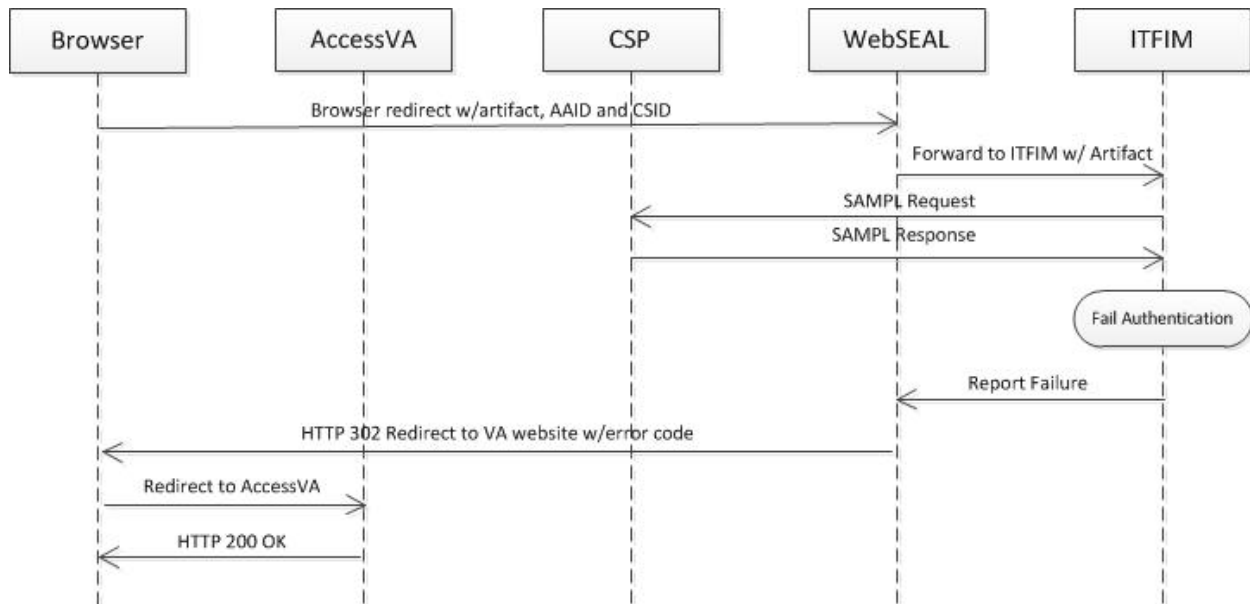


Figure 27: Invalid CSP

1. Browser receives a 302 redirect with SAML artifact, AAid, and CSid to WebSEAL when using SAML Artifact Profile. Using the Browser Post profile sends the SAML assertion rather than the artifact.
2. WebSEAL performs URL authorization and forwards to ITFIM with artifact or assertion.
3. ITFIM requests SAML assertion from CSP in the case of SAML Artifact profile. Using Browser Post profile skips this action.
4. CSP responds with SAML assertion to ITFIM. (Using Browser Post Profile skips this action)
5. ITFIM reports failure to WebSEAL.
6. WebSEAL sends browser a 302 redirect to the AccessVA web site with error code 70.
7. Browser follows redirect to the AccessVA web site error 70 page.
8. Browser displays the VAAFI web site (HTTP 200 OK).

4.7.9. Session Reset

The following flow could happen at any point during the Continuous Communication flow.

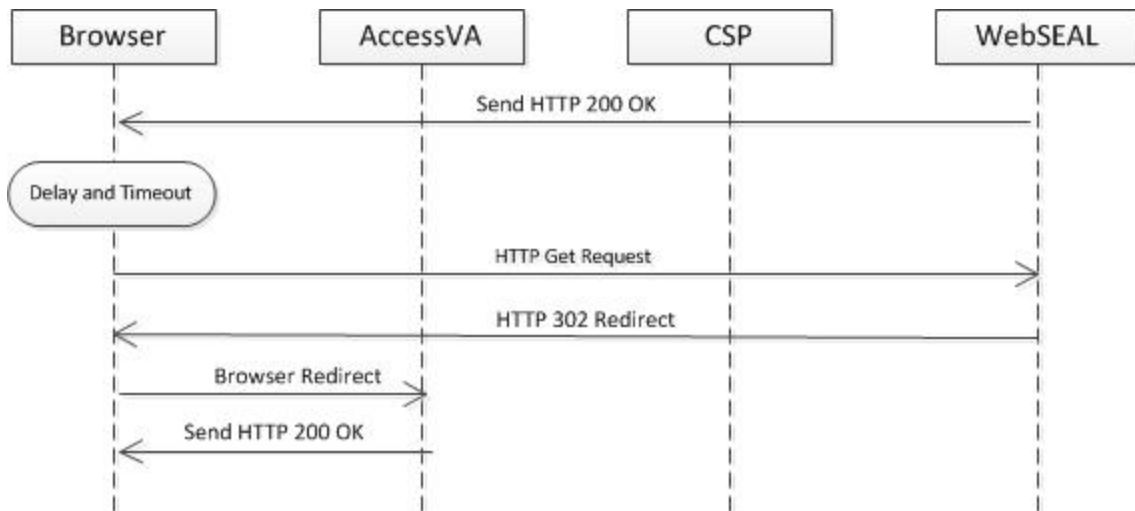


Figure 28: Session Reset

1. WebSEAL redirects page to browser (200 OK Display Page).
2. User remains idle for period of time.
3. User attempts to GET request for page.
4. Browser receives a 302 redirect to AccessVA with a session reset by WebSEAL once a 30-minute timeout period has expired.
5. Browser follows redirect to AccessVA.
6. Browser displays AccessVA unauthenticated page (HTTP 200 OK).

At this point, the user is back to the AccessVA web site and must select the application, the CSP, and re-authenticate. Depending on the specific application, the information from the previous session may be saved.

4.7.10. Protected Business Partner Web Service

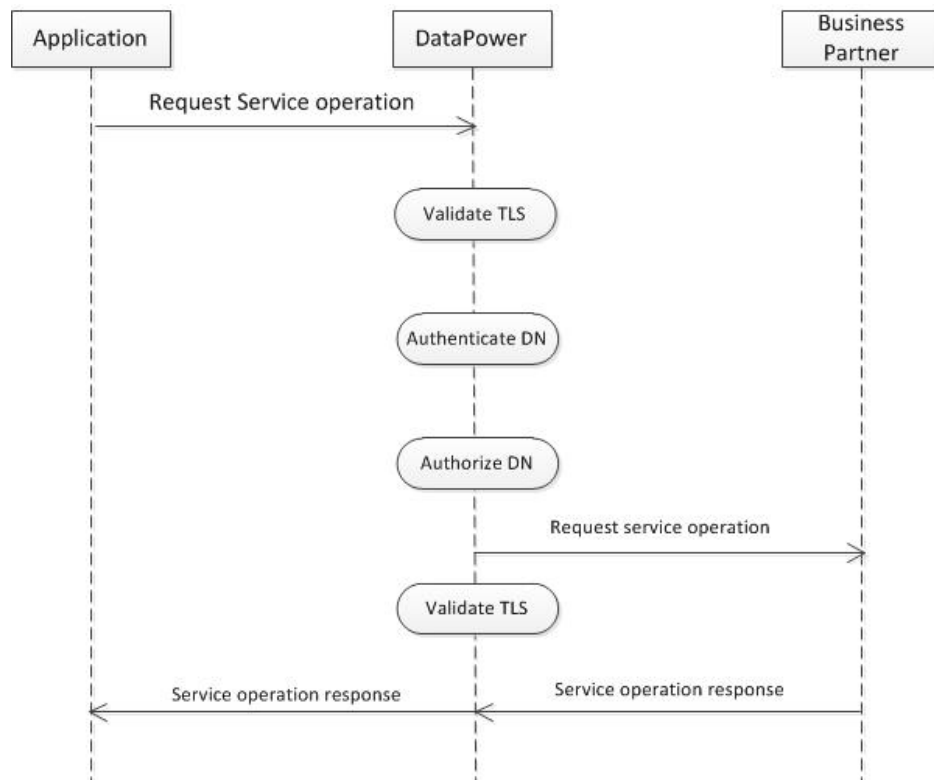


Figure 29: Protected Business Partner Web Service

1. Application server requests a protected web service operation using an HTTPS connection and providing its client certificate.
2. DataPower validates the TLS connection assuring that the client's provided certificate is trusted and valid.
3. DataPower checks the Distinguished Name (DN) from the validated application client certificate against the list of authentic users.
4. DataPower checks that the authenticated user is authorized for the requested resource.
5. DataPower makes a HTTPS connection to the "real" web service provider using the DataPower keys and certificates.
6. DataPower validates the certificate from the web service provider in the establishment of the TLS session.
7. DataPower receives the service operation response from the web service provider over the TLS session.
8. DataPower sends the service operation response to the application server over a TLS session.

4.7.11. OAuth Flows

4.7.11.1. OAuth Authorization Code Grant Flow

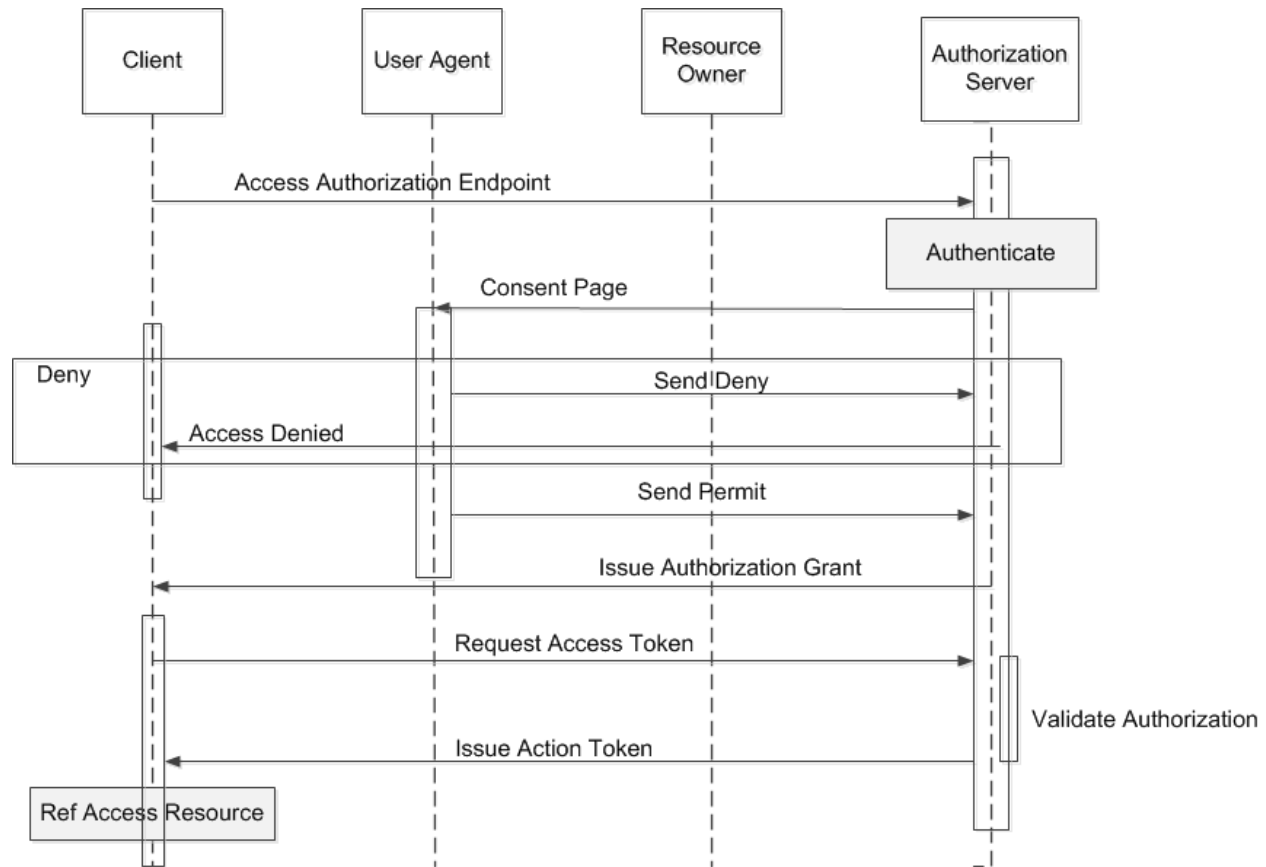


Figure 30: Authorization Code Grant Flow Sequence Diagram

The flow above includes the following steps:

1. The client initiates the flow by directing the resource owner's user-agent to the authorization endpoint. The client includes its client identifier, requested scope, local state, and re-direction URI to which the authorization server will send the user-agent back once access is granted (or denied).
2. The authorization server authenticates the resource owner (via the user-agent) and establishes whether the resource owner grants or denies the client's access request.
3. Assuming the resource owner grants access, the authorization server redirects the user-agent back to the client using the redirection URI provided earlier (in the request or during client registration). The redirection URI includes an authorization code and any local state provided by the client earlier.
4. The client requests an access token from the authorization server's token endpoint by including the authorization code received in the previous step. When making the request, the client authenticates with the authorization server. The client includes the redirection URI used to obtain the authorization code for verification.
5. The authorization server authenticates the client, validates the authorization code, and ensures that the redirection URI received matches the URI used to redirect the client in

step 3. If valid, the authorization server responds back with an access token and, optionally, a refresh token.

4.7.11.2. OAuth Implicit Grant Flow

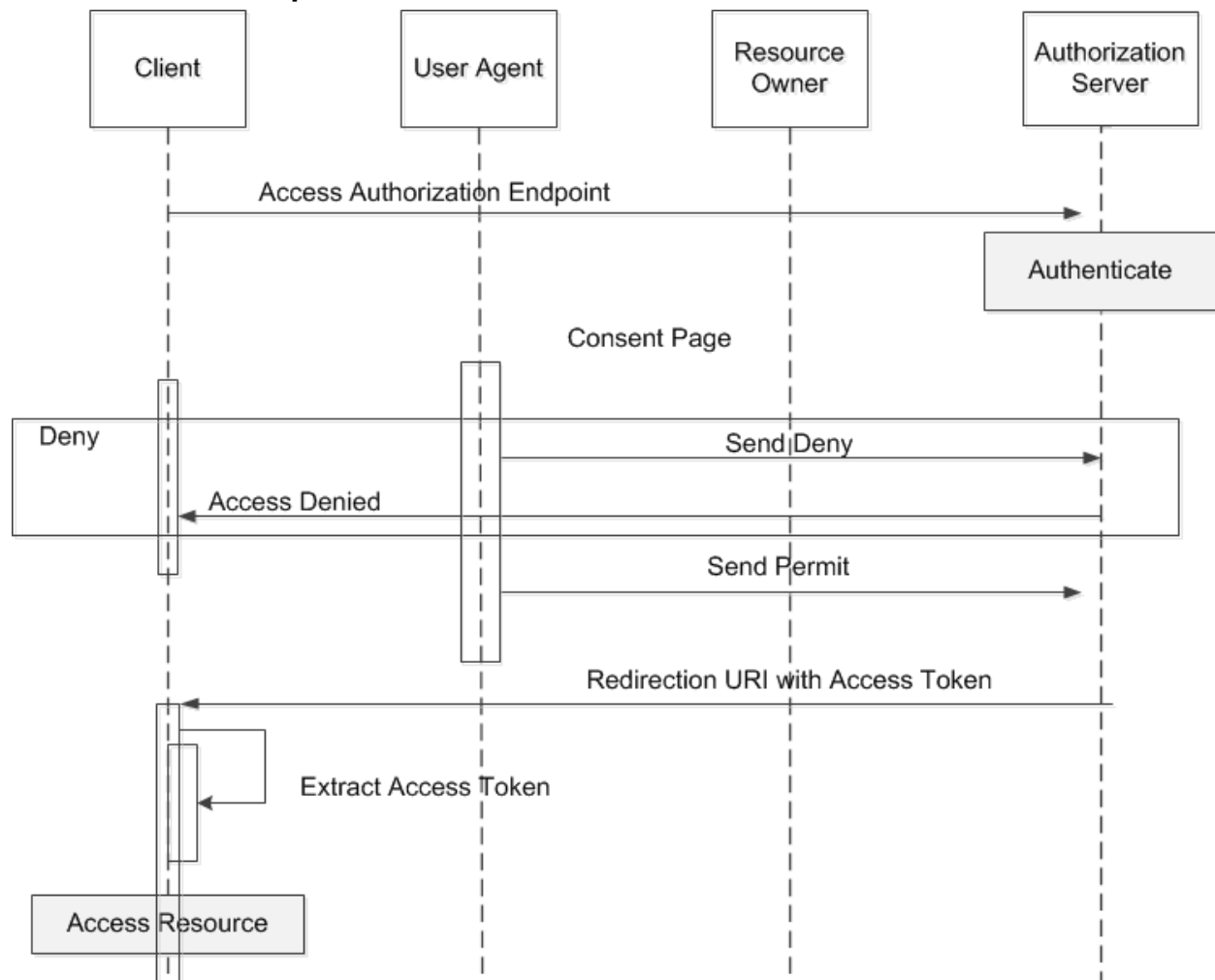


Figure 31: Implicit Grant Flow Sequence Diagram

The Implicit Grant flow, above, includes the following steps:

1. The client initiates the flow by directing the resource owner's user-agent to the authorization endpoint. The client includes its client identifier, requested scope, local state, and a redirection URI to which the authorization server will send the user-agent once access is granted (or denied.)
2. The authorization server authenticates the resource owner (via the user-agent) and establishes whether the resource owner grants or denies the client's access request.
3. If the resource owner grants access, the authorization server redirects the user-agent back to the client using the redirection URI provided earlier. The redirection URI includes the access token in the URI fragment.
4. The user-agent extracts the access token.
5. The user-agent passes the access token to the client.

4.7.11.3. OAuth Resource Owner Password Credentials Flow

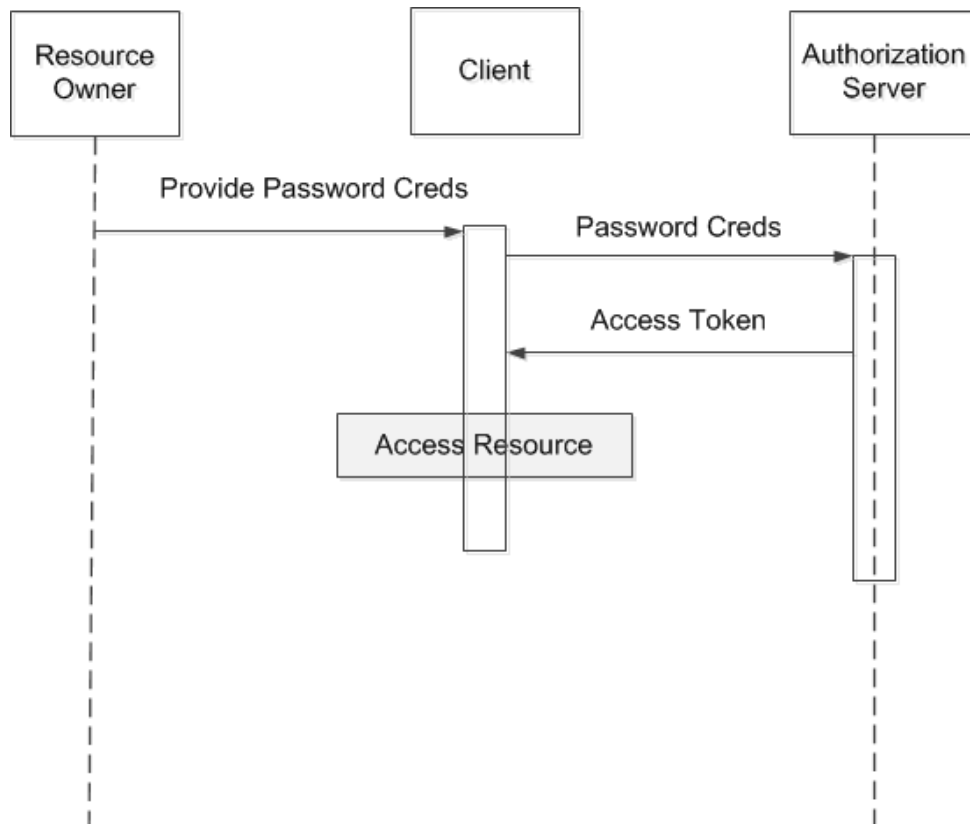


Figure 32: Resource Owner Password Credentials Sequence Diagram

The Resource Owner and Password Credentials flow, above, includes the following steps:

1. The resource owner provides the client with its username and password.
2. The client requests an access token from the authorization server's token endpoint by including the credentials received from the resource owner. When making the request, the client authenticates with the authorization server.
3. The authorization server authenticates the client and validates the resource owner credentials, and if valid, issues an access token.

4.7.11.4. Client Credentials Grant

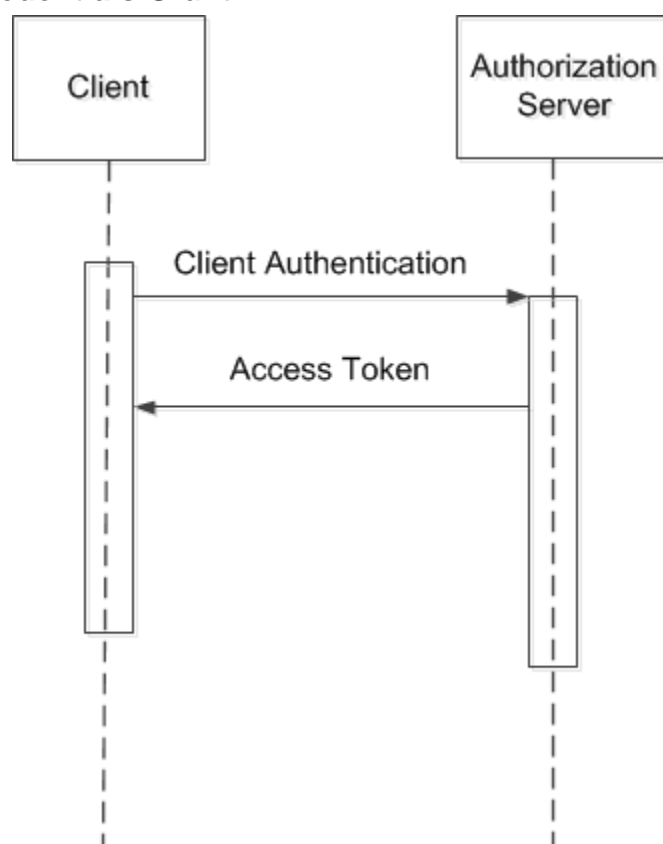


Figure 33: Client Credentials Grant Sequence Diagram

The Client Credentials Grant flow, above, includes the following steps:

1. The client authenticates with the authorization server and requests an access token from the token endpoint.
2. The authorization server authenticates the client, and if valid, issues an access token.

4.7.11.5. Session Data Injection Flow

This flow illustrates the mechanism in place to inject user session data into the OAuth flow. Current VAAFI federation solution provides a many-to-one user mapping solution, where all the users that an identity provider asserts will be mapped to a generic user within the VAAFI SSO domain. For instance, all DS Logon users will be identified as “dslogneauthuser” in the VAAFI SSO domain. As a result, all authorization grant and access tokens that the TFIM OAuth solution issues will be associated to a single user identifier in the Trusted Client data store. The number of modifications/customizations to the out-of-the-box TFIM OAuth solution are necessary to remediate this issue. These modifications/customizations inject user session data (unique identifier) into to Trusted Client data store to distinguish the users. Session Data injection is only applicable for the Authorization Code and Implicit Grant types.

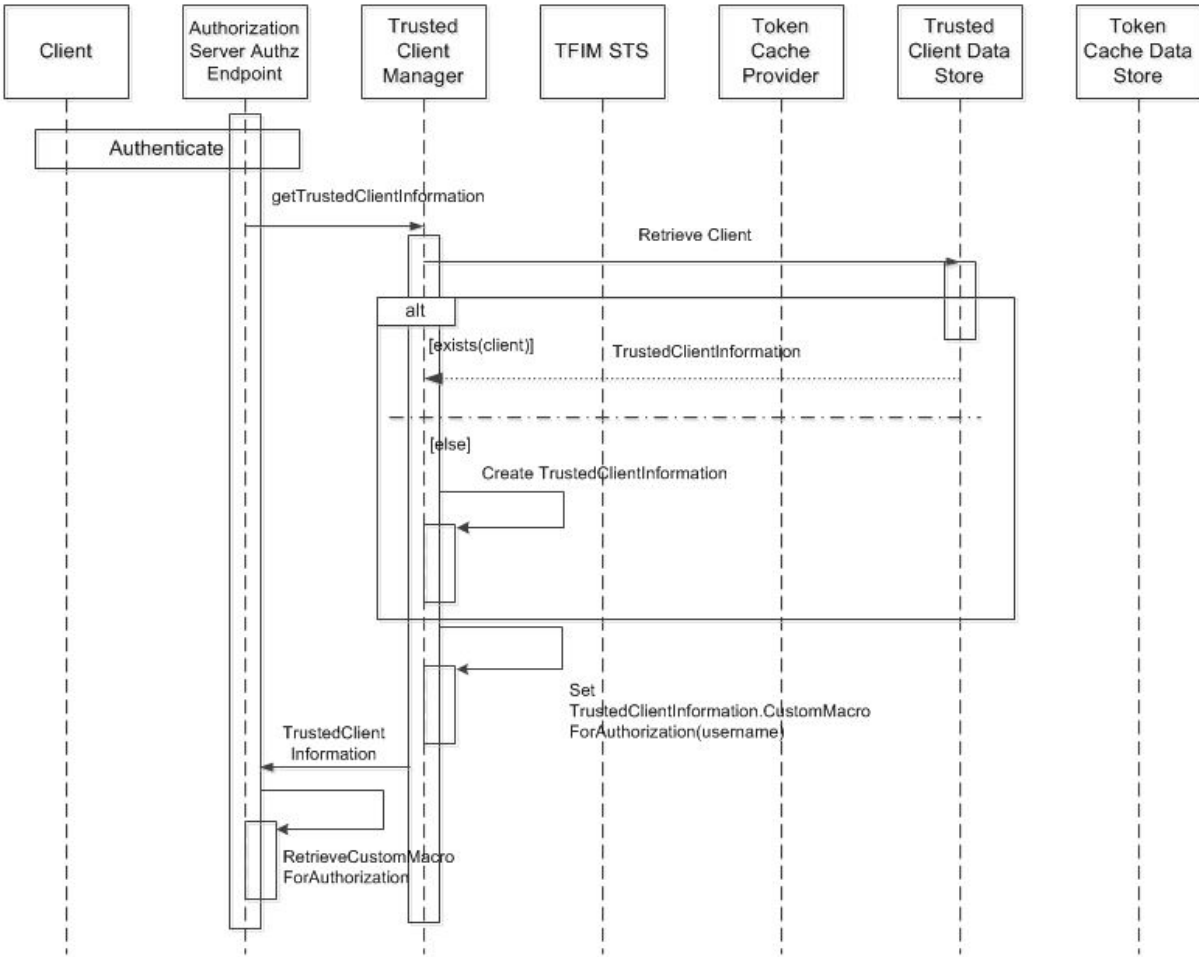


Figure 34: Session Data Interjection Flow (continued) (Pt. 1 of 3)

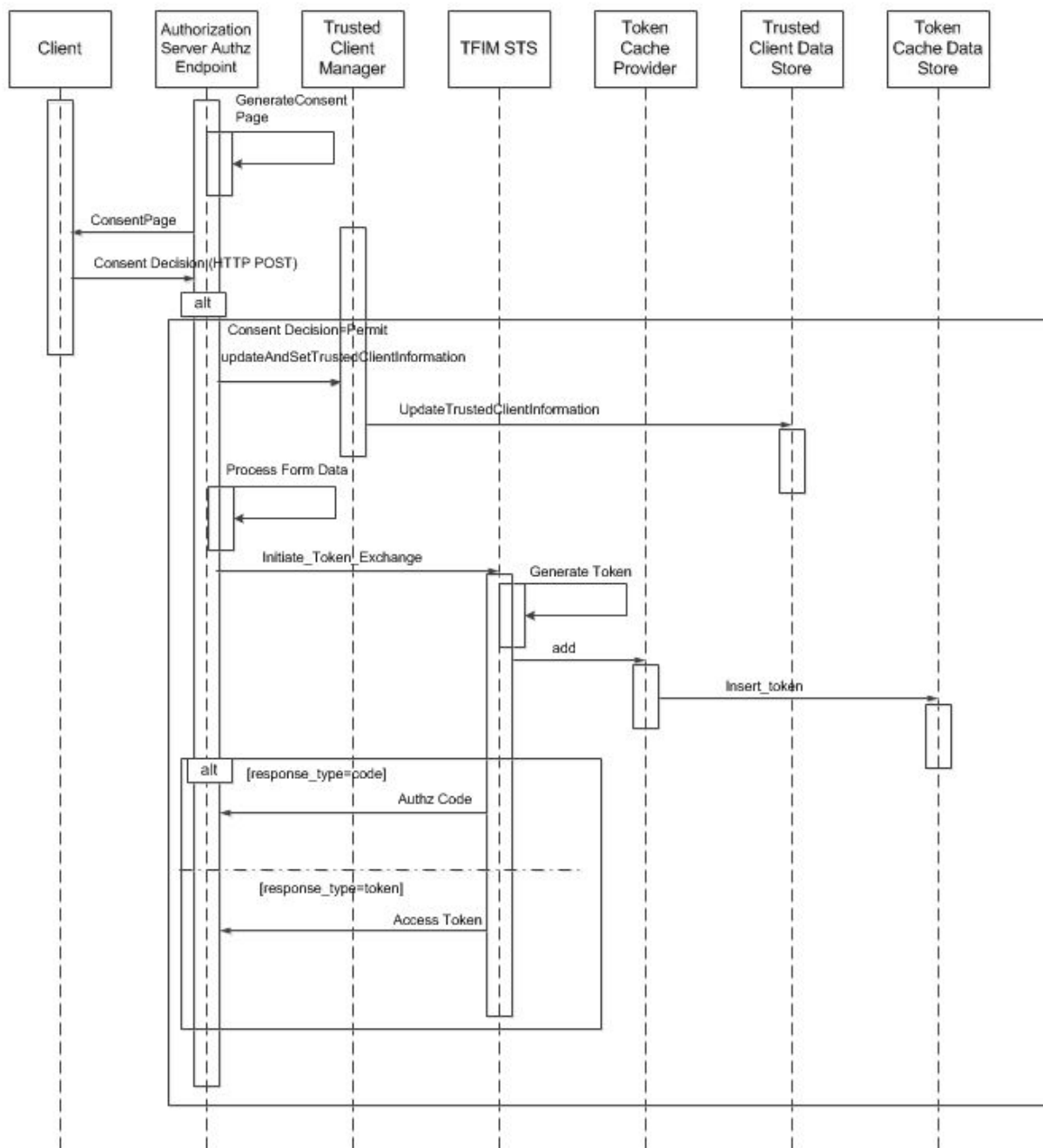


Figure 35: Session Data Interjection Flow (continued) (Pt. 2 of 3)

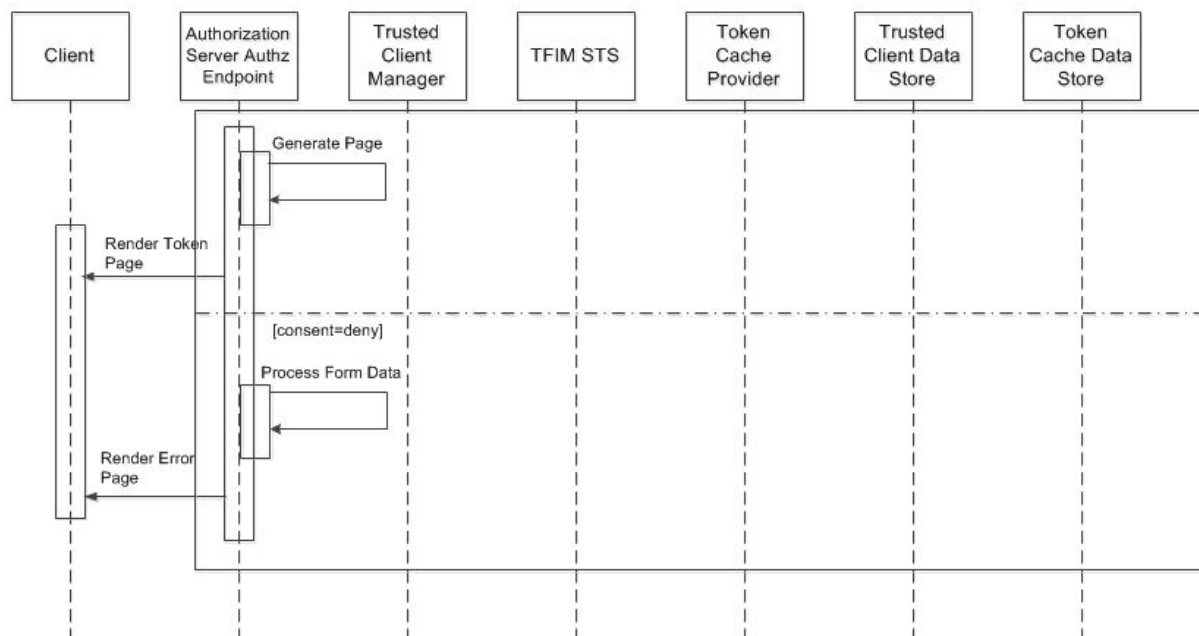


Figure 36: Session Data Interjection Flow (Pt. 3 of 3)

The Session Data Injection flow, Figure 34 through Figure 36, includes the following steps:

NOTE: Authorization Endpoint is configured to receive, in the form of a HTTP Header, the user identifier of the Resource owner that was used to authenticate with the Credential Service Provider

1. The Client initiates Authentication between the Resource Owner and the Authorization Server
2. The Client access the Authorization (Authz) endpoint, hosted by the Authorization Server, to obtain Resource Owner consent
3. Authz Endpoint invokes the Trusted Client Manager (TCM) to retrieve a Trusted Client entity that represents the Resource Owner. Authz Endpoint passes the HTTP Request object to TCM
4. TCM retrieves the Trusted Client Information (TCI) entity from the Trusted Client Data Store, if exists. Otherwise, TCM creates a new TCI entity. TCM extracts user identifier from the HTTP Request and assigns it as the username of the TCI entity
5. TCM sets the Custom Macro to the user identifier value extracted from the HTTP header in step 4.
6. TCM returns the TCI to the Authz Endpoint.
7. Authz Endpoint extracts the Custom Macro from the TCI.
8. Authz Endpoint generates the user_consent.html and populates the page with the Custom Macro retrieved from TCI.
9. Authz Endpoint renders the page to the Resource Owner. The user_consent page contains a HTML form that is submittable.

10. User provides his/her consent decision (Permit/Deny).
11. Authz Endpoint receives the consent decision. If the consent decision is a “permit”:
 - a. Authz Endpoint processes form data.
 - b. Authz Endpoint updates the TCI with the form data by invoking the TCM.
 - c. TCM updates the TCI and persists it to the Trusted Client Data Store.
 - d. Authz Endpoint initiates the token exchange by invoking the TFIM STS. The input to the TFIM STS is an STSUuniversalUser document which contains the forms data in addition to other internal data that Authz Endpoint inserts as per the product implementation.
 - e. FIM STS generates the token.
 - f. TFIM STS invokes the Token Cache Provider (TCP) to persist the generated token.
 - g. TCP inserts the token into the Token Cache.
 - h. TFIM STS returns the Authorization Code or the Access Token depending on the initial request type to the Authz Endpoint.
 - i. Authz Endpoint provides the Authorization Code/Access Token to the Resource Owner/Client.
12. If the consent decision is “deny,” the Authz Endpoint generates the error page and orders it to the Resource Owner/Client.

(The flow above shows the authorization code/access token being returned to the Resource Owner in the form of a HTML page. This is to simplify the illustration. This form of delivery is not typical.)

4.7.11.6. Token Validation Sequence Flows

The sequence diagram below illustrates the OAuth token validation process. TFIM OAuth validates both Authorization Grant and the Access Tokens (referred to as tokens) that is has issued. Authorization Grants are validated by the Authorization Server (only in the Authorization Code Grant) and the Access Tokens are validated by the PEP. Since the VAAFI OAuth solution uses an external database to store tokens, the token validation is performed against this Token Cache. TFIM STS service as the external interface for the token validation process. Components of Authorization Server and PEP act as STS clients.

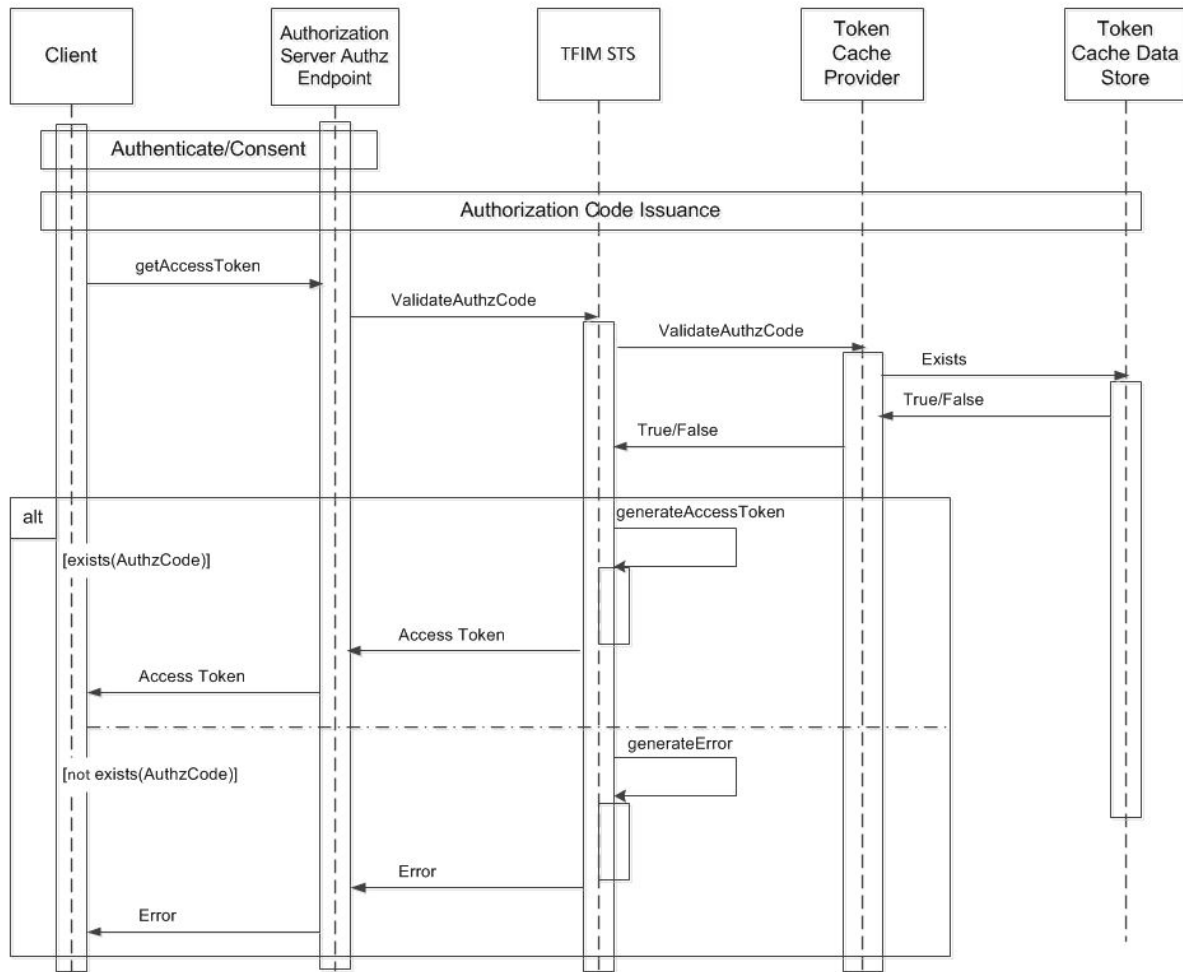


Figure 37: AuthZ Grant Validation

NOTE: This flow is only applicable for the Authorization Code Grant flow.

1. The Client initiates Authentication and Resource Access Consent between the Resource Owner and the Authorization Server.
2. The Client requests the Authorization Code.
3. Authorization Server generates and issues the Authorization Grant to the Client. The Authorization Code is stored in the Token Cache Data Store thru the Token Cache Provider.
4. The Client requests an Access Token by submitting the Authorization Code issues in Step 3.
5. Authorization Server sends a Token Validation Request to the TFIM STS.
6. TFIM STS uses the Token Cache Provider to verify the existence of the Authorization Code in the Token Cache Data Store.
7. If the Authorization Code exists in the Token Cache Data Store, TFIM STS generates an Access Token and returns the token to the Authorization Server. TFIM STS stores the Access Token in the Token Cache Data Store.

8. If the Authorization code does not exist in the Token Cache Data Store, then TFIM STS returns an error to the Authorization Server.
9. Authorization Server returns the result to the Client.

4.7.11.7. Access Token Validation

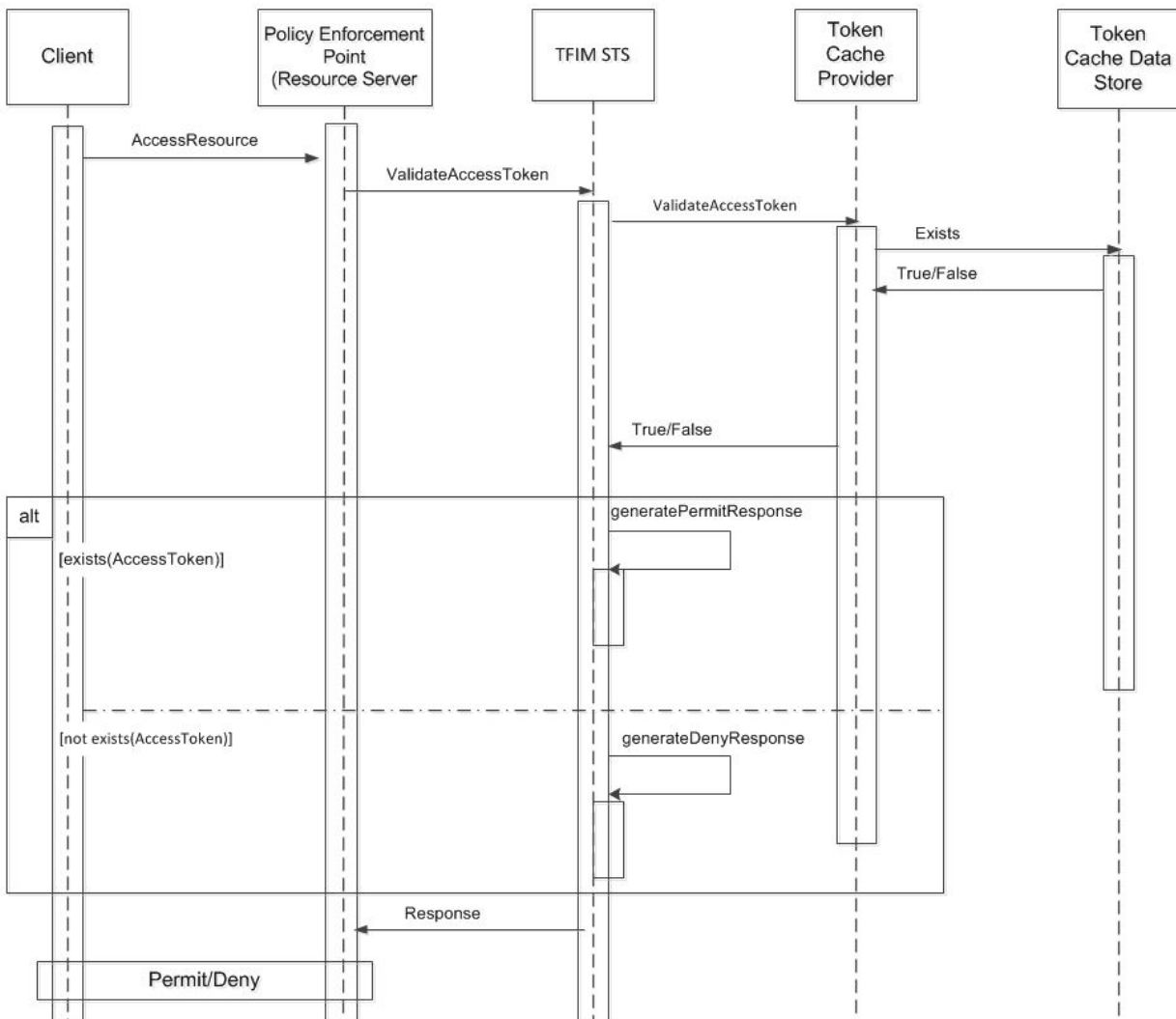


Figure 38: Access Token Validation

10. The Client requests access to the resource by submitting the Access Token.
11. The Resource Server submits a token validation request to the TFIM STS.
12. TFIM STS uses the Token Cache Provider to verify the existence of the Access Token in the Token Cache Data Store.
13. If the Access Token exists and is valid (has not expired), then the TFIM STS returns a response with the grant authorization set to “Permit.”
14. If the token does not exist or is not valid, then TFIM STS returns a response with the grant authorization set to “Deny.”

15. Based on the response, the Resource Server Permits/Denies Client request.

NOTE: TFIM STS is a WS-Trust compliant service.

4.7.12. STS Sequence Diagram

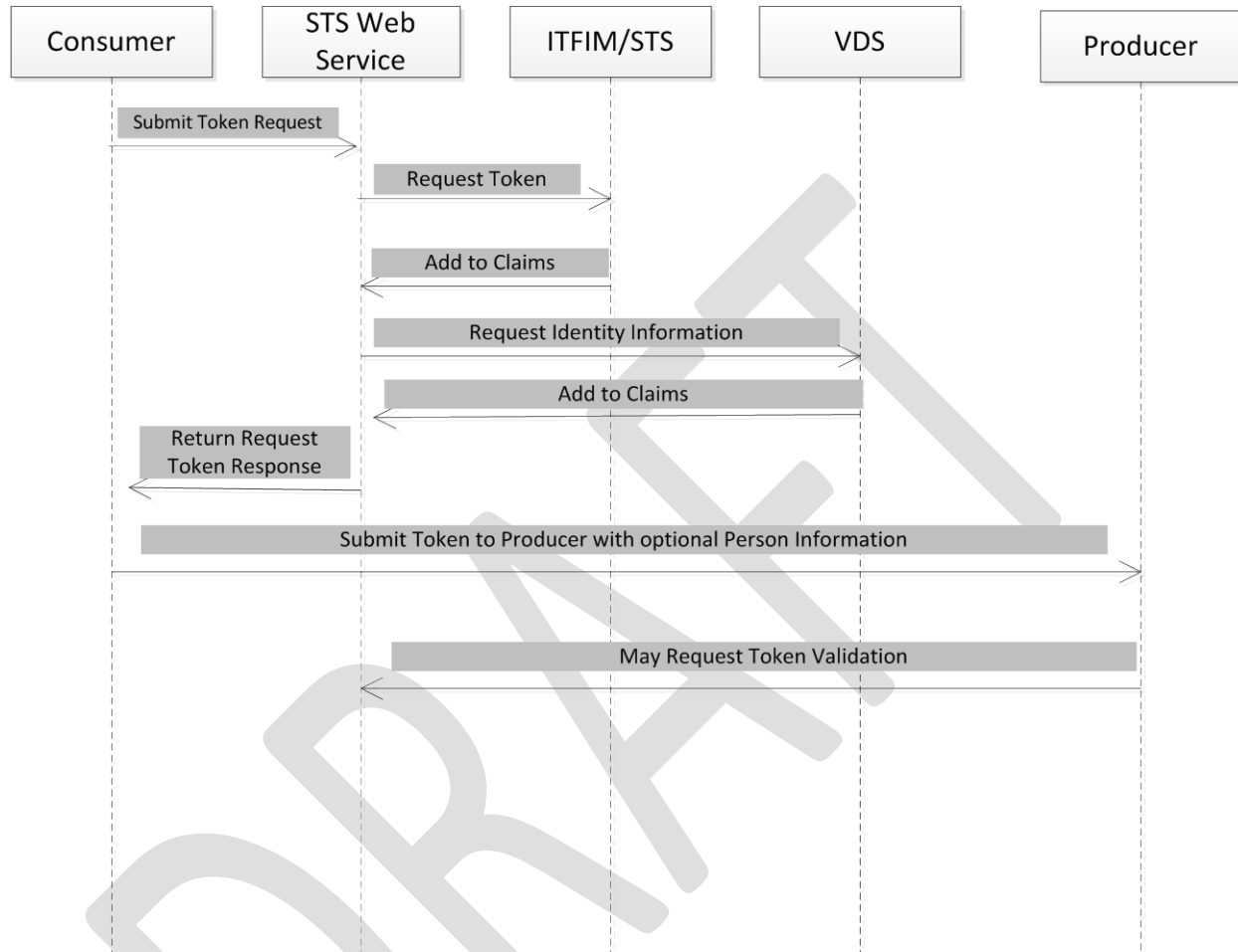


Figure 39: STS Sequence Diagram

The VAAFI STS Web Service diagram session displayed above is as follows:

Assumption: Veteran (client) has established an active security token with VAAFI.

1. A Veteran user (client) selects a VAAFI enabled Enterprise Application (EA).
2. The client presents an SSO session token to VAAFI and the EA for authentication and authorization.
3. Client actions within the application cause the EA to need to consume additional resources within the VA. The EA becomes a consumer of an SOA Producer.
4. On behalf of the client, the consumer obtains required authorization credentials to the Producer. It the client's SSO Session Token to create a WS-Trust Request Security Token (RST) message.
5. The WS-Trust RST message is sent to the VAAFI STS Web Service utilizing SOAP over a mutually authenticated TLS and WS-Security protected connection.

6. VAAFI STS Web Service receives the RST message, decrypts it, and requests a security token from VAAFI STS Identity Provider.
7. VAAFI STS Identity Provider authenticates the client, and issues a new security token or claim back to the STS Web Service.
8. VAAFI STS Web Service optionally queries AcS Provisioning for enhanced claims data (e.g., VDS).
9. VAAFI STS Web Service responds with the requested security token in a WS-Trust RSTR message or an “Unable to Fulfill” error message to the consumer if any of the lookups in steps 6-8 fail for any reason.
10. The consumer receives the WS-Trust RSTR message, and forwards the required claim(s) along with a data request to the Producer.
11. The producer receives the claim and request, and performs token validation based on business rules which may include validation of issuer, calling application binding, conditional and digital signature, and encryption.
12. Once it validates, the Producer evaluates the claim and request, and responds with business data as appropriate for the successful claim authorization for that client.
13. The Producer provides the authorized business data to the consumer.
14. The consumer receives the business data, and responds back to the client as appropriate.

5. Data Design

Because VAAFI is a COTS product implementation that does not store any user data, it has no data storage and therefore no data design. AccessVA has no external data store. The matrix is stored as XML and is internal to the AccessVA application. OAuth solution uses an external RDBMS to support several features.

5.1. DBMS Files

5.1.1. OAuth Physical Data Model

The diagram below depicts the physical data model of the OAuth solution. This model serves as the underlying data model for OAuth customizations and the OAuth web application:

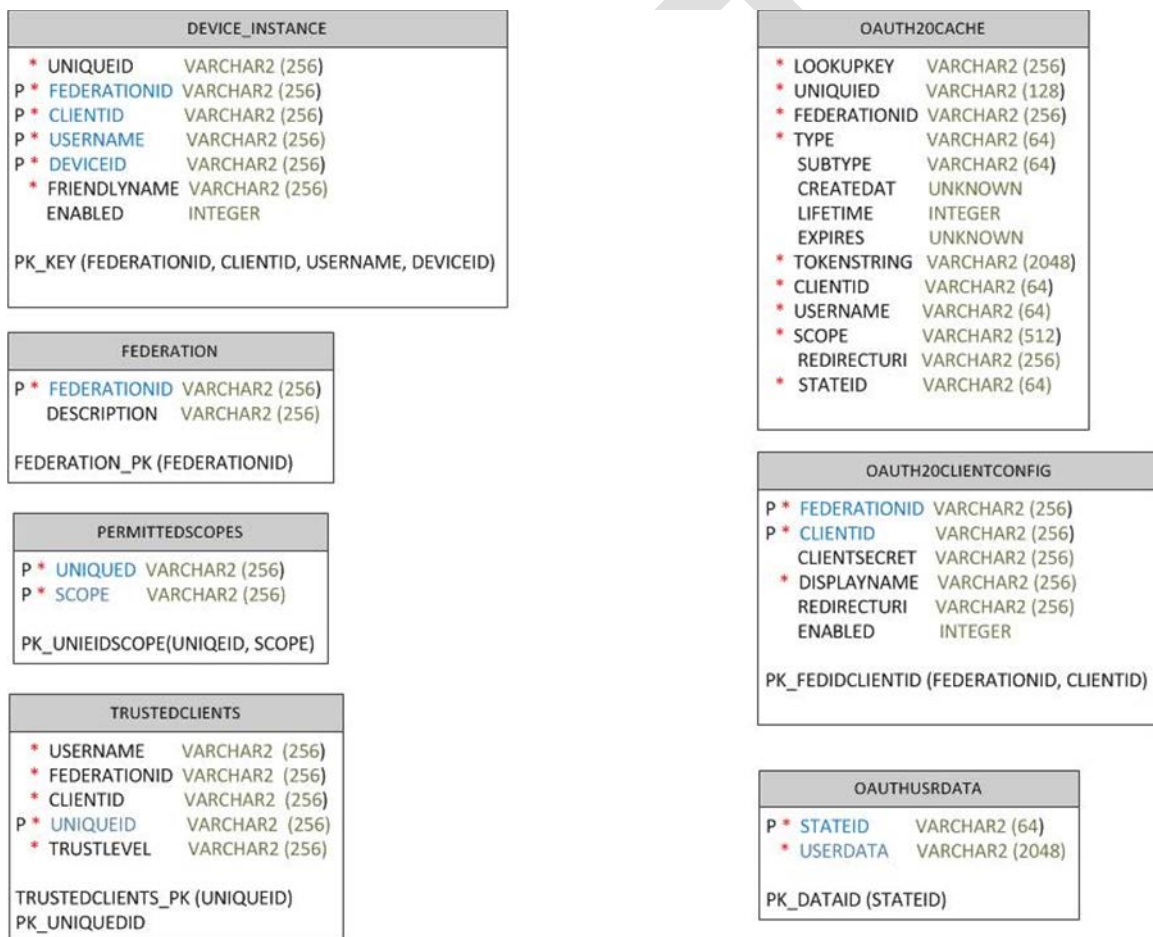


Figure 40: OAuth Physical Data Model

Please note the model does not leverage the database referential integrity through foreign keys in the implementation. As a result, the relationships between the tables are not depicted. The components perform the necessary referential integrity controls.

5.1.2. OAuth Tables and Fields

The table below describes the table fields and their type definitions.

NOTE: The type is identified by the JAVA data type and the syntax is defined using Perl Compatible Regular Expressions (PCRE)

DRAFT

Table 31: OAuth Table Fields and Types

Field Name	Type	Syntax	Description	Comments
FEDERATIONID	STRING	. +	Unique identifier for a federation in TFIM when a new federation is defined. This is the name assigned to the OAuth federation in the WebSphere Administration console during the creation of the OAuth federation.	Unique for a TFIM management domain.
CLIENTID	STRING	. +	Unique identifier assigned to OAuth client during registration process.	CLIENTID is unique within a OAuth Federation
CLIENTSECRET	STRING	. +	A secret assigned to OAuth client during registration process. Used for client authentication	System generated value of length 10 using a random number generator
DISPLAYNAME	STRING	. +	A friendly name for Client	None
REDIRECTURI	URL	^[a-zA-Z0-9\-\.\.]+\.(com org net mil edu COM ORG NET MIL EDU)\$	A URI established during client registration. Tokens will be sent to this endpoint if defined	None
ENABLED	INTEGER	{0,1}	Active state of the client	

TRUSTEDCLIENTS

Field Name	Type	Syntax	Description	Comments
USERNAME	STRING	. +	Unique user identifier from the CSP	Concatenation of the CSP Identifier and the username used to authenticate at the CSP site
FEDERATIONID	STRING	. +	Unique identifier for a federation in TFIM when a new federation is defined.	Same as the OAUTH20CLIENTCONFIG->FEDERATIONID

Field Name	Type	Syntax	Description	Comments
CLIENTID	STRING	. +	Unique identifier assigned to OAuth client during registration process	Same as the OAUTH20CLIENTCONFIG->CLIENTID
UNIQUEID	STRING	. +	Unique identifier for each TRUSTEDCLIENTS entity.	System generated value using the java.util.UUID module
TRUSTLEVEL	STRING	{KNOWN, UNKNOWN}	Indicator of user consent for scopes	None

PERMITTEDSCOPES

Field Name	Type	Syntax	Description	Comments
UNIQUEID	STRING	. +	Unique identifier for each TRUSTEDCLIENTS entity.	Same as TRUSTEDCLIENTS->UNIQUEID
SCOPE	STRING	.+	Scope Name	None

DEVICE_INSTANCE

Field Name	Type	Syntax	Description	Comments
USERNAME	STRING	. +	Unique user identifier from the CSP	Same as TRUSTEDCLIENTS->USERNAME
FEDERATIONID	STRING	. +	Unique identifier for a federation in TFIM when a new federation is defined.	Same as the OAUTH20CLIENTCONFIG->FEDERATIONID
CLIENTID	STRING	. +	Unique identifier assigned to OAuth client during registration process	Same as the OAUTH20CLIENTCONFIG->CLIENTID
UNIQUEID	STRING	. +	Unique identifier for each TRUSTEDCLIENTS entity.	Same as TRUSTEDCLIENTS->UNIQUEID

Field Name	Type	Syntax	Description	Comments
DEVICEID	STRING	. +	Unique identifier for user's device TRUSTEDCLIENTS entity.	None
FRIENDLYNAME	STRING	. +	A friendly name for Client	None
ENABLED	INTEGER	{0,1}	Active state of the device	None

OAuth20Cache

Field Name	Type	Syntax	Description	Comments
LOOKUPKEY	STRING	. +	Unique identifier for each issued token.	Concatenation of the Federation ID and token String
UNIQUEID	STRING	. +	Unique identifier for each TRUSTEDCLIENTS entity.	Same as TRUSTEDCLIENTS->UNIQUEID
FEDERATIONID	STRING	. +	Unique identifier for a federation in TFIM when a new federation is defined.	Same as the OAuth20ClientConfig->FEDERATIONID
TYPE	STRING	{access_token, authorization_code}	Token type	None
SUBTYPE	STRING	{ authorization_grant, implicit_grant, client_credentials, resource_owner_password}	Grant Type	None
CREATEDAT	long	\d+	Token creation date, represented as number of milliseconds since the standard base time, January 1, 1970, 00:00:00 GMT	None

Field Name	Type	Syntax	Description	Comments
LIFETIME	integer	\d+	Token validity period in seconds	None
EXPIRES	long	\d+	Token creation date, represented as number of milliseconds since the standard base time, January 1, 1970, 00:00:00 GMT	None
TOKENSTRING	STRING	. +	Token Value	System generated unique value
CLIENTID	STRING	. +	Unique identifier assigned to OAuth client during registration process	Same as the OAUTH20CLIENTCONFIG->CLIENTID
USERNAME	STRING	. +	Unique user identifier from the CSP	Same as TRUSTEDCLIENTS->USERNAME
SCOPE	STRING	.+	Scope Name	None
REDIRECTURI	URL	^[a-zA-Z0-9\-\.\+\.](com org net mil edu COM ORG NET MIL EDU)\$	A URI established during client registration. Tokens will be sent to this endpoint if defined	None
STATEID	STRING	{refresh_token, authorization_code}	The state id of this token. Tokens that are derivatives from another token (i.e. from a refresh token or authorization_code) must maintain the same stateid as the parent token	None

OAUTHUSRDATA

Field Name	Type	Syntax	Description	Comments
STATEID	STRING	{refresh_token, authorization_code}	The state id of this token. Tokens that are derivatives from another token (i.e. from a refresh token or authorization_code) must maintain the same stateid as the parent token	None
USERDATA	STRING	.+	Session data	None

The following is the Data Definition Language statements for creating the above data model:

```
CREATE TABLE OAUTH20CLIENTCONFIG
(FEDERATIONID VARCHAR(256) NOT NULL,
CLIENTID VARCHAR(256) NOT NULL,
CLIENTSECRET VARCHAR(256),
DISPLAYNAME VARCHAR(256) NOT NULL,
REDIRECTURI VARCHAR(256),
ENABLED INT);
```

```
ALTER TABLE OAUTH20CLIENTCONFIG ADD CONSTRAINT PK_FEDIDCLIENTID
PRIMARY KEY (FEDERATIONID,CLIENTID);
```

```
CREATE TABLE TRUSTEDCLIENTS
(USERNAME VARCHAR(256) NOT NULL,
FEDERATIONID VARCHAR(256) NOT NULL,
CLIENTID VARCHAR(256) NOT NULL,
UNIQUEID VARCHAR(256) NOT NULL PRIMARY KEY,
TRUSTLEVEL VARCHAR(256) NOT NULL);
```

```
ALTER TABLE TRUSTEDCLIENTS ADD CONSTRAINT PK_UNIQUEID PRIMARY KEY
(UNIQUEID);
```

```
CREATE TABLE PERMITTEDSCOPES
(UNIQUEID VARCHAR(256) NOT NULL,
SCOPE VARCHAR(256) NOT NULL);
```

```
ALTER TABLE PERMITTEDSCOPES ADD CONSTRAINT PK_UNIQUEIDSCOPE
PRIMARY KEY (UNIQUEID,SCOPE);
```

```
CREATE TABLE DEVICE_INSTANCE
(UNIQUEID VARCHAR(256) NOT NULL,
FEDERATIONID VARCHAR(256) NOT NULL,
```

CLIENTID VARCHAR(256) NOT NULL,
USERNAME VARCHAR(256) NOT NULL,
DEVICEID VARCHAR(256) NOT NULL,
FRIENDLYNAME VARCHAR(256) NOT NULL,
ENABLED INT);

ALTER TABLE DEVICE_INSTANCE ADD CONSTRAINT PK_KEY PRIMARY KEY
(FEDERATIONID,CLIENTID,USERNAME,DEVICEID);

CREATE TABLE FEDERATION
(FEDERATIONID VARCHAR(256) NOT NULL PRIMARY KEY,
DESCRIPTION VARCHAR(256));

CREATE TABLE OAUTH20CACHE
(LOOKUPKEY VARCHAR(256) NOT NULL,
UNIQUEID VARCHAR(128) NOT NULL,
FEDERATIONID VARCHAR(256) NOT NULL,
TYPE VARCHAR(64) NOT NULL,
SUBTYPE VARCHAR(64),
CREATEDAT BIGINT,
LIFETIME INT,
EXPIRES BIGINT,
TOKENSTRING VARCHAR(2048) NOT NULL,
CLIENTID VARCHAR(64) NOT NULL,
USERNAME VARCHAR(64) NOT NULL,
SCOPE VARCHAR(512) NOT NULL,
REDIRECTURI VARCHAR(256),
STATEID VARCHAR(64) NOT NULL);

CREATE TABLE OAUTHUSRDATA
(STATEID VARCHAR(64) NOT NULL PRIMARY KEY,
USERDATA VARCHAR(2048));

5.2. Non-DBMS Files

5.2.1. VAAFI Data

VAAFI provides the following data elements to consuming applications in the form of HTTP headers for Standard Junction partners, or SAML attributes for SAML Reassertion partners. (This section corresponds to SPEC192.7.9.14.)

DRAFT

Table 32: HTTP Headers

HTTP Header	Description	Domain of Values/Validation Rule(s) Any header whose value cannot be populated will be returned with the value of "NOT_FOUND"	MVI Identifier	LOA
va_eauth_csid	Credential Service Provider's Unique identifier within the federation's boundaries	"DS Logon", "33" (PKI CSP), "Symantec", "FCCX"	TBD	TBD

HTTP Header	Description	Domain of Values/Validation Rule(s) Any header whose value cannot be populated will be returned with the value of "NOT_FOUND"	MVI Identifier	LOA
va_eauth_uid	A unique identifier for the end user so that no two subscribers of the same CSP can have the same uid	<p>DS Logon Examples:</p> <p>There are three forms of this DN depending on whether the va_eauth_authenticationmethod is DS Logon, CAC, or Defense Finance and Accounting Service (DFAS). Each DN is the same except for the identifier portion. All follow the same format: "id=" <identifier> ",ou=user,o=beneficiaries,dc=osd,dc=mil"</p> <p>DS Logon username and password example: id=firstname.lastname,ou=user,o=beneficiaries,dc=osd,dc=mil</p> <p>DS Logon CAC Logon example: id=1234567890,ou=user,o=beneficiaries,dc=osd,dc=mil</p> <p>Symantec example: username@email.domain</p> <p>PKI CSP example: cn=test user, 0.9.2342.19200300.100.1.1=test.user@va.gov, ou=people, o=internal,dc=va,dc=gov</p> <p>FCCX example: <identifier></p>	TBD	TBD

HTTP Header	Description	Domain of Values/Validation Rule(s) Any header whose value cannot be populated will be returned with the value of "NOT_FOUND"	MVI Identifier	LOA
va_eauth_hash	The hash of the userID and CSID	27 characters that are in the ranges a-z, A-Z, 0-9 and the special characters "+" and "/". These are followed by an "=" character for a total of 28 characters. Example: CpjrdSXMPb2P+YK2VOCV81l/iW0=	TBD	TBD
va_eauth_commonname	Field intended to contain the user's name for display purposes. However its contents varies greatly between CSPs.	Symantec Example: N/A PKI CSP Example: "Test.User@va.gov" or "tuser1" FCCX Example: "user name"	TBD	TBD
va_eauth_assurancelevel	The user's LOA at the time of authentication to the CSP	"1", "2", "3", "4"	TBD	TBD
va_eauth_issueinstant	The time of authentication to the CSP	YYYY-MM-DD followed by the character "T" followed by HH:MM:SS followed by the character "Z". Example: 2012-06-12T16:41:59Z	TBD	TBD
va_eauth_email	User's email address as sent by the CSP or if the CSP does not provide an email address as provided by VDS.	user@email.domain	TBD	TBD

HTTP Header	Description	Domain of Values/Validation Rule(s) Any header whose value cannot be populated will be returned with the value of "NOT_FOUND"	MVI Identifier	LOA
va_eauth_firstname	User's first name is from MVI, or is from the CSP if VDS is unavailable or if the LOA is 1.	Character string up to length 20	TBD	TBD
va_eauth_middlename	User's middle name is from MVI, or is from the CSP if VDS is unavailable or if the LOA is 1.	Character string up to length 20	TBD	TBD
va_eauth_lastname	User's last name is from MVI, or is from the CSP if VDS is unavailable or if the LOA is 1.	Character string up to length 26	TBD	TBD
va_eauth_street	User's physical street address as registered at CSP. The value will never be passed if the CSP is the VA PKI CSP.	Character string up to length 35 Example: 123 Pleasant Street	TBD	TBD
va_eauth_street1	User's physical street address as registered at CSP. The value will never be passed if the CSP is the VA PKI CSP.	Character string up to length 35 Example: 123 Pleasant Street	TBD	TBD
va_eauth_street2	User's physical street address as registered at CSP. The value will never be passed if the CSP is the VA PKI CSP.	Character string up to length 35 Example: 123 Pleasant Street	TBD	TBD

HTTP Header	Description	Domain of Values/Validation Rule(s) Any header whose value cannot be populated will be returned with the value of "NOT_FOUND"	MVI Identifier	LOA
va_eauth_street3	User's physical street address as registered at CSP. The value will never be passed if the CSP is the VA PKI CSP.	Character string up to length 35 Example: 123 Pleasant Street	TBD	TBD
va_eauth_city	User's physical city as registered at CSP. The value will never be passed if the CSP is the VA PKI CSP.	Character string up to length 30 Example: Hometown	TBD	TBD
va_eauth_state	User's physical state as registered at CSP. The value will never be passed if the CSP is the VA PKI CSP.	Two-character abbreviations for U.S. States. Example: WV Foreign addresses will contain a region or province as text.	TBD	TBD
va_eauth_country	User's physical country as registered at CSP. The value will never be passed if the CSP is the VA PKI CSP.	ISO 3166-1 alpha-3 for the 'Country' attribute. Until foreign addresses are supported "US" is passed.	TBD	TBD
va_eauth_postalcode	User's physical postal code as registered at CSP. The value will never be passed if the CSP is the VA PKI CSP.	United States Example: 123451234 Foreign addresses will contain a postal code value appropriate to the country.	TBD	TBD

HTTP Header	Description	Domain of Values/Validation Rule(s) Any header whose value cannot be populated will be returned with the value of "NOT_FOUND"	MVI Identifier	LOA
va_eauth_phone	User's phone number as registered with CSP.	20 character field.	TBD	TBD
va_eauth_pnid	User's ID is from MVI, or is from the the CSP if VDS is unavailable or if the LOA is 1.	9- or 10-character numeric field, may contain leading zeros so should not be stored as a number: 123456789 or 0123456789 [Only sent on first logon to Symantec NOT_FOUND in all subsequent Symantec logins]	TBD	TBD

HTTP Header	Description	Domain of Values/Validation Rule(s) Any header whose value cannot be populated will be returned with the value of "NOT_FOUND"	MVI Identifier	LOA
va_eauth_pnidtype	User's ID Type will be SSN if the SSN is available from MVI, or from the CSP if VDS is unavailable or if the LOA is 1.	<p>DEPENDENT=Special 9-digit code created for individuals (i.e., babies) who do not have or have not provided an SSN when the record is added to DEERS (dependents only)</p> <p>FOREIGN=Special 9-digit</p> <p>TAX_IDENTIFICATION=Tax identification number</p> <p>PATIENT=Patient Identifier</p> <p>INVALID_SSN=Invalid SSN. The PN_ID was submitted as an SSN, but does not conform to the valid SSN structure. Obsolete value, no longer applied.</p> <p>SERVICE=Special 9-digit code created for U.S. military personnel from Service Numbers before the switch to Social Security Numbers</p> <p>CONTRACTOR=Special 9-digit code created for a DoD contractor who refused to give his or her SSN to RAPIDS. The associated PN_ID will begin with 99.</p> <p>SSN=Social Security Number</p> <p>TEST=Test Identifier</p> <p>DOD_ED⁸²PN_ID=10-digit code created for DoD affiliates, SSN [only sent on first logon to Symantec NOT_FOUND in all</p>	TBD	TBD

HTTP Header	Description	Domain of Values/Validation Rule(s) Any header whose value cannot be populated will be returned with the value of "NOT_FOUND"	MVI Identifier	LOA
va_eauth_birthdate	User's birth date as registered at MVI, or from the CSP if VDS is unavailable or if the LOA is 1. Only available until June 1, 2016 - deprecated	The format of DoB is YYYY-MM-DDT00:00:00-00:00	TBD	TBD
va_eauth_birthdate_v1	Will represent the actual data received from MVI. Imprecise date values will be supported.	CCYY[MM[DD[HHMM[SS[.S[S[S[S]]]]]]][+/-ZZZZ]^< timezone difference from zulu>	TBD	TBD
va_eauth_dodedipnid	The user's DoD electronic data interchange person identifier from DMDC if the credential is the DS Logon, or from MVI if the CSP is not DS Logon.	A 10-character string where each character is a numeral 0-9. Beware storing this as leading zeroes are common. If the CSP is not DS Logon this field could contain two EDIPs separated by a comma. Multiple values indicates that multiple EDIPs exist at DoD for the same individual.	TBD	TBD
va_eauth_authenticationmethod	The method the user used to authenticate at DS Logon web site.	"DSLogon"; "CAC"; "DFAS"	TBD	TBD

HTTP Header	Description	Domain of Values/Validation Rule(s) Any header whose value cannot be populated will be returned with the value of "NOT_FOUND"	MVI Identifier	LOA
va_eauth_authenticationauthority		"V" indicates association with the VA, "D" indicated association with the DoD and "B" indicates association with both.	TBD	TBD
va_eauth_icn	ICN assigned by MVI.	An ICN can be a character string up to 29 characters.	TBD	TBD
va_eauth_pid	Participant ID assigned by the Corporate Database (DB).	0 to many Participant IDs assigned by the Corporate Database separated by a comma. If multiple Participant IDs are returned for a single identity, these represent multiple records in CorpDB that are linked to the same identity. The system does not know which of these Participant IDs is most recent or preferred.	TBD	TBD
va_eauth_birlsfilenumber	File Number from the Beneficiary Identification Records Locator Subsystem (BIRLS) database.	0 to many File Numbers from the BIRLS database separated by a comma. If multiple BIRLS File Numbers are returned for a single identity, these represent multiple records in BIRLS that are linked to the same identity. The system does not know which of these BIRLS File Numbers is most recent or preferred.	TBD	TBD

HTTP Header	Description	Domain of Values/Validation Rule(s) Any header whose value cannot be populated will be returned with the value of "NOT_FOUND"	MVI Identifier	LOA
va_eauth_secid	<p>Security Identifier from IAM Provisioning Service. This information will never be passed for LOA1. SecID is the enterprise user Security Identifier. SecID is a unique ID assigned to a user when added to the Provisioning System via an Onboarding event (3POB or CRISP). SecID once assigned remains the same even if the user status with the VA changes over time (i.e. Veteran becomes contractor, and then later becomes employee). SecID is the identifier used to correlate Provisioning records to the MVI's ICN (ICN is the unique person identifier).</p> <p>SecID is intended to provide applications a unique identifier to be used in identify users and recording logs and audit records for what a user has done across systems.</p>	<p>SEC_ID is a 10 digit value and will always have padding zeroes in the front if the meaningful digits are less than 10 (e.g. 0000012345)</p> <p>One SECID from the VDS database or multiple numbers from the VDS database separated by a comma. If multiple SECIDs are returned for a single identity, these represent multiple records in VDS that are linked to the same identity. The system does not know which of these is most recent or preferred.</p>	TBD	TBD

HTTP Header	Description	Domain of Values/Validation Rule(s) Any header whose value cannot be populated will be returned with the value of "NOT_FOUND"	MVI Identifier	LOA
va_eauth_mhvien	Person ID assigned by the MHV System	9 numeric characters	TBD	TBD
va_eauth_csponly	CSP Data Only	True if only CSP Data will be provided (If the user has completed 3POB or not), false otherwise.	TBD	TBD
va_eauth_backenddown	IAM Service Down	True if unable to provide Portal Strategy data due to an IAM service being down, false otherwise.	TBD	TBD
va_eauth_prefix	User's Prefix is from MVI, or is from the CSP if VDS is unavailable or if the LOA is 1.	Character string up to length 10	TBD	TBD
va_eauth_suffix	User's Suffix is from MVI, or is from the CSP if VDS is unavailable or if the LOA is 1.	Character string up to length 10	TBD	TBD
va_eauth_gender	User's Gender is from MVI, or is from the CSP if VDS is unavailable or if the LOA is 1.	Male, Female, Unknown	TBD	TBD
va_eauth_hdr_version	Version number of current header set.	"1" for this version and incremented for future versions.	TBD	TBD

HTTP Header	Description	Domain of Values/Validation Rule(s) Any header whose value cannot be populated will be returned with the value of "NOT_FOUND"	MVI Identifier	LOA
va_eauth_transactionid	A unique value for the user session.	A SHA-2 hash value of length 256 bits. This will result in a 44-character length string with Base 64 encoding.	TBD	TBD
cookie	This is a multi-valued field that contains application cookies as well as the VAAFI cookie called PD-S-SESSION-ID.	PD-S-SESSION-ID=1_vu11JUW4Vt+E8D4uDyKZKAZ5JTHfN4ctk/Lv2GobyrPL2pbWck=_AAAAAAE=_tLzhw5aMd8ryFkCu6DlpKjCbES8=;	TBD	TBD

HTTP Header	Description	Domain of Values/Validation Rule(s) Any header whose value cannot be populated will be returned with the value of "NOT_FOUND"	MVI Identifier	LOA
va_eauth_authorization	<p>This is JSON encoded data that is specific to the CSP. An example is provided for the DS Logon Registry and Authorization Attribute Web (RAAWS) response. As additional CSPs have additional traits, the example column will be updated.</p> <p>NOTE: Authorization header data should never be used in lieu of the individual header data, the individual header data should be viewed as authoritative over the data provided in the authorization header.</p>	<p>The following is a test user example for a DS Logon non-surrogate user:</p> <pre>{"authorizationResponse":{"id":796220828,"idType":"SSN","edi":1045848716,"firstName":"ARTHUR","middleName":"E","lastName":"ROSE","gender":"MALE","birthDate":"1954-05-26T00:00:00-07:00","deceased":false,"status":"SPONSOR","personnel":[{"category":"CIVILIAN_RETIREE","entitlementCondition":"00","organization":62,"serviceBranchClassification":"F"},{"category":"GREY_AREA_RETIREE","entitlementCondition":"00","organization":42,"rank":"CMSGT","reservistIndicator":"V2","serviceBranchClassification":"F"}],"benefit":42,"headOfFamily":""}}</pre> <p>The following is a test user example is for a DS Logon surrogate user:</p> <pre>{"subjectAuthorizationResponse":{"surrogate":{"edi":1045848716,"firstName":"ARTHUR","middleName":"E","lastName":"ROSE"},"subject":{"id":796257980,"idType":"SSN","edi":1088808920,"firstName":"ANTONIO","middleName":"Gerard","lastName":"GREEN","gender":"MALE","birthDate":"1967-06-05T00:00:00-07:00","deceased":false,"status":</pre>	TBD	TBD

HTTP Header	Description	Domain of Values/Validation Rule(s) Any header whose value cannot be populated will be returned with the value of "NOT_FOUND"	MVI Identifier	LOA
iv-groups		"dslogon_users", "symanteceauthusers", "piveauthusers"	TBD	TBD
iv-user	This field can be user to differentiate users that are from different CSPs like the va_eauth_csid, but can also differentiate between users that log in with either a PIV or CAC to the PKI CSP or whether they are logging in as themselves or as a surrogate for another.	"dslogoneauthuser", "symanteceauthusers", "piveauthusers", "caceauthuser", "usaaeauthuser", "dslogonsurrogateeauthuser"	TBD	TBD
va_eauth_sponsordod edipnid	DoD EDI PN ID of Family Sponsor (same as DoD EDI PN ID of individual if he/she is a sponsor; DS Logon specific)	A 10-character string where each character is a numeral 0-9. Beware storing this, as leading zeros are common.	TBD	TBD

NOTE: The credential for LOA1 will depend on the credential provider.

Per the Portal Strategy, some attributes are required and some are optional. Required attributes are only required when using the AcS VDS service. When VAAFI is passing CSP traits, VAAFI may not have the user information to pass that attribute. The following presents the attributes, required and optional, and the source attribute or trait name.

Table 33: Attribute Headers

Source Attribute/Trait Name	Required/Optional	Source Primary (Secondary)
ICN	R	MVI
Person ID (Corporate DB)	O	MVI
File Number (BIRLS)	O	MVI
secID	R	Prov
EDIPI	O	DS Logon/MVI
CSID	R	CSP
UID	R	CSP
Hash	R	CSP
Assurance Level	R	CSP
Authentication Method	O	CSP
Authentication Authority	O	CSP
Commonname	O	CSP
Email	O	CSP (Prov)
Last Name	R	MVI
First Name	O	MVI
Middle Name	O	MVI
PNID (usually SSN)	O	MVI (CSP)
Prefix	O	MVI
Suffix	O	MVI
Gender	O	MVI
Date of Birth (DOB)	O	MVI
Imprecise Date of Birth (DOB)	O	MVI
Transaction ID for session user	R	SSOe
The time of authentication to the CSP (Issue Instant)	R	CSP

Source Attribute/Trait Name	Required/Optional	Source Primary (Secondary)
Derived JSON package of CSP's additional traits*	O	CSP
MHV IEN	O	MVI
Street	O	MVI (CSP)
Street 1	O	MVI (CSP)
Street 2	O	MVI (CSP)
Street 3	O	MVI (CSP)
City	O	MVI (CSP)
State	O	MVI (CSP)
Country	O	MVI (CSP)
Postal Code	O	MVI (CSP)
Phone	O	MIV (CSP)
CSP Data Only	R	SSOe
IAM Service Down	R	SSOe

5.2.2. AccessVA Data

AccessVA uses property files and dynamic Spring Beans for storing the mappings between the various applications, the credential assurance levels that the applications accept, and the credential service providers that supply the credential. These files dynamically generate the web views that display the various credential service providers and the level of assurance of the credential that a particular application accept. These mapping files also provide registration URLs for the various CSPs to facilitate user registration and obtain a particular credential to access a particular application.

5.2.3. PKI Registration Data

To be added in a future sprint. Meeting are in process to define a new design to support the Miami/Culpeper Flips.

Currently a PKI Registration Database is used to capture the registration of all PIV and CAC cards.

5.2.4. IAM STS WS Data

NOTE: These three data structures will be updated after their respective Sprint 5 is complete.

5.2.4.1. Request Security Token (RST) Message

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
```

```

<S:Header>
</S:Header>
<S:Body>
  <RequestSecurityToken xmlns="http://docs.oasis-open.org/ws-sx/ws-trust/200512"
    xmlns:wsa="http://www.w3.org/2005/08/addressing"
    xmlns:pol="http://schemas.xmlsoap.org/ws/2004/09/policy">
    <RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</RequestType>
    <pol:AppliesTo>
      <wsa:EndpointReference>
        <wsa:Address>http://DestinationHost/HelloService</wsa:Address>
      </wsa:EndpointReference>
    </pol:AppliesTo>
    <OnBehalfOf>
      <wsse:BinarySecurityToken
        xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
        ValueType="http://sts.ssoi.████████siteminder/std_token"
        EncodingType="base64">
          ugly_token_here.....sdfOIDFKLSoidesdfk
        </OnBehalfOf>
      <TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0>
    </RequestSecurityToken>
  </S:Body>
</S:Envelope>

```

5.2.4.2. Request Security Token Response (RSTR) Message

TBD – To be designed in Sprint 4

5.2.4.3. SAML2 Token

TBD – To be designed in Sprint 4

5.2.4.4. JSON Token

TBD – To be designed in Sprint 4

5.2.4.5. Kerberos Token

TBD – To be designed in Sprint 6

5.3. Data View

TBD – Sprint 2 Design

TBD – Sprint 5 Implementation

5.3.1. OAuth Data

5.3.1.1. JSON Bearer Token

TBD – Sprint 2 Design

TBD – Sprint 5 Implementation

DRAFT

6. Detailed Design

This section describes the proposed design in detail. It provides the necessary information for the development team to integrate the hardware components and configure the software components, so that the hardware and software components will provide a functional product.

6.1. Hardware Detailed Design

The IBM DataPower appliance is the only hardware device that plays a key role in the VAAFI system design. All of the other hardware that is used within the VAAFI infrastructure is to host the Virtual Servers that host all of the software components detailed in Section 6.2. The specifics of those virtual servers is outlined in the Server Planning Sheet.

6.1.1. IBM XI52 DataPower Appliance

The IBM Data Power XI52 is a nondisruptive network device that provides common message transformation, integration, and routing functions on a single platform. It is a purpose-built hardware device that is used to route service related traffic and proxy REST, JSON, and SOAP services to the various VAAFI enablements. It consists of the following key hardware elements:

- 2U high-density rack-mount design; and
- Two network I/O modules for increased flexibility and serviceability (eight 1-Gb and two 10-Gb ports)

Specific throughput and processor capacity depend greatly on the implementation and use of the device. A detailed performance analysis is currently under review with IBM to determine the specifics of the VA implementation.

6.2. Software Detailed Design

VAAFI is fairly complex and composed of many different component subsystems. This section describes these and their implementation subsystems in high-level detail.

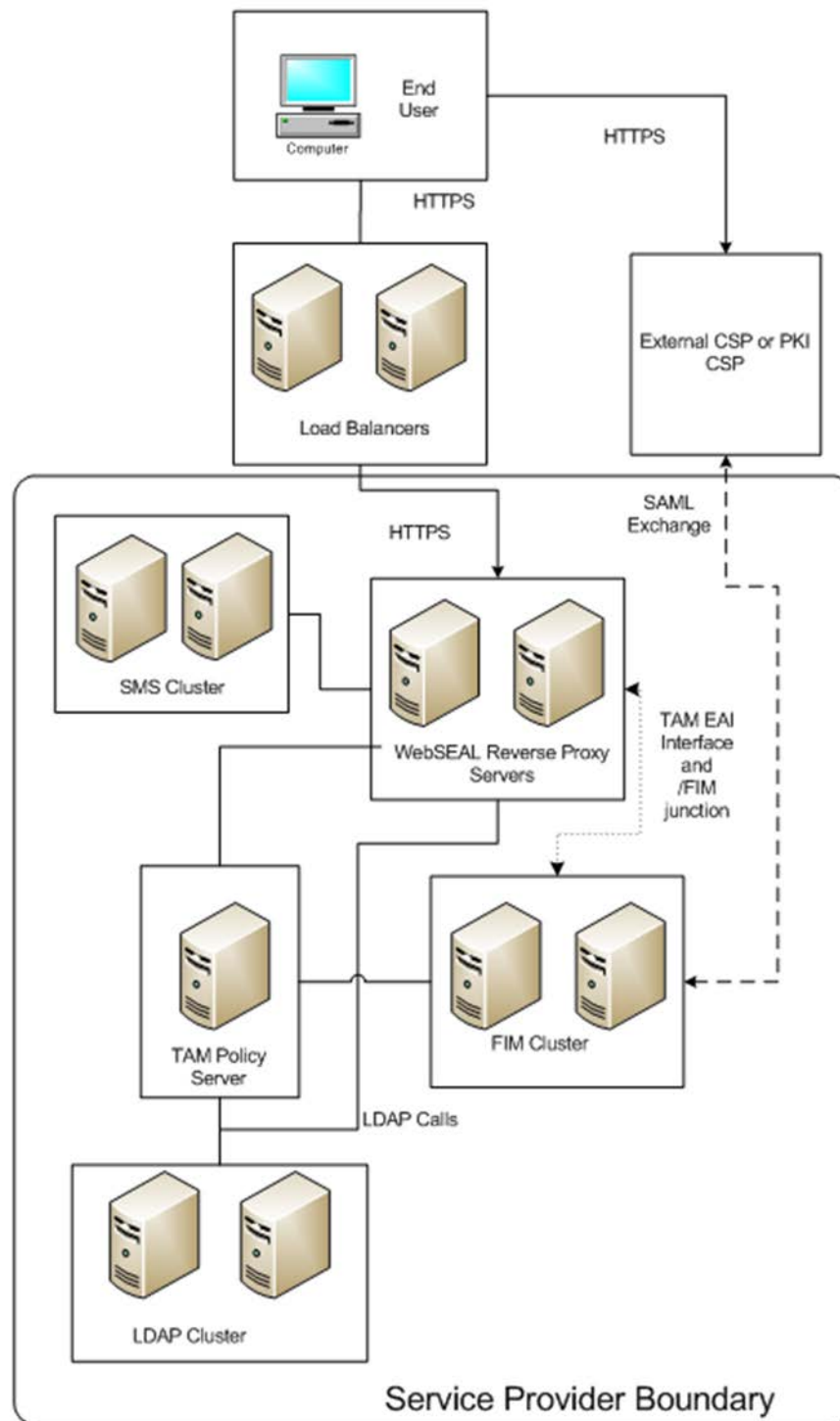


Figure 41: Service Provider Detailed Design

6.2.1. IBM Tivoli Directory Server – User Repository

IBM Tivoli Directory Server (ITDS) V6.2 is a Lightweight Directory Access Protocol (LDAP) directory server. It uses IBM DB2 as the storage repository for all data and serves as the authoritative data source to Tivoli Access Manager for authentication and authorization. The administrative consoles for WebSphere and ITFIM also use the ITDS for authentication and authorization.

6.2.1.1. Software

The following is a list of software components that must be installed on each server to support this subsystem:

- IBM Tivoli Directory Server V6.2;
- IBM DB2 Universal Database Enterprise Server Edition V9.2: This is a standalone version of DB2 rather than the one that is bundled with the IBM Tivoli Directory Server. It allows for better scalability and is more flexible for administration; and
- GSKit-V7.0.4. 36: This component contains the IBM Global Security Kit and provides SSL and TLS capabilities.

6.2.1.2. Database Design

ITDS uses the IBM DB2 relational database as the underlying data store and Structured Query Language (SQL) as the database access language. TFIM OAuth solution also uses the same database but on a separate schema to persist OAuth specific entities.

6.2.1.3. Interfaces

Description	Purpose	Port	Connecting Component(s)	Zone	FW Rule?
DB2 Administration Server Port	DB2 admin application	■	N/A	INT	N
DB2 Instance Connection Port	LDAP server communication to DB2	■	OAuth modules: Token Cache Provider, Trusted Client Provider, Client Configuration Provider OAuth Web Application: Client Registration, Device Registration, Consent Management	INT	N

Description	Purpose	Port	Connecting Component(s)	Zone	FW Rule?
Directory Server Secure Connection Port, Directory Replication Port	LDAPS communication and replication	■	LDAP partners TAM FIM WebSEAL WebSphere	INT INT INT DMZ INT	N N N Y N
Directory Server Administration Secure Port	TDS admin application	■	WebSphere	INT	N

The diagram in shows the ITDS interfaces with other modules within the system.

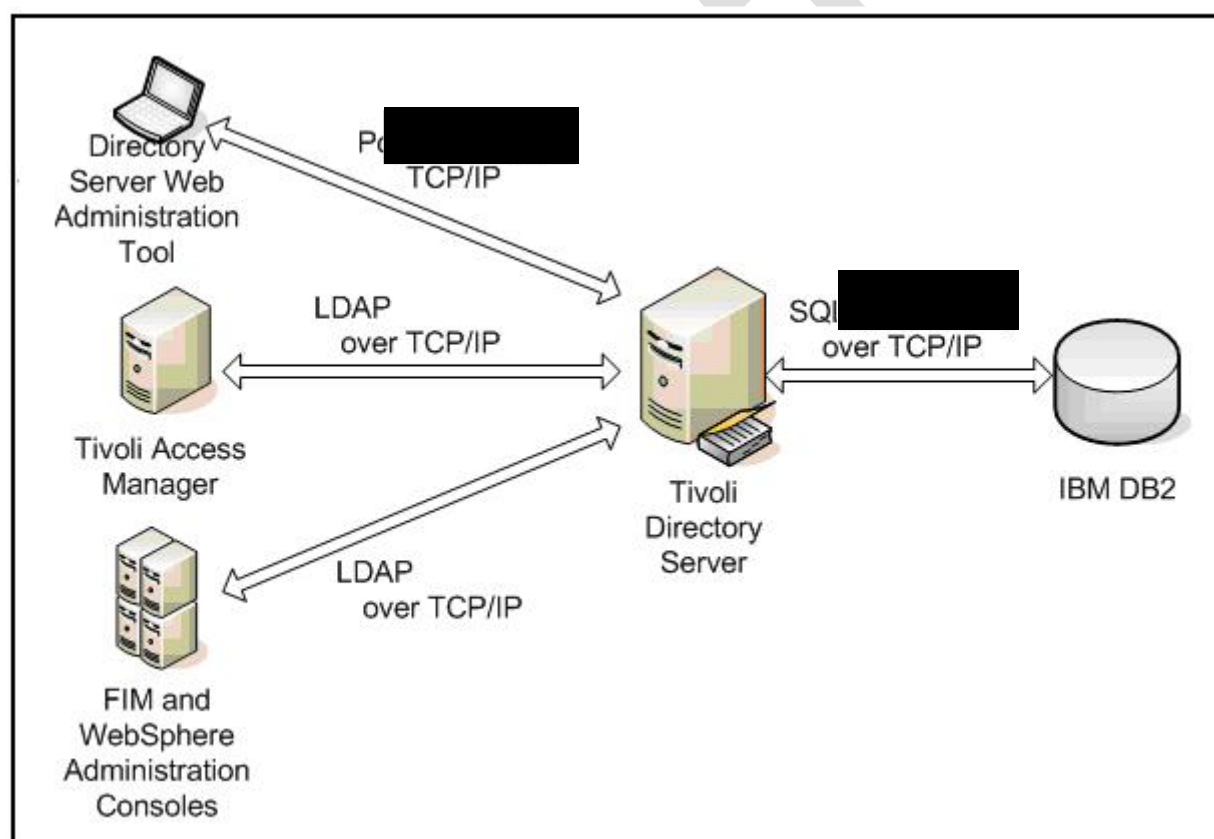


Figure 42: ITDS Interfaces

6.2.1.4. External Interfaces

Lightweight Directory Access Protocol (LDAP) is an open industry standard that defines a method for accessing information directories. Its basis are the standards contained within the X.500 standard, but is significantly simpler and supports TCP/IP. LDAP/S is the encrypted and secure version of LDAP. The communication between IBM Tivoli Access Manager (ITAM) and the ITDS uses LDAP/S.

6.2.1.5. User Interface

The IBM Tivoli Directory Server Web Administration Tool is installed on an embedded version of IBM WebSphere Application Server – Express, which is included with the ITDS, and administered through a console. The IBM Tivoli Directory Servers that have been added to the console can be managed through the Web Administration Tool without having the tool installed on each server. The preferred method of administering the server is by using the Web Administration Tool.

The Web Administration Tool enables a wide range of tasks, such as the following:

- Providing basic server administration tasks;
- Setting server properties;
- Configuring security settings
- Managing the ITDS schema, including managing object classes and attributes;
- Managing replication;
- Managing logs;
- Managing directory entries;
- Managing Access Control Lists, including performing all functions described in the previous sections;
- Managing groups, roles, and proxy authorization groups; and
- Performing user-specific tasks, including managing realms, templates, groups, and users.

6.2.2. IBM Tivoli Access Manager

The IBM Tivoli Access Manager (ITAM) is the core of VAAFI. It is responsible for providing authentication and authorization related services for applications within the VA environment. The ITAM for e-business suite consists of several different software modules operating together to provide the core infrastructure.

6.2.2.1. Software

The WebAccess Manager software package is the console for configuring the Tivoli Access Manager Policy server and WebSEAL components. WebAccess Manager requires several base modules, including the WebSEAL software itself, to operate. The module versions are the following:

- IBM Tivoli Access Manager – Policy Server 6.1.1;
- IBM Tivoli Access Manager – Authorization Server 6.1.1;
- IBM Tivoli Access Manager Runtime 6.1.1;
- IBM Tivoli Access Manager Java Runtime Environment 6.1.1;
- IBM Tivoli Access Manager Web Portal Manager 6.1.1;
- IBM Tivoli Directory Client 6.2;
- IBM GSKIT 7.0.4.36; and
- IBM JRE 1.5.0.

6.2.2.2. Dependencies

The installation of the ITAM requires the following modules be present:

- IBM JRE 1.5
- The IBM GSKIT must be version 7.0.3.8 or higher
- Linux package dependencies (listed in the table below)

Linux Package Dependencies
compat-libstdc++-33.x86_64
compat-libstdc++-33.i386
compat-libstdc++-296.i386
glibc-devel.i386
glibc-devel.x86_64
compat-glibc-headers.x86_64
glibc-headers.x86_64
compat-gcc-34-c++.x86_64
compat-gcc-34.x86_64
libXp.i386
libXp.x86_64

NOTE: These package dependencies are the same for all of the VAAFI servers.

6.2.2.3. Database Design

The ITAM requires a database for storage of policy configuration and administration. This database is a proprietary database implementation based on a flat file btree type database. The database is stored in /var/PolicyDirector/db by default.

6.2.2.4. Interfaces

The ITAM maintains several interfaces () for communicating with applications, databases, and internal processes.

Table 34: Tivoli Access Manager Interfaces

Description	Purpose	Port	Connecting Component(s)	Zone	FW Rule?
LDAP	ITAM uses LDAP to perform user lookups. VAAFI users must exist in the user repository. ITAM uses to make policy decisions regarding access.	636	LDAP	INT	N

Description	Purpose	Port	Connecting Component(s)	Zone	FW Rule?
Policy Server	Most communication that occurs within ITAM occurs or initiates on this port. This port is used for administering ITAM and initiating communication.	████	WebSEAL Servers	DMZ	Y
Policy Replication	Policy database replication with the WebSEAL reverse proxy	████	WebSEAL Servers	DMZ	Y

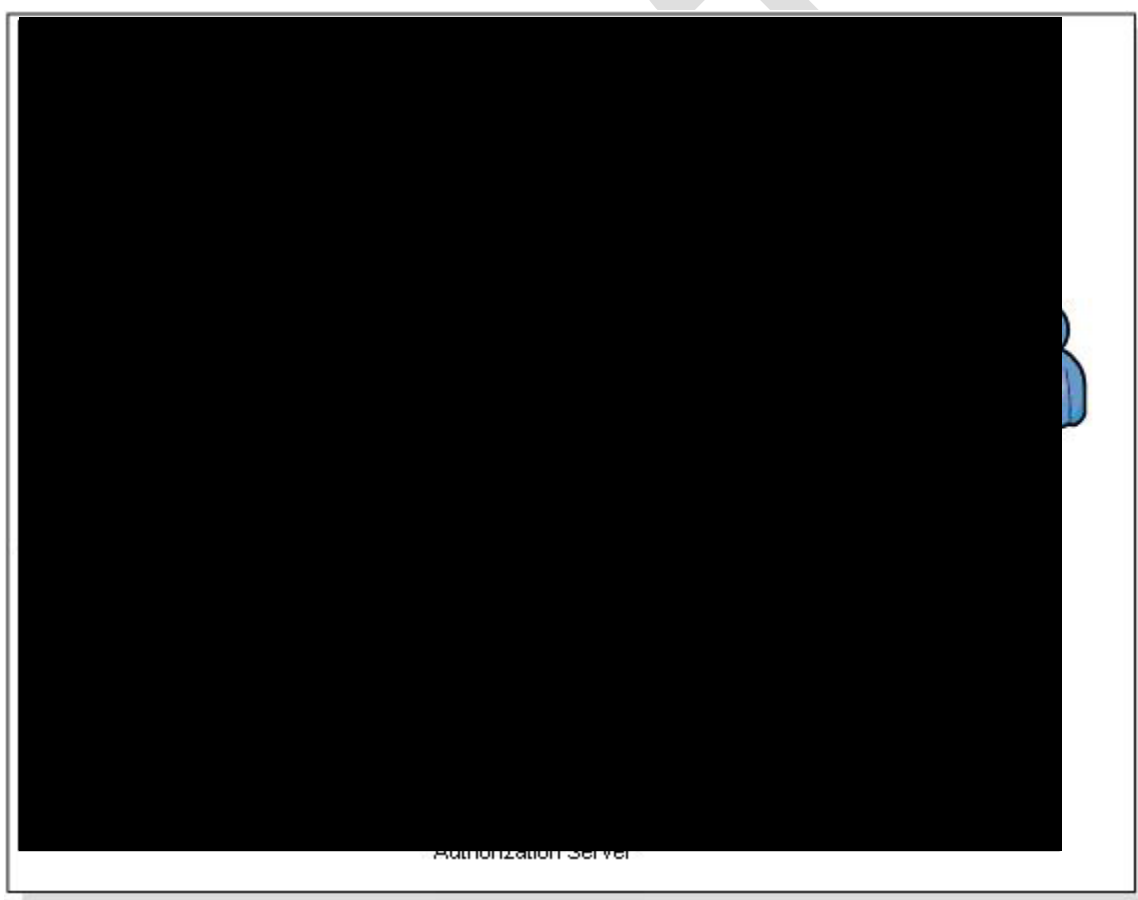


Figure 43: TAM Subsystem

6.2.2.5. User Interface

The ITAM has several utilities that are available for managing the configuration of the Access Manager product. Executing most functions is through command line or the web-based Integrated Service Console (see Section 6.2.2.6, below).

6.2.2.6. Access to Applications

TAM uses three primary means of controlling access to protected resources: Authorization (Authz) Rules, Access Control Lists (ACLs), and Protected Object Policies (POPs). VAAFI uses ACLs to restrict the CSPs for accessing a specific junction. VAAFI uses Authz Rules to restrict the Assurance Level that can access a specific junction. The details of how to use these mechanisms are on a per junction basis and described in the partner's Interface Control Document (ICD). New in VAAFI Increment 3, VAAFI will use Authz Rules to enforce the following rules:

- If a CSP is DS Logon, a va_eauth_dodedipnid cannot be NOT_FOUND.
- A va_eauth_hash cannot be NOT_FOUND.

Currently, when an AuthZ Rule fails, the page that displays is the Insufficient Assurance Level error page.

6.2.3. IBM WebSphere Deployment Manager

WebSphere is a high-end web application server, on which scalable, Java-based e-business applications can deploy. This product is used in VAAFI, which hosts the administrative consoles for the IBM Tivoli Directory Server (ITDS), the ITFIM, and the IBM Tivoli Access Manager Web Portal Manager. WebSphere within VAAFI hosts several applications, as later sections within Section 6.1 describe. The WebSphere Deployment Manager (DM) manages all instances of WebSphere, providing a central management point for all of the WebSphere instances.

6.2.3.1. Software

VAAFI has the following components that operate with different versions of WebSphere:

- IBM WebSphere Application Server Network Deployment V6.1.0.41 for ITFIM;
- IBM WebSphere Extreme Scale;
- SMS Command Line Client;
- TFIM ISC;
- TAM ISC;
- OAuth web application
- SMS ISC; and
- TAM Authorization Server.
- A single deployment manager for the SP subsystem will manage the ITFIM cluster, the SMS cluster and the SMS Catalog Server Cluster, and the IAM Landing Page. This server will also be the management server for ITFIM, SMS, LDAP, TAM, OAuth web application. Management will occur through the Integrated Service Console (ISC) accessible on this server at <https://<localhost>/admin>

Application servers will be built for each of the clusters containing a minimum of two instances of each component for High Availability. The initial production build of this system will include four applications servers in the ITFIM cluster, two application servers in the SMS cluster, and two application servers in the SMS catalog cluster.

6.2.3.2. Database Design

This subsystem does not maintain a database.

6.2.3.3. Internal Interfaces

The WebSphere Application Servers, which run the installations of ITFIM, connect to the Directory Server to authenticate and authorize users that access the WebSphere administration console. As deployed, this authentication and authorization are enabled. WebSphere administration is also blocked at the firewall.

6.2.3.4. External Interfaces

WebSphere acts as a middleware server for the applications that run on it, and those applications encapsulate within them any interfaces that WebSphere has with external systems. For a complete description of these interfaces, refer to the following sections: 6.2.1, 6.2.2, 6.2.4, 6.2.5, 6.2.6, and 6.2.6.13.

6.2.3.5. User Interface

- The IBM WebSphere Application Server Network Deployment V6.1.0.45: This is used for ITFI and server administrators use this. It has a web-based management console (known as ISC), and runs on 9043.
- Web Administration Tool 6.0: This web application is the administration console for the IBM Tivoli Directory Server. This tool enables the server management, both locally and remotely. It is a part of the Directory Server package.

Table 35: Deployment Manager Interfaces

Description	Purpose	Port	Connecting Component(s)	Zone	FW Rule?
ISC	Administrative Interface	9043	Admin Server	INT	N
WAS	Node management and runtime configuration	9043	Deployment Manager	INT	N

The diagram in below depicts how the deployment manager manages the SMS catalog cluster, SMS cluster, and ITFIM cluster.

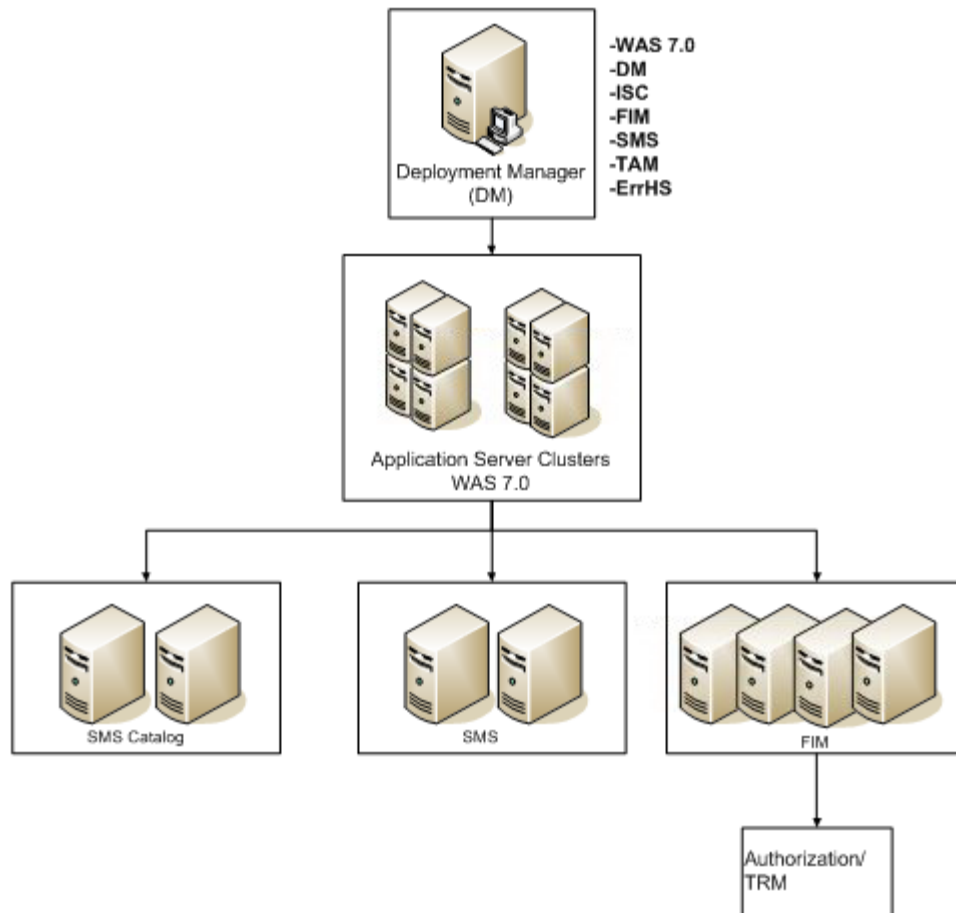


Figure 44: SP Deployment Manager

6.2.3.6. Cluster Naming

The naming conventions for clusters in each environment are below (naming is case sensitive):

- VAAFI-SP-FIM-cluster
- VAAFI-SP-SMS-cluster
- VAAFI-SP-CAT-cluster

6.2.3.7. Node Naming

The naming conventions for the nodes within the clusters in each environment are below (naming is case sensitive):

<server short name>Node<node #>
(where <node #> is always 01)

6.2.3.8. FIM Domain Naming

The naming convention of the FIM domain in each environment is below (naming is case sensitive):

VAAFI-SP

6.2.4. IBM Tivoli Session Management Server (SMS)

The session management server is an optional Tivoli Access Manager component that runs as a WebSphere service. It manages user sessions across Tivoli Access Manager servers, ensures that the session state remains consistent across the participating servers, and allows the implementation of session policy across the participating servers.

The session management server is a J2EE application that runs on the WebSphere server, or within a WebSphere cluster. Within VAAFI, the SMS component will provide a unified view of all WebSEAL user sessions. It will also allow the infrastructure to maintain user sessions without relying on the stickiness settings of the Load Balancers. Stickiness settings have caused problems for both the software load balancers the infrastructure currently uses and the hardware load balancers that are planned for the near future. SMS will also facilitate the security of web services as they pass from internal web sites out through the DataPower devices by enhancing the ability to rewrite the data from applications with the authentication data of the logged in user.

6.2.4.1. Software

The SMS Server is installed into a WebSphere Application Server (WAS) cluster. Each node of this cluster has the following software:

- IBM WebSphere Application Server Network Deployment V6.1.0.45 (this component runs the SMS WebService);
- Tivoli AM Session Management Server v6.1.2;
- WebSphere Extreme Scale (this and WebSphere ND go on the Catalog Servers);
- Tivoli AM Authorization server 6.1.1;
- Tivoli AM Runtime 6.1.1;
- Tivoli AM SMS Command Line 6.1.2; and
- IBM GSKit v.7.0.4.36.

The Integrated Solutions Console (ISC) Session Management Server extension is only installed on the Network Deployment node (into the DM profile).

The WebSphere Application Server configuration has two server profiles on the primary node and one application server profile on the secondary node. The primary node has a Deployment Manager profile, which contains the Cell Manager server. This server runs the WebSphere Integrated Solutions Console along with the SMS extension that can deploy, configure, and administer the Session Management Server.

The other profile on both nodes is an Application Server profile. It contains a node agent server, which manages the state of two application servers that are also contained in this profile. The node is federated to the Deployment Manager, and the two application servers are members of an application server cluster. The SMS application DSess.ear is deployed on that cluster.

6.2.4.2. Dependencies

The installation of the IBM Tivoli Access Manager SMS components requires the following modules be present:

- The IBM GSKIT must be at least version 7.0.4.11;

- WebSphere App Server must be v6.0;
- A fully configured TAM and user registry in the domain; and
- Linux package dependencies (see 6.1.2.2).

6.2.4.3. Database Design

The TAM SMS could use a database to store WebSEAL login history information. Creating the database must occur before deploying the SMS application.

VAAFI does not store login history.

6.2.4.4. Internal Interfaces

The WebSphere Application Servers on the SMS Server establish the following internal connections:

- WebSphere Application Server security is enabled, using a LDAP user account repository.
- The individual application servers maintain a secure connection over TCP port 636 to the Master LDAP user repository server of the VAAFI system.

6.2.4.5. External Interfaces

The replicated WebSEAL servers connect to the SMS Webservice over the secure port 9443. WebSEAL authenticates to the SMS with client certificates.

SMS administrative users can also manage the WebSphere environment by connecting to the WebSphere Integrated Solutions Console. The console accepts secure connections on TCP port 9043.

Table 36: SMS Interfaces

Description	Purpose	Port	Connecting Component(s)	Zone	FW Rule?
SMS application	Communication between SMS and WebSEAL	9443	WebSEAL	DMZ	Y
Server Authentication	Authentication of servers	636	LDAP	INT	N
Deployment Manager	Node management	9043	Deployment Manager	INT	N

6.2.4.6. User Interface

Access to the WebSphere ISC is via <https://<localhost>:9043/ibm/console> on the Deployment Manager. SMS administrative users can use it to administer the WebSphere server environment, including SMS servers and cluster, JDBC data sources, SSL configuration, administrative and application security, etc. It also features the SMS Configuration Console to perform the SMS configuration or unconfiguration, manage session realms, and deploy new SMS instances into a WAS cluster.

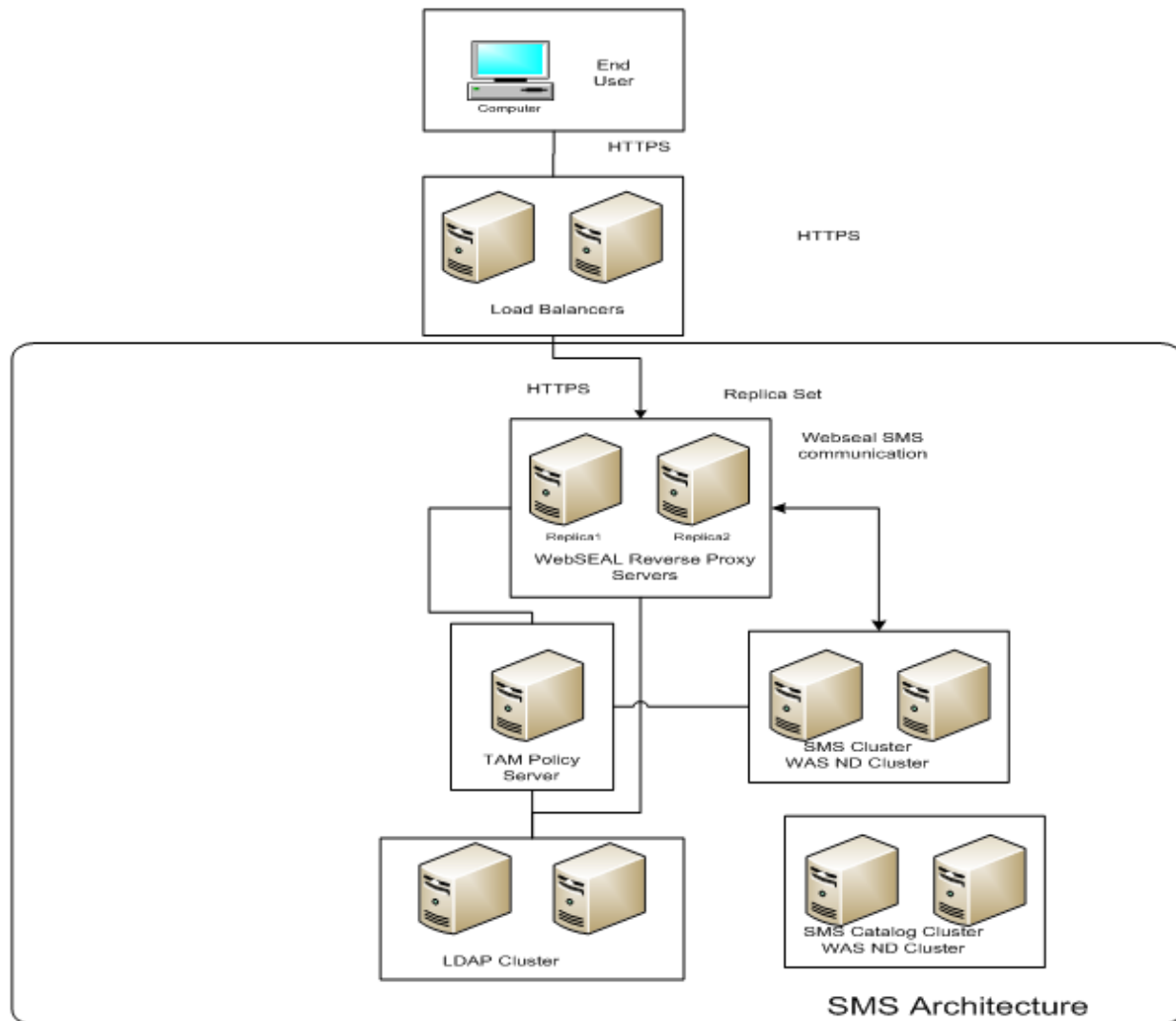


Figure 45: User Interface

6.2.5. IBM Tivoli Session Management Server (SMS) Catalog Server

The session management catalog server is an optional Tivoli Access Manager component that runs as a WebSphere service. It manages user sessions across Tivoli Access Manager servers, ensures that the session state remains consistent across the participating servers, and allows the implementation of session policy across the participating servers.

The session management server is a J2EE application that runs on the WebSphere server, or within a WebSphere cluster. Within VAAFI, the SMS component will provide a unified view of all WebSEAL user sessions. It will also allow the infrastructure to maintain user sessions without relying on the stickiness settings of the Load Balancers. Stickiness settings have caused problems for both the software load balancers the infrastructure currently uses and the hardware load balancers that are planned for the near future. SMS will also facilitate the security of web services as they pass from internal web sites out through the DataPower devices by enhancing the ability to rewrite the data from applications with the authentication data of the logged in user.

6.2.5.1. Software

The SMS Server is installed into a WAS cluster. Each node of this cluster has the following software:

- IBM WebSphere Application Server Network Deployment V6.1.0.45. This component runs the SMS WebService;
- Session Management Server v6.1;
- WebSphere Extreme Scale – this and WebSphere ND go on the Catalog Servers;
- Tivoli AM Authorization server 6.1.1;
- Tivoli AM Runtime 6.1.1;
- Tivoli AM SMS Command Line 6.1.1; and
- IBM GSKit v.7.0.4.36.

The Integrated Solutions Console (ISC) Session Management Server extension is only installed on the Network Deployment node (into the DM profile).

The WebSphere Application Server configuration has two server profiles on the primary node and one application server profile on the secondary node. The primary node has a Deployment Manager profile, which contains the Cell Manager server. This server runs the WebSphere Integrated Solutions Console along with the SMS extension that can serve to deploy, configure, and administer the Session Management Server.

The other profile on both nodes is an Application Server profile. It contains a node agent server, which manages the state of two application servers that are also contained in this profile. The node is federated to the Deployment Manager, and the two application servers are members of an application server cluster. The SMS application DSess.ear is deployed on that cluster.

6.2.5.2. Dependencies

Installing the IBM Tivoli Access Manager SMS components requires the following modules be present:

- The IBM GSKIT must be version 7.0.4.36;
- WebSphere App Server must be v6.0;
- A fully configured TAM and user registry in the domain; and
- Linux package dependencies (see 6.1.2.2).

6.2.5.3. Database Design

The TAM SMS could use a database to store WebSEAL login history information. Creating the database must occur before deploying the SMS application.

VAAFI does not store login history.

6.2.5.4. Internal Interfaces

The WebSphere Application Servers on the SMS Server establish the following internal connections:

- WebSphere Application Server security is enabled, which uses a LDAP user account repository.
- The individual application servers maintain a secure connection over TCP port 636 to the Master LDAP user repository server of the VAAFI system.

6.2.5.5. External Interfaces

The replicated WebSEAL servers connect to the SMS Webservice over the secure port 9443. WebSEAL authenticates to the SMS with client certificates.

SMS administrative users can also manage the WebSphere environment by connecting to the WebSphere Integrated Solutions Console. The console accepts secure connections on TCP port 9043.

Table 37: SMS Interfaces

Description	Purpose	Port	Connecting Component(s)	Zone	FW Rule?
SMS application	Communication between SMS and WebSEAL	9443	WebSEAL	DMZ	Y
Server Authentication	Authentication of servers	636	LDAP	INT	N
Deployment Manager	Node management	9043	Deployment Manager	INT	N

6.2.5.6. User Interface

Access to the WebSphere ISC is via <https://<localhost>:9043/ibm/console> on the Deployment Manager. SMS administrative users use it to administer the WebSphere server environment, including SMS servers and cluster, JDBC data sources, SSL configuration, administrative and application security, etc. It also features the SMS Configuration Console to perform the SMS configuration or unconfiguration, manage session realms, and deploy new SMS instances into a WAS cluster.

6.2.6. IBM Tivoli Federated Identity Manager (ITFIM)

Created for conducting Government to Government (G2G), Government to Citizen (G2C), and Government to Business (G2B) transactions, the basis of the VAAFI solution are several industry open standards and agreements with partners for tailoring those standards. The core of this solution is a software component, which has been tested and approved to be interoperable with the products from additional vendors that the VAAFI partners use. Each of the vendors' software components follow, and include support for, one or more versions of federated identity management standards (e.g., SAML, WS Federation, Shibboleth, Liberty). A developed piece of code (known as va-hash.jar) also runs within ITFIM.

6.2.6.1. Software

ITFIM has separate components that are independent of each other:

- IBM Tivoli FIM management service and runtime component (this includes the ISC component) – installed on Deployment Manager;
- IBM Tivoli FIM Web services security manager – installed on Deployment Manager;
- IBM Tivoli Access Manager – Authorization Server 6.1.1;
- IBM Tivoli Access Manager Runtime 6.1.1;
- IBM Tivoli Access Manager Java Runtime Environment 6.1.1;
- IBM Tivoli Directory Client 6.2;
- IBM GSKIT 7.0.4.36; and
- IBM JRE 1.5.0.

Each of these components has separate software prerequisites.

6.2.6.2. Runtime Component Prerequisites

The management service and runtime component requires that the following software component be pre-installed on the same system:

- IBM WebSphere Application Server Version V6.1.0.45

6.2.6.3. Database Design

The ITFIM subsystem does not directly maintain a database, but interfaces with an existing LDAP for storing component administrative data. The data in the directory is used during the installation and use of the ITFIM subsystem. The following sections outline the manual and automated configuration of the instance of the directory used by ITFIM.

6.2.6.4. Internal Interfaces

The ITFIM subsystem has several entry/exit points which the user and systems (other than the federation-related services) never access directly. Table 27 identifies these interfaces.

WebSEAL (the reverse proxy) supports an ITAM feature, external authentication interface (EAI). This facility allows a protected application such as ITFIM to return the authenticated identity of the user to ITAM. An HTTP header that is part of the HTTP response contains the identity. ITFIM returns this information in the form of an extended privilege attribute certificate (EPAC). The EPAC contains the user credential information. ITAM uses the EPAC to establish an authentication session.

6.2.6.5. External Interfaces

The external interface of the ITFIM components is defined within VAAFI ICDs with CSP partners. The communication implementation will be over whichever port the partner provides with its federation member metadata for the SOAP assertion retrieval. SSL/TLS encryption of this communication is a requirement... Implementing a client certificate authentication mechanism, as well as digital signing of the request and assertion, is also possible.

6.2.6.6. User Interface

In the role of a Relying Party, the ITFIM module accepts runtime input from client browsers in the form of HTTP requests with specific data in them. Internal transformation framework

Extensible Stylesheet Language Transformation (XSLT) sheets parse the HTTP request, following any defined business rules. Then the information passes in the form of HTTP header data to the protected resource for user identification and authorization. Customization of the output is according to the protected application's particular architecture.

Table 38: ITFIM Interfaces

Description	Purpose	Port	Connecting Component(s)	Zone	FW Rule?
Bootstrap/RMI Port	The address for the bootstrap function and the port number for the Java(TM) Remote Method Invocation (RMI) connector in the application server		DM	INT	N
SOAP Port	The port for the Simple Object Access Protocol (SOAP) connector in the application server		DM		N
Registration SSL Port	The port for accessing the registration application via SSL		HS	INT	N
Server to Server	Node management from deployment manager		DM	INT	N
IBM Tivoli Directory Server Ports	ITAM provides a layer of protection through its reverse proxy, the last step of the federation sign-on process. It accepts the browser request with an assertion artifact and passes the request to the ITFIM external interface to retrieve the assertion from the VAAFI partner. After the authentication/ authorization process is completed, the response to the client browser is sent back with a redirect or proxy to the requested federation protected resource.		WS	DMZ	Y

6.2.6.7. Hardware Architecture

The ITFIM is a complex system of interconnected components and has the superimposed resource requirements for the hardware platform of its subcomponents. The Federated Identity

Manager Core services are one of the most demanding, according to the software documentation. The hardware configurations may vary depending on the architecture chosen for the production environment, but the general recommendation is to use at least a 4-way unit with 8 GB of RAM (roughly 2 GB per CPU available for applications' use). The basis of this recommendation is mainly the software prerequisites, but it also supports the scalability and load balancing schemes embedded in the planned solution. Because the VAAFI project is in its beginning stage, no statistics for transaction volumes or concurrent user numbers from other similar projects for comparison are available. The only numbers available for this stage of the project are the AA numbers for current and total users over the last 4-5 years. Although the expected number of initial users on VAAFI is not significant, the system plans include the ability to process 300 concurrent sessions at any given time (depending on usage patterns). If saturation of the current systems occurs, it should easily scale up to the needed capacity by adding hardware to the existing cluster.

6.2.6.8. VA_Hash and TransactionID JAR

A custom jar file in the WebSphere class path creates two data elements that are added to the STS user. The first is a SHA-1 hash of the csid and uid that is stored as the va_eauth_hash. The second is a SHA-2 hash of the csid, uid, and AuthenticationInstant that is stored as the va_eauth_transactionID. This jar file is called from the XSLT for federation partners.

6.2.6.9. Federation Partner XSLT Files

Each federation partner has an XSLT file that takes the CSP attributes from the users TFIM session and updates it with Portal Strategy attributes when available. The XSLT also can override the Target location when certain circumstances arise—such as sending a user to the AccessVA 3POB Confirmation page or to the AccessVA Error pages.

6.2.6.9.1. DSLogon Partner XSLT

The XSLT performs the following actions:

- Attributes from the CSP are extracted into variables
- The authorization and subject_authorization attribute is transformed to JSON.
- The cspid attribute is created by concatenation of the csid + '_' + uid.
- The hash attribute is a hash created on the csid + uid + issueInstant attributes.
- The birthdate attribute is created by stripping off the 1st 10 characters and then appending T00:00:00 to normalize the time/timezone since we only want to deal with the date portion of the birthdate but also keep it in the YYYY-MM-DDTHH:MM:SS format.
- A query string is created for the PortalStrategy REST call.
- If the LOA attribute is greater than 1, the PortalStrategy REST call is performed by sending the query to AccessVA PortalStrategy services to determine if the cspid exists.
- If it does not exist, the PortalStrategy service will attempt softboarding.

- If the REST service returns VDS information, the attributes are added or updated if they exist.

6.2.6.9.2. PKICSP Partner XSLT

The XSLT performs the following actions:

- Attributes from the CSP are extracted into variables
- The cspid attribute is created by concatenation of the csid + '_' + uid.
- The hash attribute is a hash created on the csid + uid + issueInstant attributes.
- The PIV full name is parsed from the uid attribute and is parsed using the rules defined in the Federal Common Policy Certificate Policy section 3.1.1:
 - firstname lastname
 - firstname initial. lastname
 - firstname middlename lastname
 - firstname lastname GQ
 - firstname initial. Lastname GQ
 - firstname middlename lastname GQ
 - firstname lastname #####
 - firstname initial. Lastname #####
 - firstname middlename lastname #####
 - firstname lastname GQ #####
 - firstname initial. Lastname GQ #####
 - firstname middlename lastname GQ #####
 - firstname lastname (affiliate)
 - firstname initial. lastname (affiliate)
 - firstname middlename lastname (affiliate)
 - firstname lastname GQ (affiliate)
 - firstname initial. Lastname GQ (affiliate)
 - firstname middlename lastname GQ (affiliate)
 - firstname lastname ##### (affiliate)
 - firstname initial. lastname ##### (affiliate)
 - firstname middlename lastname ##### (affiliate)
 - firstname lastname GQ ##### (affiliate)
 - firstname initial. Lastname GQ ##### (affiliate)
 - firstname middlename lastname GQ ##### (affiliate)
- The CAC full name is parsed from the uid attribute using the following rule:
 - lastname.firstname.middlename
- A query string is created for the PortalStrategy REST call.

- If the LOA attribute is greater than 1, the PortalStrategy REST call is performed by sending the query to AccessVA PortalStrategy services to determine if the cspid exists.
- If it does not exist, the PortalStrategy service will attempt softboarding.
- If the REST service returns VDS information, the attributes are added or updated if they exist.

6.2.6.9.3. Symantec Partner XSLT

The XSLT performs the following actions:

- Attributes from the CSP are extracted into variables.
- The zipcode attribute is extracted and transformed using the following rules:
 - If zipcode character length is 9, reformat to add a '-' to end with #####-####
 - If zipcode character length is 5 or 10, return unchanged
- The cspid attribute is created by concatenation of the csid + '_' + uid.
- The birthdate attribute is created by stripping off the first 10 characters and then appending T00:00:00 to normalize the time/timezone since we only want to deal with the date portion of the birthdate but also keep it in the YYYY-MM-DDTHH:MM:SS format.
- A query string is created for the PortalStrategy REST call.
- If the LOA attribute is greater than 1, the PortalStrategy REST call is performed by sending the query to AccessVA PortalStrategy services to determine if the cspid exists.
- If it does not exist, the PortalStrategy service will attempt softboarding.
- If the REST service returns VDS information, the attributes are added or updated if they exist in the CSP.

6.2.6.10. Attribute Retrieval Service

The VAAFI Team created the Attribute Retrieval Service to support the EVSS architecture. The EVSS architecture requested minimal attributes in headers from VAAFI and made web service calls to other data sources on an as-needed basis. The VAAFI Team created the Attribute Retrieval service to replace the Registry and Authorization Attribute Web Service (RAAWS). RAAWS provided many important Veteran attributes and traits to eBenefits. DMDC announced that RAAWS would not continue. Instead, the attributes that RAAWS provided would be sent in the SAML assertion of a DS Logon user to VAAFI.

The Attribute Retrieval Service provided a means for partners to make web service calls as they had always done using session information and obtain the attributes they formerly received through RAAWS. This web service had frequent issues and EVSS decided that they wanted the data in headers rather than through a web service call.

6.2.6.11. Portal Strategy Implementation

VAAFI implements the Target Portal Strategy, as shown in Figure 46, which includes third-party onboarding and is described in the AccessVA detailed design, Section 6.2.10.

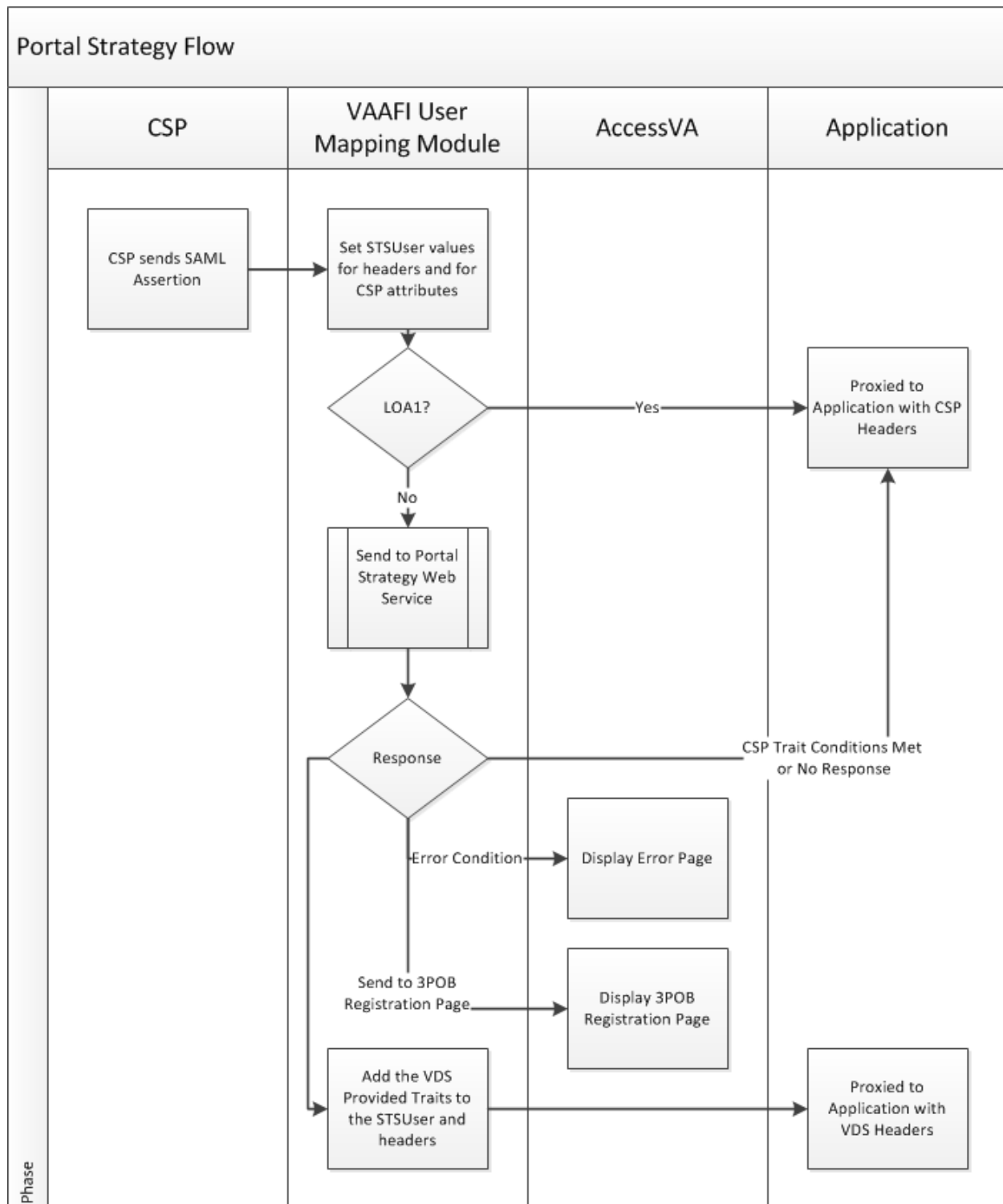


Figure 46: Portal Strategy Log On

1. A SAML Assertion comes from a partner CSP.
2. The VAAFI Headers are set to the CSP Attribute Values. The va_eauth and the va_csp headers are both set to the CSP values at the start of the flow. This should help ensure that the NOT_FOUND value is not sent when VAAFI has a value for that attribute. It

will also allow for a comparison of when the CSP and IAM values differ (SPEC1980.2.1.7). This is also when the custom code is called to generate the hash and the Transaction ID.

3. The Level of Assurance for the SAML Assertion is checked to be greater than.
4. If the result of the check is positive (the LOA=1) process follows the legacy VAAFI flow and the user's browser is directed to its intended target with the CSP traits.
5. If the result is negative (the LOA >1) attributes are transmitted to the Portal Strategy Web Service.
6. Based on the response of that web service, users are either forwarded to the application with the CSP traits, sent to the 3POB Registration page, sent to an error page or the VDS attributes are mapped into the STSUser and they are forwarded to the application with the VDS traits.

The Portal Strategy Web Service is a custom service developed to interact with the AcS VDS and Provisioning services. It is located on the VAAFI WebLogic instance that also contains AccessVA. The CSP attributes are sent to the Portal Strategy Web Service, which is described in Figure 47 and Figure 48.

Portal Strategy Web Service Flow

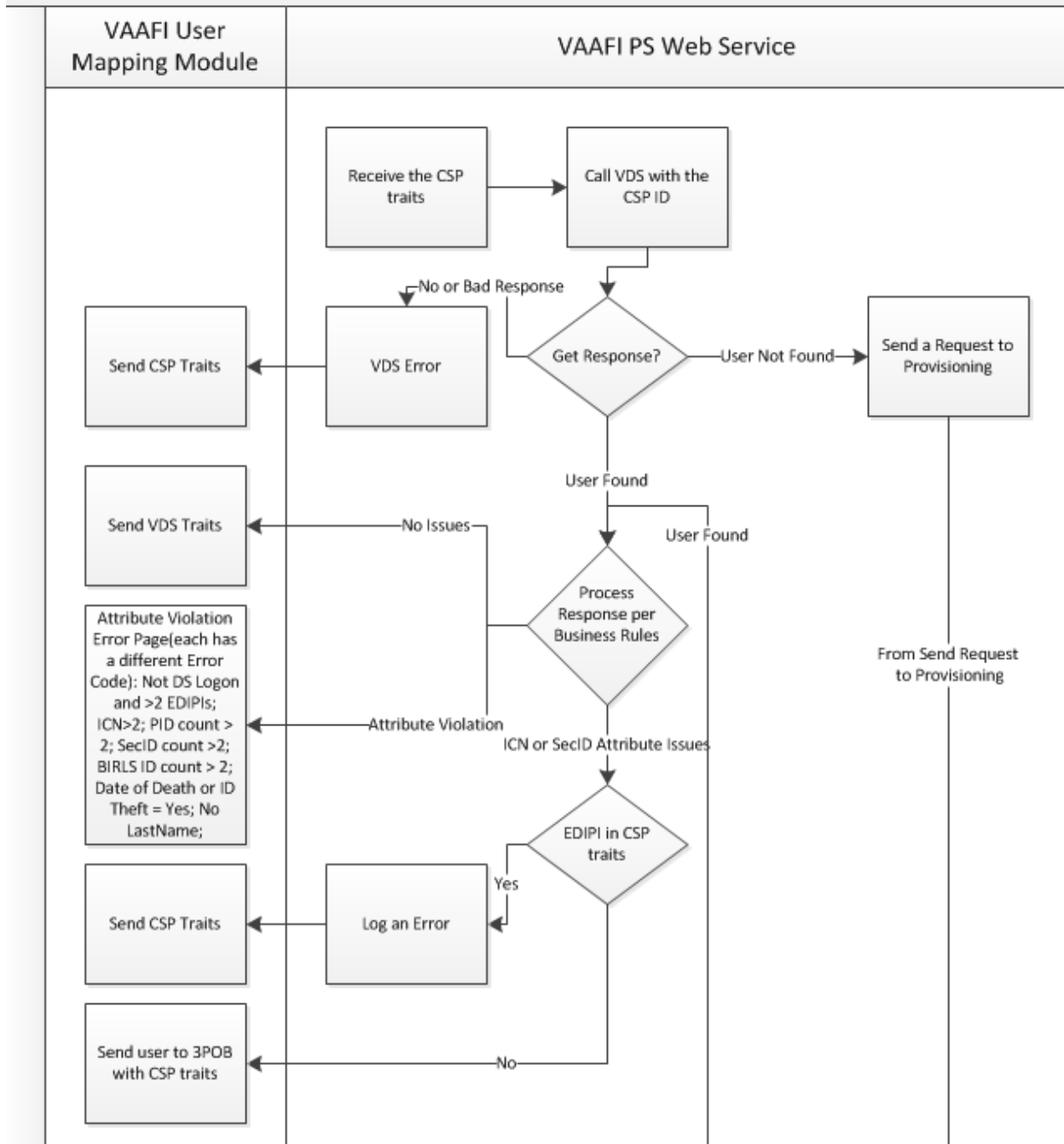


Figure 47: Portal Strategy Web Service (Part 1 of 2)

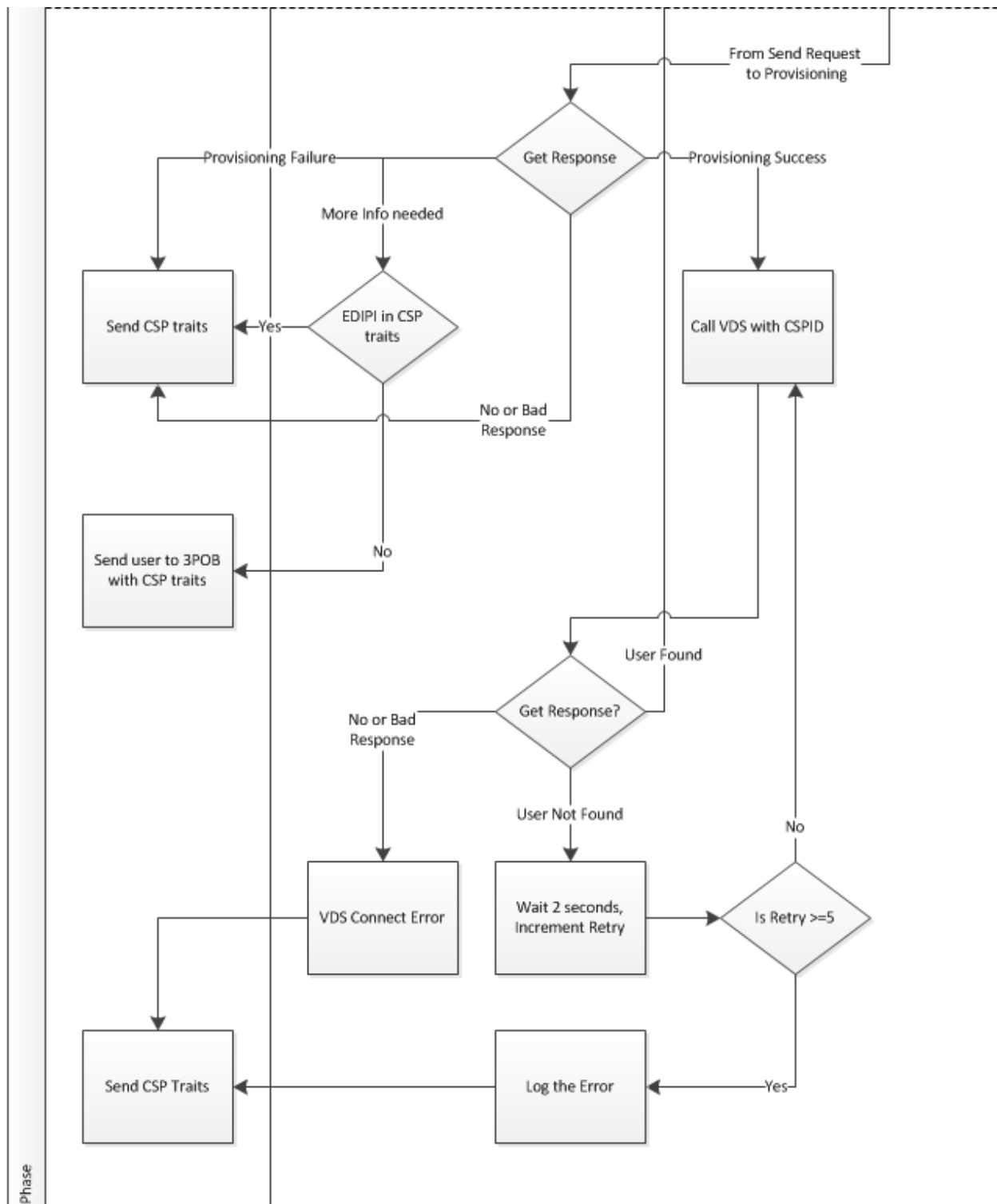


Figure 48: Portal Strategy Web Service (Part 2 of 2)

7. The Portal Strategy Web Service will have one of four possible responses.
 - a. The first possible response is to send the CSP traits to the application. This can occur if an AcS or other dependent service is down, if there is no response or a bad response

from the web service, or if there is not enough information to complete provisioning but an EDIPI was sent as a trait (per Figure 47 and Figure 48).

- b. The second possible response is to display an error message displays. Per the tables in the i5 RSD, the follow condition invoke an error page:
 - No Last Name;
 - Identity Theft Flag; and
 - A Date of Death.
- c. The third possible response is to require more information to complete provisioning. This can occur when either not enough traits are present to perform a Search or not enough traits are present to perform an Add. The VAAFI User Mapping Module directs the user's browser to the 3POB Confirmation page with the CSP traits. From there the user follows the Third Party On-boarding flow in Section 6.2.10.8.
- d. The fourth possible response is that the person is known, data traits are mapped into the STSUser and those traits are sent to the Target application for that individual. VDS provided traits are sent in preference to the CSP traits except the EDIPI in the case of DS Logon. The VDS traits are:
 - First Name;
 - Middle Name;
 - Last Name;
 - Prefix;
 - Suffix;
 - Street1;
 - Street2;
 - Street3;
 - City;
 - State;
 - ZIP;
 - Country;
 - Phone number;
 - Social Security Number;
 - Date of Birth;
 - Gender;
 - Email;
 - CSPID;
 - SecID;
 - ICN;
 - BIRLS File Number;
 - CorpDB Participant ID;
 - MyHealtheVet IEN;

- DoD EDIPI;
 - Date of Death; and
 - Identity Theft Flag.
- 8. The VAAFI User Mapping Module calls the Portal Strategy Web Service sending the encrypted CSP traits.
- 9. The Portal Strategy Web Service decrypts the traits.
- 10. The Portal Strategy Web Service calls VDS using the CSPID.
- 11. There are three possible responses from VDS: it gives a bad or no response, it finds the user, or it does not find the user.
 - a. If VDS gives a bad or no response, an error logs and a message sent back to the FIM to send the CSP traits.
 - b. If VDS finds the user, the data is processed per the rules provided in the iRSD :
 - 1) If an Attribute Violation conditions exists, the User Mapping Module receives an Error Message. Currently these conditions include:
 - No Last Name;
 - Identity Theft Flag; and
 - A Date of Death.
 - 2) If the ICN or SecID is not sent, two actions are possible:
 - a) If the CSP traits included an EDIPI, an error logs and a message goes back to FIM to send the CSP traits.
 - b)) If the CSP traits do not include an EDIPI, a message goes back to FIM to send the user to the Third Party Onboarding Confirmation page for reprovisioning with more data.
 - c. If VDS does not find the user, it sends a request to Provisioning.
- 12. Four possible Provisioning responses are possible: no response or a bad response; a Provisioning failure; requiring more information to complete Provisioning; or successful Provisioning.
 - a. If Provisioning returns a bad or no response, an error logs and a message sent back to the FIM to send the CSP traits.
 - b. If a Provisioning failure occurs, an error logs and a message goes to the FIM to send the CSP traits.
 - c. If the Provisioning response is that it needs more information, one of two things will happen:
 - 1) If the CSP traits included an EDIPI, error is logged and a message is sent back to FIM to send the CSP traits.
 - 2) If the CSP traits do not include an EDIPI, a message is sent back to FIM to send the user to the Third Party On-boarding Confirmation page to be re-provisioned with more data.
 - d. If the Provisioning response is a success, VDS is called with the CSPID.

13. The VDS response will be similar to the responses in Step 11. The possible responses are: it gives a bad or no response; it finds the user; or it doesn't find the user.
- a. The first two possibilities follow the previous logic.
 - b. A response is that it did not find the user creates a retry counter and sets it to 1, or if the counter already exists, incremented it by 1. The web service will wait 2 seconds between retries.
 - 1) If the counter equals 5 or greater, a message is sent back to FIM to send the CSP traits.
 - 2) If the counter is less than 5, another call will be made to VDS.

NOTE: The number of retries and wait time of two seconds are initial values that may change based on testing or other criteria.

To implement the VA IAM Portal Strategy, the VAAFI team created the following components:

- A VAAFI Federation Security Token Service (STS) User Mapping Module that is CSP-specific. These mapping modules contain logic to:
 - Extract attributes from the assertions provided by the CSPs;
 - Implement the business logic related to treatment of LOA1 credentials;
 - Encrypt the CSP provided attributes and send them to web service;
 - Then, depending on the response from the web service, perform more logic to send the user either to the application they requested, the Third Party On-Boarding Confirmation page or an error page.
- The VAAFI WebLogic instances that host AccessVA will also host a Portal Strategy Web Service. This web service will implement most of the Portal Strategy business logic, make the calls to the AcS VDS and Provisioning Services, process the responses to those calls and communicate the responses back to the User Mapping Module.
- The user mapping module will process the VDS results returned from the Portal Strategy Web Service and augment them to the user attribute set to make them available to SSOe integrated applications as HTTP headers per Table 23: HTTP Headers.

6.2.6.12. Error Pages

Error pages display user-friendly error messages when Federation operations fail. Error pages are dynamic pages that Tivoli Federated Identity Manager generates. ITFIM uses the following information to generate web pages:

Template files

XML or HTML files that are provided with Tivoli Federated Identity Manager and contain elements, such as fields, text, or graphics, and sometimes macros, that are replaced with information that is specific to the request or to provide a response to the request.

Page identifiers

Event information that corresponds to one or more template files. Each page identifier corresponds to a specific event condition, such as a specific error or a condition in which a message or a form must display. To create an event page, page identifiers are mapped to one or

more template files. The mapping function allows multiple page identifiers to point to the same template file.

Message catalogs

Text that replaces macros in the template files.

When a request is received, ITFIM generates the appropriate error page as follows:

1. During the processing of a request, an error occurs and requires a response.
2. Read template files and page identifiers from the file system.
3. Replace macros in the template files with values appropriate for the needed response.
4. Generate an appropriate error page.
5. Display the generated error page.

All error pages an end user of the system can encounter are located within a specific directory in FIM and a specific directory in WebSEAL. These directories will be modified to include AccessVA stylesheets and resources and then every error page in these directories will be modified to include the AccessVA header and footer information.

The following table lists the error templates and their corresponding error pages:

Table 39: Error Templates and Error Pages

Page identifier (Event)	Description	Template file	Error Code
/saml/error_parsing_soap_response.html	Displayed when there is an error encountered when the service provider attempts to retrieve the Assertion from the identity provider's SOAP endpoint	/saml/error_parsing_soap_response.html (/liberty/error_parsing_soap_response.html)	20
/saml/unknown_ip.html	Displayed when an unknown identity provider is encountered	/opt/IBM/FIM/pages/C/saml/saml/unknown_ip.html	40
/saml/no_return_token.html	Displayed when there is no return token	/opt/IBM/FIM/pages/C/saml/saml/no_return_token.html	60
/saml/missing_context_attribute.html	Displayed when the required context attribute is not present	/opt/IBM/FIM/pages/C/saml/missing_context_attribute.html	60
/saml/missing_config_parameter.html	Displayed when a required SPS configuration item is missing	/opt/IBM/FIM/pages/C/saml/missing_config_parameter.html	60
/saml/invalid_response.html	Displayed when an invalid response message is encountered	/opt/IBM/FIM/pages/C/saml/invalid_response.html	60

Page identifier (Event)	Description	Template file	Error Code
/saml/no_ip_post_page.html	Displayed when the identity provider does not have a post page	/opt/IBM/FIM/pages/C/saml/no_ip_post_page.html	60
/saml/cannot_exchange_for_sp.html	Displayed when there is an error encountered during the token exchange	/opt/IBM/FIM/pages/C/saml/cannot_exchange_for_sp.html	60
/saml/could_not_perform_local_auth.html	Displayed when an error is encountered when the EAI header is returned to WebSEAL	/opt/IBM/FIM/pages/C/saml/could_not_perform_local_auth.html	60
/saml/invalid_request.html	Displayed when a request is not valid	/opt/IBM/FIM/pages/C/saml/invalid_request.html	60
/saml/invalid_ip_request.html	Displayed when an identity provider provides an invalid request	/opt/IBM/FIM/pages/C/saml/invalid_ip_request.html	60
/saml/ip_response_invalid.html	Displayed when identity provider response is invalid	/opt/IBM/FIM/pages/C/saml/ip_response_invalid.html	60
/saml/unauth_user.html	Displayed when the running user has not authenticated	/opt/IBM/FIM/pages/C/saml/unauth_user.html	60
/saml/unknown_sp.html	Displayed when an unknown service provider is encountered	/opt/IBM/FIM/pages/C/saml/unknown_sp.html	60

Page identifier (Event)	Description	Template file	Error Code
/saml/cannot_exchange_for_resource.html	Displayed when there is an error encountered during the token exchange	/opt/IBM/FIM/pages/C//saml/could_not_retrieve_assertion.html	70
/saml/could_not_retrieve_assertion.html	Displayed when the service provider could not get the Assertion from the Response or from the SOAP back channel	/opt/IBM/FIM/pages/Csaml/could_not_retrieve_assertion.html	70
/saml/ip_post_to_sp.html	Displays the POST HTML Form when the identity provider posts the SAMLp Response to the service provider	/opt/IBM/FIM/pages/C/saml/ip_post_to_sp.html	NA
/saml/ip_post_to_sp.html	Displays the POST HTML Form when the identity provider posts the SAMLp Response to the service provider	/opt/IBM/FIM/pages/C/saml/ip_post_to_sp.html	NA
/oauth20/user_consent.html		/opt/IBM/FIM/pages/C/oauth20/user_consent_custom.html	
/oauth20/user_response.html		/opt/IBM/FIM/pages/C/oauth20/user_response_custom.html	
/oauth20/user_error.html		/opt/IBM/FIM/pages/C/oauth20/user_error.html	

6.2.6.13. Logging

The trace.log file captures FIM logs. The FIM logs the raw SAML assertion and other FIM processing, logging the CSP traits. With the Portal Strategy, the CSP traits will usually be replaced with traits sent from VDS. To capture both the CSP and the VDS traits in a single log entry and ease reporting, logging will include the STSUser transform.

6.2.7. IBM Tivoli Access Manager WebSEAL – Reverse Proxy

IBM Tivoli Access Manager WebSEAL is the resource security manager for web-based resources in a Tivoli Access Manager secure domain. WebSEAL is a high-performance, multi-threaded Web server that applies fine-grained security policy to the protected web object space. WebSEAL can provide SSO solutions and incorporate back-end web application server resources into its security policy.

Within VAAFI, the WebSEAL component will be the reverse proxy to control access to the AA. Working in conjunction with the ITAM and the ITFIM modules, WebSEAL will provide the security layer to process access requests coming from users authenticated by the VAAFI CSPs. WebSEAL manages access decisions on junction points. Junction points are locations within a namespace that allow fine-grained access control rules to be defined on a per junction basis.

6.2.7.1. Software

The WebSEAL software subsystem requires several base modules, including the WebSEAL software itself, to operate. The module versions include the following:

- IBM Tivoli Access Manager Runtime 6.1.1;
- IBM Tivoli Access Web Security Runtime 6.1.1;
- IBM Tivoli Access Manager WebSEAL 6.1.1;
- IBM Tivoli Directory Client 6.2;
- IBM GSKIT 7.0.4.36; and
- IBM JRE 1.5.0.

6.2.7.2. Dependencies

The installation of the IBM Tivoli Access Manager WebSEAL components requires the following modules be present:

- IBM JRE 1.5;
- The IBM GSKIT must be version 7.0.4.36 or higher; and
- Linux package dependencies (see 6.2.2.2).

6.2.7.3. Database Design

The IBM Tivoli Access Manager WebSEAL requires a database for storing policy configuration and administration. This database is actually a replicated database from the IBM Tivoli Access Manager Policy Server. This database is a proprietary database implementation. By default, the database resides in /var/PolicyDirector/db.

Table 40: TAM Interfaces

Description	Purpose	Port	Connecting Component(s)	Zone	FW Rule?
User Lookup	WebSEAL uses LDAP to perform user lookups. VAAFI users must exist in the user repository. WebSEAL uses this in making policy decisions regarding access.		LDAP	INT	Y
Application Reverse Proxy	Reverse proxy to the AA		Apps (see individual ICDs)	EXT	Y
User Reverse Proxy	Reverse proxy to the end user		Users	INT	Y
FIM Reverse Proxy	Reverse proxy to the FIM servers		FIM	INT	Y
Policy Replication	Policy database replication with the IBM Tivoli Access Manager Policy Server		TAM	INT	Y
Policy Replication	WebSEAL to TAM		TAM	INT	Y
			TAM	INT	Y
External Authorization Interface (EAS)	Policy Enforcement Point for OAuth 2.0		TFIM STS	INT	Y

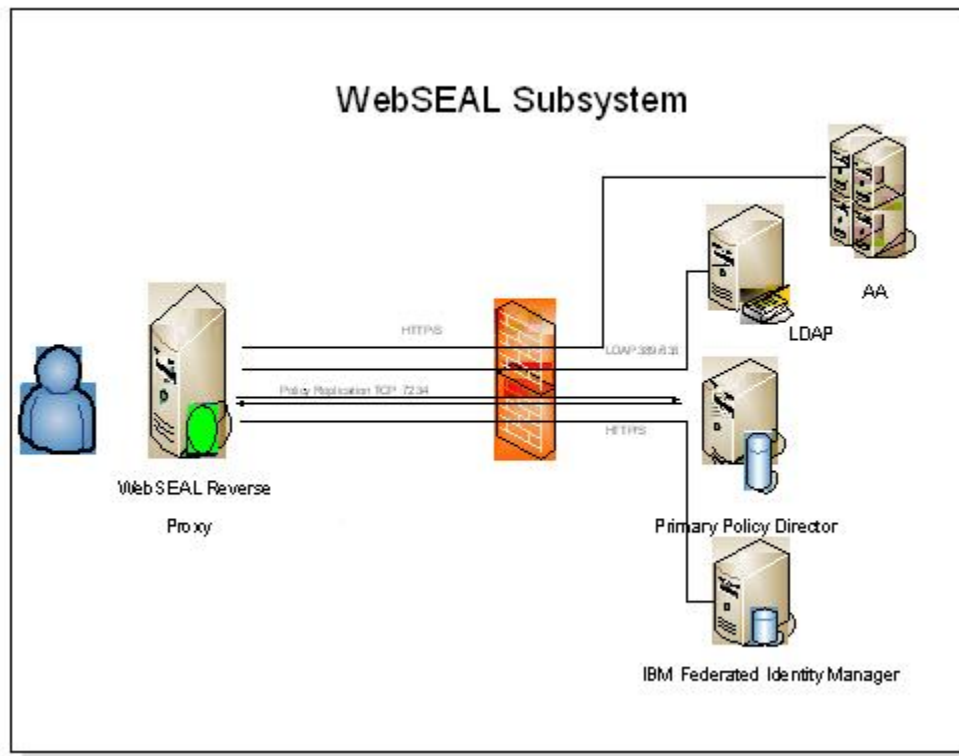


Figure 49: WebSEAL Subsystem

6.2.7.4. External Interfaces

The IBM Tivoli Access Manager WebSEAL component provides reverse proxy capability. The WebSEAL component accepts HTTP and HTTPS requests and upon successful authentication and authorization, proxies the requests to internal resources. WebSEAL appears as a Web server to clients and appears as a Web browser to the WebSEAL junction back-end server that is being protected.

6.2.7.4.1. User Interface

The IBM Tivoli Access Manager has several utilities for managing the configuration of the Access Manager product. Most functions can execute through command line or web-based tools.

6.2.7.4.2. Shared Object Space for WebSEAL

In a heavy load environment, replicating front-end WebSEAL servers to provide better load-balancing and fail-over capability is advantageous. With replicated front-end WebSEAL servers, each server must contain an exact copy of the web space, the junction database, and the dynurl database.

The Shared Object Space is a feature of the Tivoli TAM Policy Server software suite that allows the database objects required by the TAM Policy Server to be created and stored at a central location rather than being created on each WebSEAL individually. Implementing this feature helps prevent errors that would occur through the repeated creation of objects in each of the WebSEAL servers. Currently, twelve WebSEAL servers are in each of the environments

(Production and Contingency Site). As traffic through the WebSEAL servers increases, additional WebSEAL servers can be added.

Within VAAFI, the WebSEAL object space is set up as a shared object space as /WebSEAL/eauth.va.gov. All the policy objects (ACL/POP/AuthzRules) as well as extended attributes for junctions are attached to this shared object space. All junctions created to back-end application servers will appear under this shared object space.

OAuth solution manages the Authorization Endpoint and the Token Endpoint under the/FIM object space. The full path of these object spaces are:

Authorization Endpoint: [REDACTED]

This endpoint will be configured to propagate the va_eauth_uid

Access Token Endpoint: [REDACTED]

6.2.7.5. WebSEAL Instance Naming

The WebSEAL instance names should be the short server name in all lower case. WebSEAL will then concatenate the following: -webseald-
<longservername> (for example, vaauswebiamws80-webseald- vaauswebiamws80.vha.med.va.gov).

6.2.7.6. Logging

WebSEAL logging is through the logs in the /var/ibm/tivoli/common/DPW/logs/www-
<servername>/log directory on each server. Splunk collects and indexes the logs in this directory, which are the Agent, Referrer and Request logs. The Requests logs are especially important for reporting purposes. These logs record every request a VAAFI user makes and logs many of the attributes sent with the request. Configuration file /opt/pdweb/etc/webseald-
<<ws-name>>>.conf identifies which attributes to log. During Increment 4, CAR will begin consuming VAAFI logs. CAR requirements will include many of the attributes for Portal Strategy, specifically all the VAAFI headers sent as part of the Portal Strategy.

6.2.8. PKI CSP Detailed Design

The PKI CSP consists of an implementation of the ITFIM very similar to the VAAFI Service Provider. Connectivity between components is similar as is the redundancy between components. The diagram in below describes the interaction between the components.

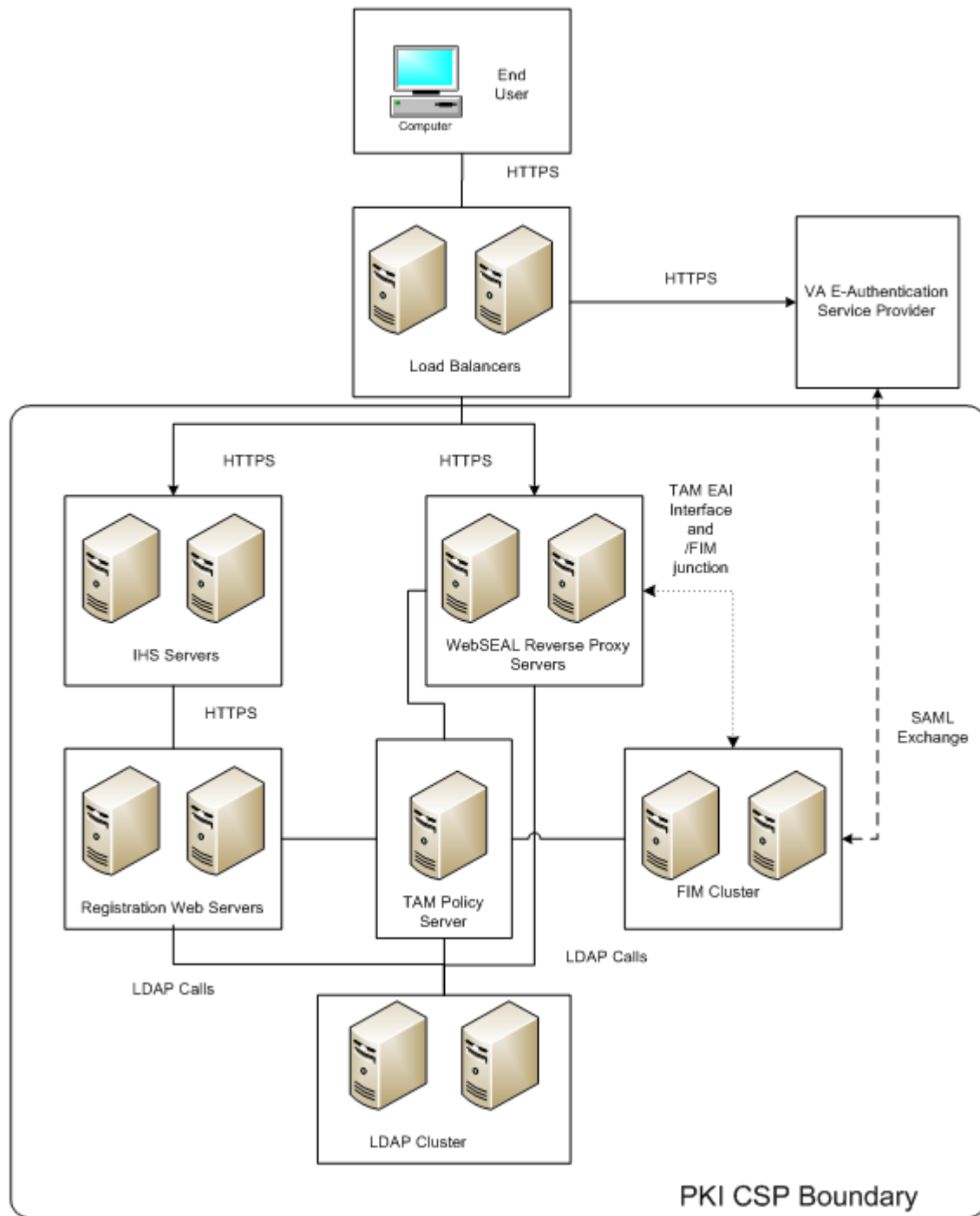


Figure 50: PKI CSP Detailed Design

There are two sides to this sub-system; the first is authentication, the second is registration. The first part of the subsystem is where user authentication takes place when the user uses the CAC or PIV card to access the VAAFI CSP. In this case, the user accesses the system by confirming that the certificate's signer is trusted, the certificate is not expired, and that the CA has not revoked the certificate. Once complete, and after the user selects a link to a protected application a SAML

1.1 browser artifact and SAML 1.1 assertion is created by the ITFIM and a browser artifact is sent back to the user's browser. The user will send the SAML browser artifact to the VAAFI Service Provider (SP) using the SAML browser post profile and the SP retrieves the assertion from the VAAFI PKI CSP by returning the SAML artifact in exchange for the corresponding SAML assertion.

In the registration case, the user accesses the [REDACTED] web site through the IBM HTTP Server (IHS), which is a front-end for the IBM WebSphere Application Server hosting the PKI Registration Application. The WebSphere Application Server is a WebSphere cluster running the PKI Registration Application, which accesses the user's PKI certificate from a smart card. It then reads the Distinguished Name from the certificate and adds the user's DN to the ITAM user store.

6.2.8.1. PKI CSP Authentication

The authentication half of the system used the exact same versions and installations as the Service Provider except that Session Management Server is not used to track sessions. Instead, stickiness setting in the F5 Load Balancer keeps a user's session tied to the same WebSEAL for the duration of the user's session. The following sections detail the LDAP, WebSphere, TAM, ITFIM, and WebSEAL implementations:

- IBM Tivoli Directory Server – User Repository (see Section 6.2.1)
- IBM Tivoli Access Manager (see Section 6.1.3)
- IBM WebSphere Deployment Manager (see Section 6.1.4)
- IBM Tivoli FIM (see Section 6.1.7)
- IBM Tivoli Access Manager WebSEAL – Reverse Proxy (see Section 6.1.8)

In this case of the PKI CSP, ITFIM configuration is as an Identity Provider (IdP) (also known as a Credential Service Provider) rather than as a Service Provider.

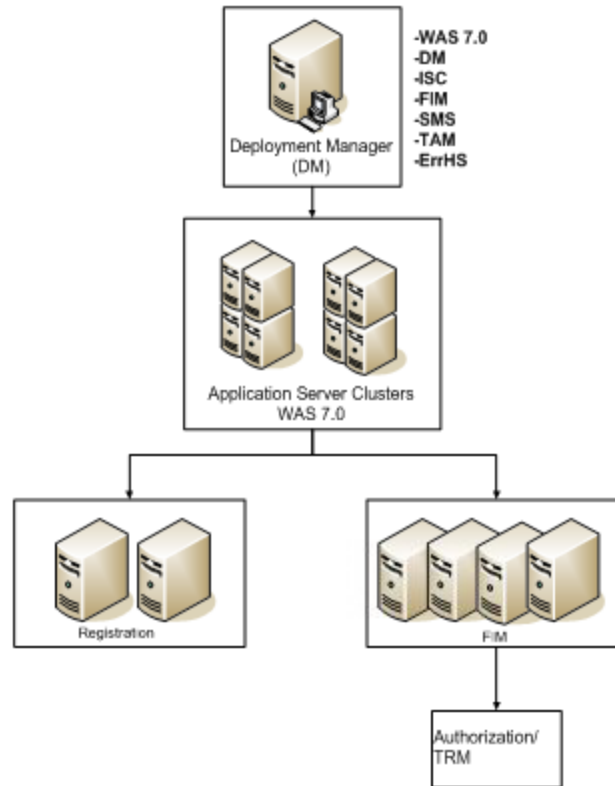


Figure 51: PKI CSP Deployment Manager

6.2.8.1.1. Cluster Naming

The clusters in each environment should follow the naming conventions detailed below (naming is case sensitive):

- VAAFI-PKI-REG-cluster
- VAAFI-PKI-FIM-cluster
- VAAFI-IHS-cluster

6.2.8.1.2. ITFIM Domain Naming

The ITFIM domain in each environment follow the naming conventions detailed below (naming is case sensitive):

- VAAFI-SP
- VAAFI-PKI-CSP

6.2.8.2. PKI CSP Registration

CACs and PIVs are registered with the VAAFI CSP by reading the Distinguished Name (DN) from the PKI certificate contained on the card and creating an ITAM user with that DN in the IBM Tivoli Directory Server (ITDS) LDAP directory. The ITAM WebSEAL reads that same DN from the smart card, and successfully authenticates ITAM users found in the ITDS LDAP with identical DN to those read from the smart cards used to authenticate to the VAAFI CSP. Both the VAAFI PKI Registration Application and VAAFI WebSEAL maintain identical key

databases containing the public key chains of trusted Certificate Authorities (CAs). Only PKI credentials that these CAs sign will be able to register with and use the VAAFI CSP.

6.2.8.3. Registration Servers

To use their CAC or PIV card for authentication, end users will first have to register their CAC or PIV card. This process adds the DN from their smart card to the ITAM user store.

6.2.8.3.1. Software

The WebSphere software package is the application server for adding the user's DN to the ITAM user store. WebSphere requires several base modules, including the WebSphere software itself, to operate. The module versions are the following:

- IBM WebSphere(R) Application Server Version V6.1.0.45;
- IBM GSKIT 7.0.4.36; and
- IBM JRE 1.5.0.

6.2.8.3.2. Dependencies

The installation of the ITAM requires the following module to be present:

- Linux dependencies are the same as detailed above.

6.2.8.3.3. Database Design

There is no database.

Table 41: Registration Interfaces

Description	Purpose	Port	Connecting Component(s)	Zone	FW Rule?
Bootstrap/RMI Port	The address for the bootstrap function and the port number for the Java(TM) Remote Method Invocation (RMI) connector in the application server		Node Agents	INT	N
SOAP Port	The port for the Simple Object Access Protocol (SOAP) connector in the application server		Node Agents	INT	N
Registration SSL Port	The port for accessing the registration application via SSL		IHS	INT	N
Server to Server	Node management from deployment manager		Node Agents	INT	N

6.2.8.3.4. Security

Access controls for the registration server include authentication to the console through accounts stored in the ITDS LDAP directory. Communication between the WebSphere server and other components such as the LDAP and the IHS Servers are encrypted. The WebSphere environment is clustered to provide fault tolerance and High Availability.

6.2.8.4. IBM HTTP Server (IHS) Servers

The IHS Servers protect the Registration Server by sitting in the DMZ network, accepting HTTPS connections and acting as a web front-end proxy to the PKI Registration WAS ND application server.

6.2.8.4.1. Software

The IHS software package is the web front-end server for the Registration application. IHS requires several base modules, including the IHS software itself, to operate. The module versions are the following:

- IBM Tivoli IHS Server V6.1.0.45;
- IBM Tivoli WebSphere Application Server Plugin V6.1.0.45;
- IBM Tivoli IBM GSKIT 7.0.4.36; and
- IBM JRE 1.5.0.

6.2.8.4.2. Dependencies

The installation of the ITAM requires the following module be present:

Linux dependencies are the same as listed above.

6.2.8.4.3. Database Design

There is no database.

Table 42: IHS Interfaces

Description	Purpose	Port	Connecting Component(s)	Zone	FW Rule?
Back end communication	HTTPS received from end users browser back to the Registration Server	443	Registration Server	INT	Y
Front end communication	End user SSL communication	443	End user	Internet	Y

6.2.9. IBM WebSphere DataPower Appliances

The IBM WebSphere DataPower XML Security Gateway is a network appliance that simplifies, secures, and accelerates XML and Web Services in VAAFI. It provides a comprehensive set of functions including Transport Layer Security (TLS), WS-Security, XML encryption, XML/SOAP firewall filtering, XML digital signatures, XML schema validation, two-way SSL, XML access control, and detailed logging.

The main goal of the DataPower appliance is to consolidate and simplify integration with partners. Proxying web services through the DataPower appliance reduces the total number of connections that partners are required to make. Instead of a consumer having to make a connection to each producer, making a single connection to the DataPower appliance can allow all services to be consumed. This also ensures that changes made to services (i.e., certificate changes) do not impact clients.

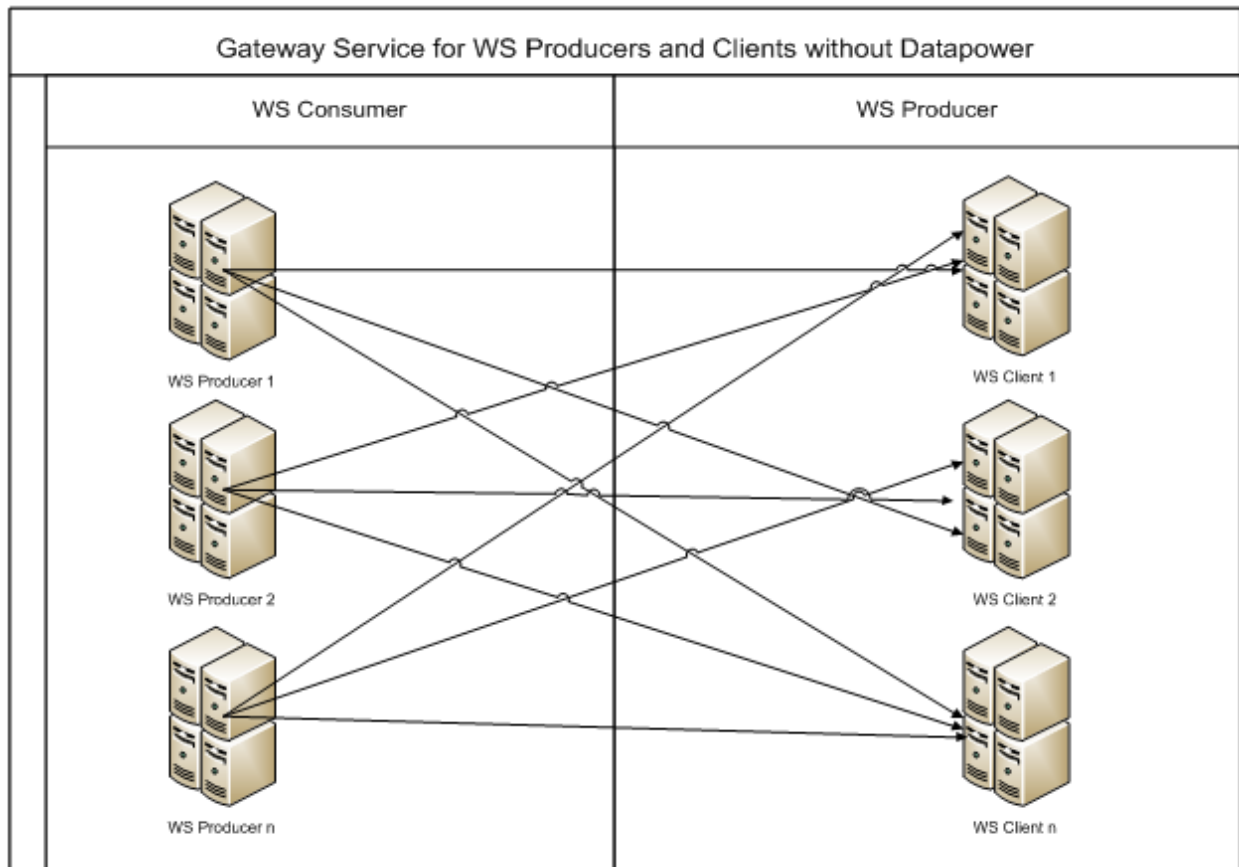


Figure 52: Web Services without DataPower

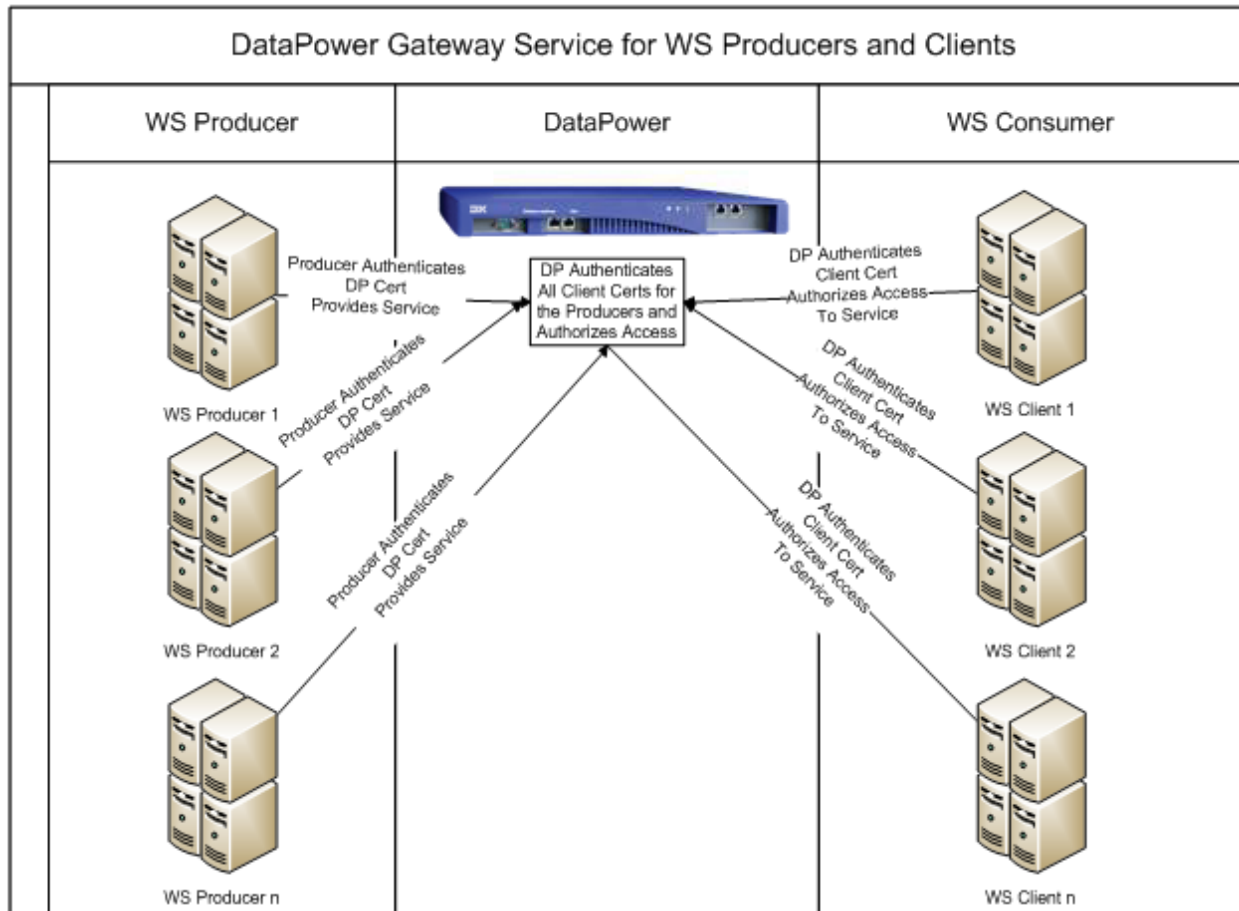


Figure 53: Web Services with DataPower (WS Proxy)

The management of a DataPower device separates into “Application Domains,” which allow the DataPower administrator to create multiple management spaces and assign different individuals responsibility for the configuration of that space. Application domains also allow easier porting of domain configurations among systems without concern affecting the core network.

6.2.9.1. Software

Access to and configuration of the IBM WebSphere DataPower XML Security Gateway hardware appliance is through several interfaces. Access to DataPower appliances can be through a command-line interface, a SOAP XML Management Interface, and a Web-based graphical user interface (WebGUI). The functionality of the system is through its firmware for both operational and administrative actions.

6.2.9.2. Dependencies

The DataPower appliances rely on the load balancer to receive traffic and the Splunk server to capture logs.

6.2.9.3. Interfaces

The DataPower interfaces support connectivity to and from internal VA systems and external partners, and provide the listening port for the Web Management User Interface, defined below. The internal interface can act as a client (initiating TCP connections) or as a listening service (front-end proxy connection point for internal services to access).

Table 43: DataPower Interfaces

Description	Purpose	Port	Connecting Component(s)	Zone	FW Rule?
Web Admin Client	Administration and configuration	████	Admin Server	INT	Y
SSH	SSH administrative interface	██	SSH Client	VA	Y
VA Internal Web Services Inbound	Listens for web service requests	████	Applications	VA	Y
DoD Web Services	Listens for web service requests from DoD	████	DoD	Ext	Y
VA Resource Gateway to DMDC	Provides JavaScript images and CSS from DMDC	████	DoD	Ext	Y
DMDC Portlets		██	DMDC Web Service Producers	Ext	Y

6.2.10. AccessVA Detailed Design

AccessVA consists of a Spring MVC based servlet J2EE application. The following section details the design of those components.

accessva-X.X.X.X.ear is a label that is the generic version of the deployed file name, where the X.X.X.X would be replaced with a version number.

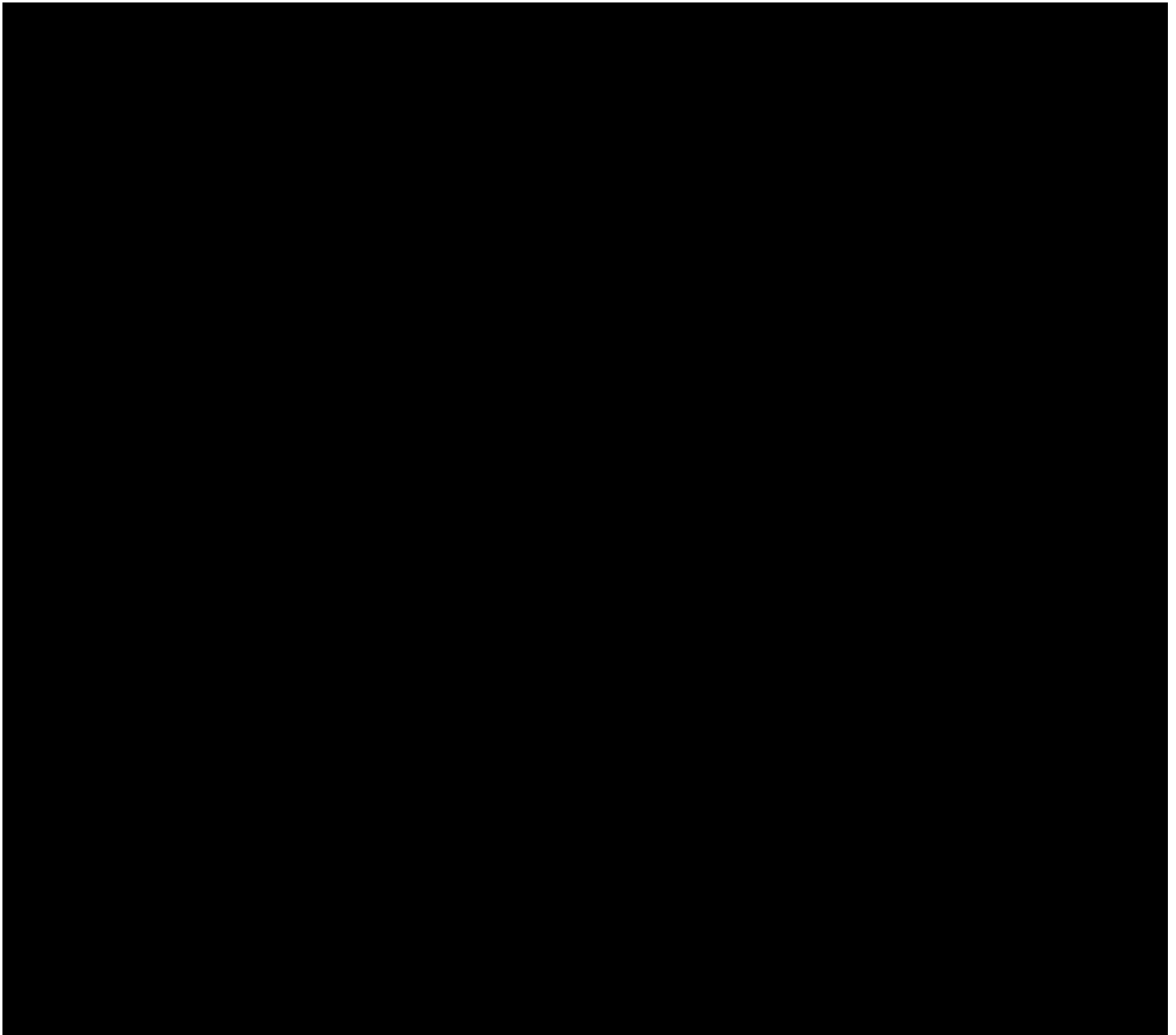


Figure 54: AccessVA Matrix Classes Diagram Example

6.2.10.1. Mobile Responsive (Bootstrap Framework)

AccessVA has been designed using a framework that is responsive, that is, the layout of web pages adjusts dynamically, taking into account the characteristics of the device used (desktop, tablet, mobile phone).

The Bootstrap framework was used because of its design philosophy of “responsive first” design. It is Open Source and currently the most-starred project on GitHub.

Bootstrap comes with several JavaScript components that AccessVA uses, including Modal Dialogs, Form Validation, and Navigation-Breadcrumb Bars.

The Bootstrap framework provides a uniform, modern appearance for formatting text, tables, and form elements in the various pages of the AccessVA subsystem.

6.2.10.2. Spring Framework

The Spring Framework is a lightweight, modular solution for building applications. Spring provides Inversion of Control (IoC) and Dependency Injection features for configuring AccessVA.

The AccessVA properties and matrix are both injected into the application at startup of AccessVA.

AccessVA also uses the Spring MVC web application framework to manage incoming requests and redirect the proper response. It integrates directly with template-based rendering technologies. AccessVA uses JSP.

Spring MVC Controllers interpret user input and transform it into a model that is represented to the user by the view. The model object is transformed into a JSP request attribute and can be utilized in the JSP page.

6.2.10.3. JavaServerPages (JSP) + Tiles

AccessVA uses JSP to create dynamic web pages. JSP allows Java code and certain predefined actions to be interleaved with static web markup content, with the resulting page being compiled and executed on the server to deliver a web page to the browser.

Apache Tiles is a template composition framework used to define page fragments that can be assembled into a complete page at runtime. AccessVA uses Tiles to provide a consistent look and feel across the AccessVA application by providing header, content, and footer templates.

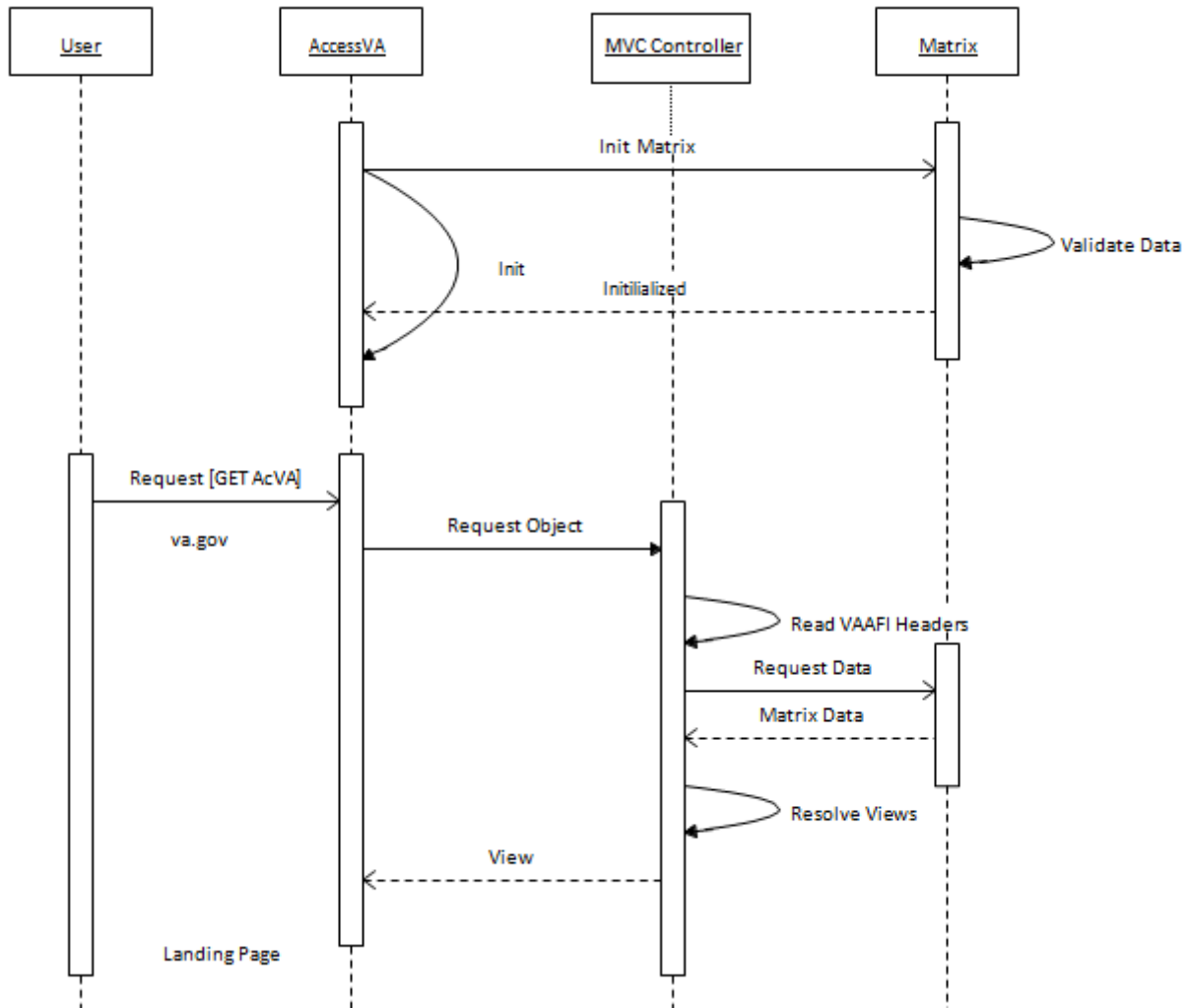


Figure 55: Request Controller Sequence Diagram Example

The Request Controller sequence begins with an HTTP GET request to AccessVA, which accesses the matrix objects containing the relationships and data for protected applications, CSPs and Levels of Assurance (LOAs).

6.2.10.4. CSP Register Sequence

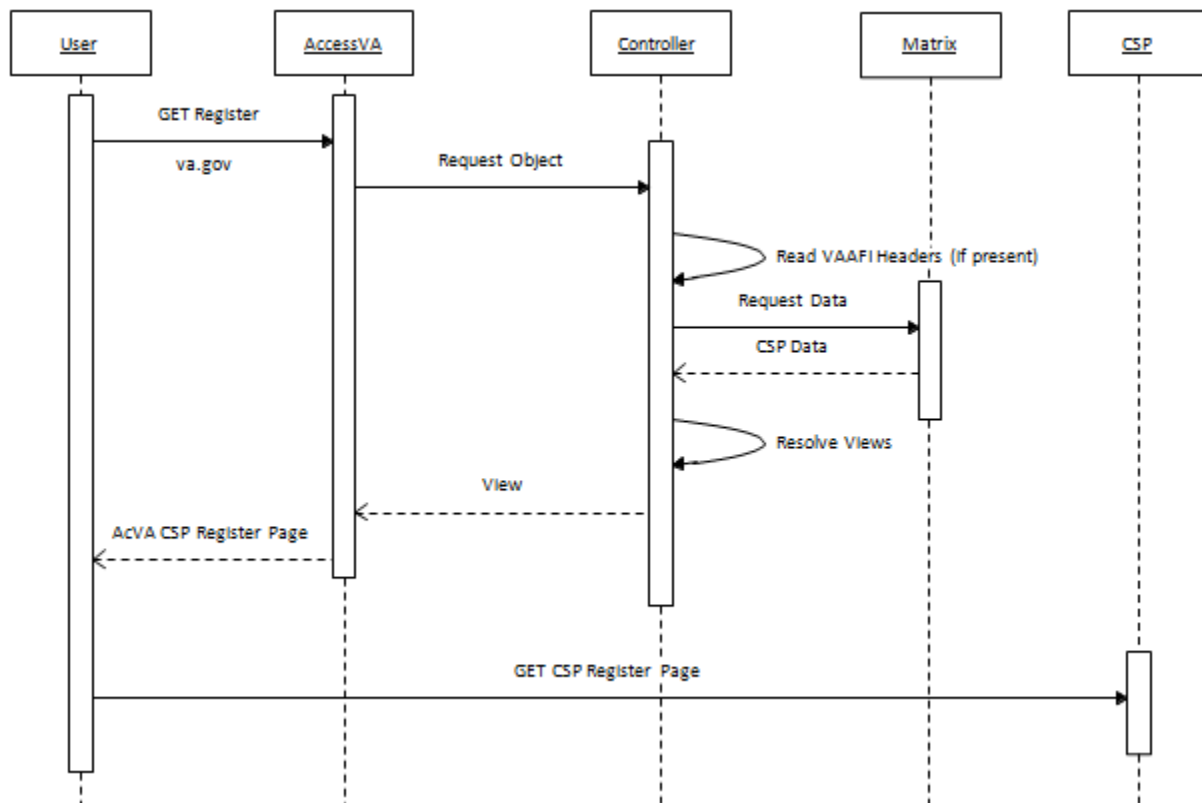


Figure 56: CSP Register Sequence Diagram Example

The CSP Register sequence begins with an HTTP GET request to the Spring MVC controller to get registration links in AccessVA. The user then requests the CSP registration page via the links provided by the AccessVA matrix.

6.2.10.5. CSP Login Sequence

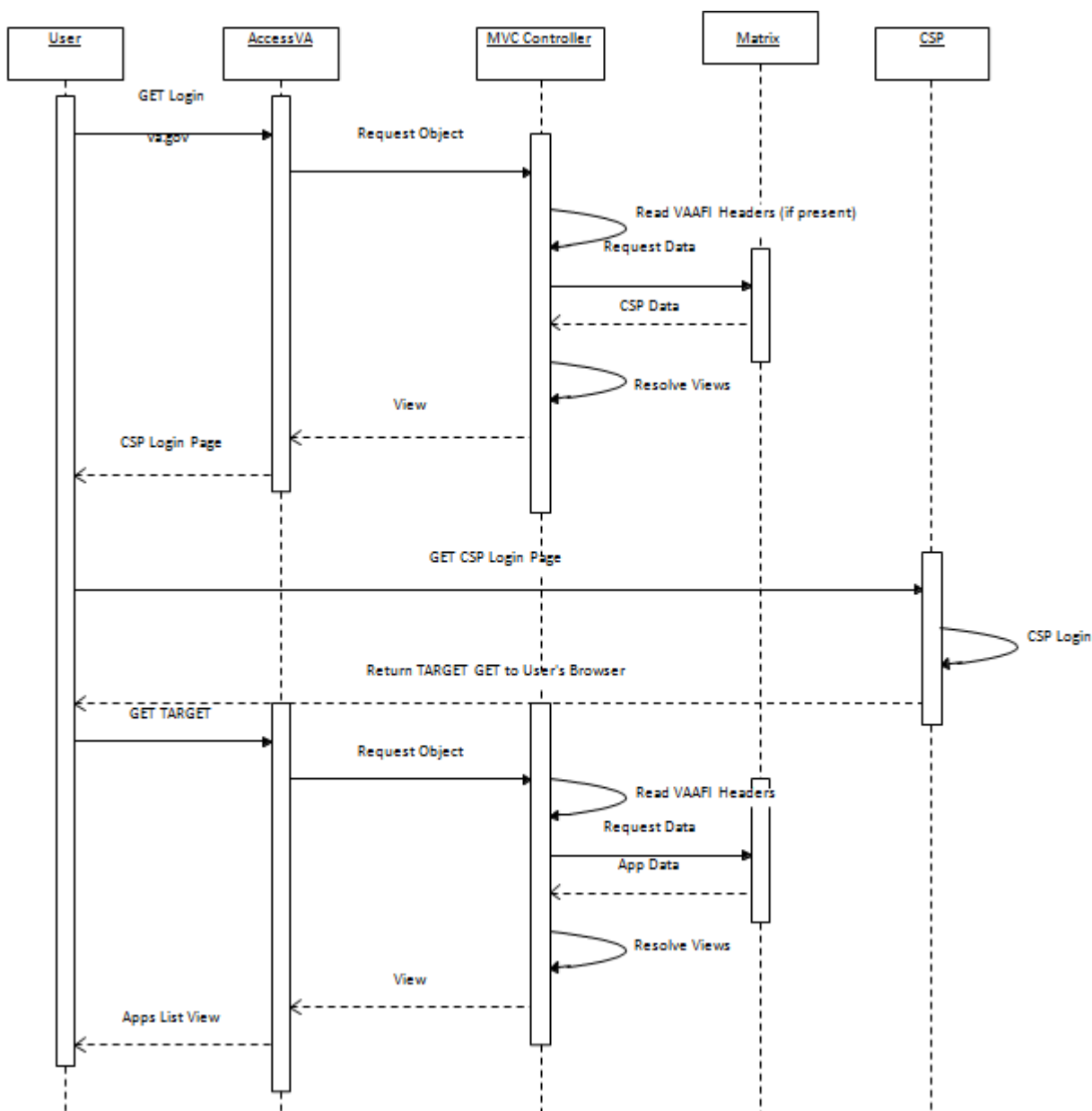


Figure 57: CSP Login Sequence Diagram Example

The CSP Login sequence allows the user to choose a protected application and CSP combination, and then authenticate with that CSP and redirect to the chosen application. Additionally, after logging in with the CSP, the user can return to AccessVA as authenticated and select other applications without needing to authenticate again. During this sequence, a banner page will display as defined in Section 8.4, below. The VAAFI SSOe integration is not in this sequence, but is in place to communicate with the CSP and establish the SSO session before the user is redirected back to AccessVA or to another target protected application.

6.2.10.6. CSP Logout Sequence

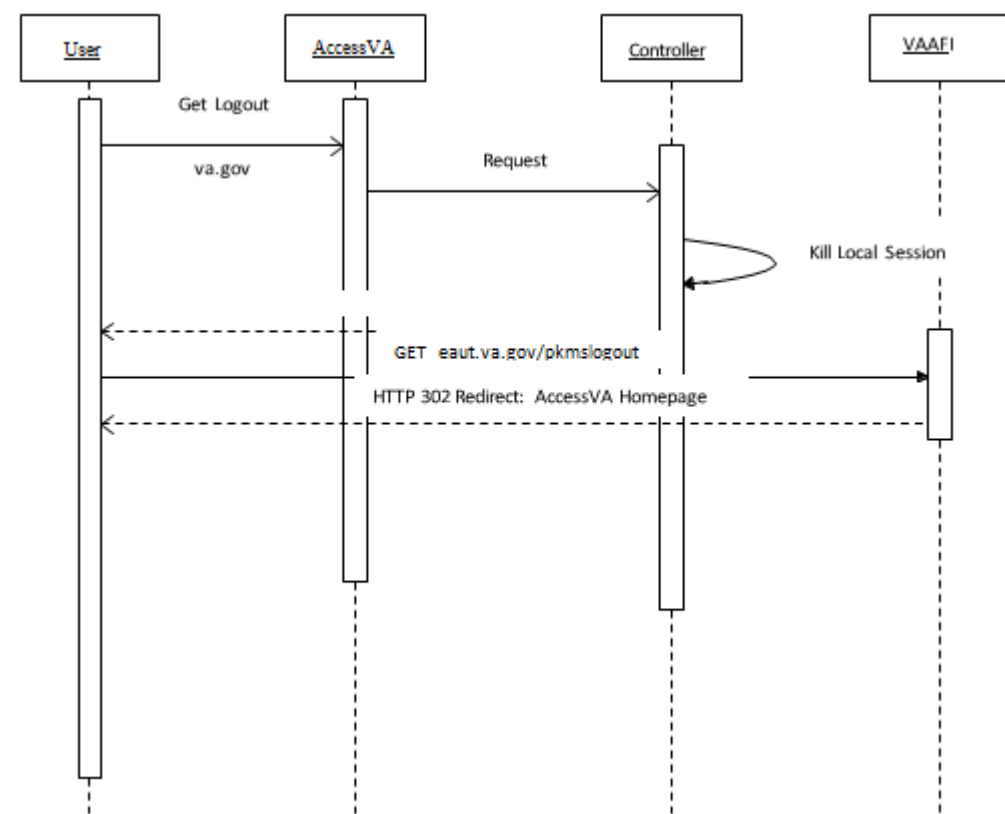


Figure 58: CSP Logout Sequence Diagram Example

The CSP Logout sequence begins with an HTTP GET request to the Spring MVC controller to request a logout in AccessVA. AccessVA kills the local session and redirects the user to VAAFI PKMSLogout. The PKMSLogout kills the SSO session and redirects the user to the unauthenticated AccessVA homepage.

6.2.10.7. AccessVA Widget

The AccessVA Widget provides the application with a standard HTML link that the application may embed on its home page. Documentation and recommendations on how to integrate the widget are in the VAAFI Application Integration Guide. Once the user clicks the button on the applications page, the widget appears as a pop-up in an iFrame. The user can select the CSP to authenticate and log in to the application. The CSPs that display are determined by the AccessVA Matrix and are those authorized by the application.

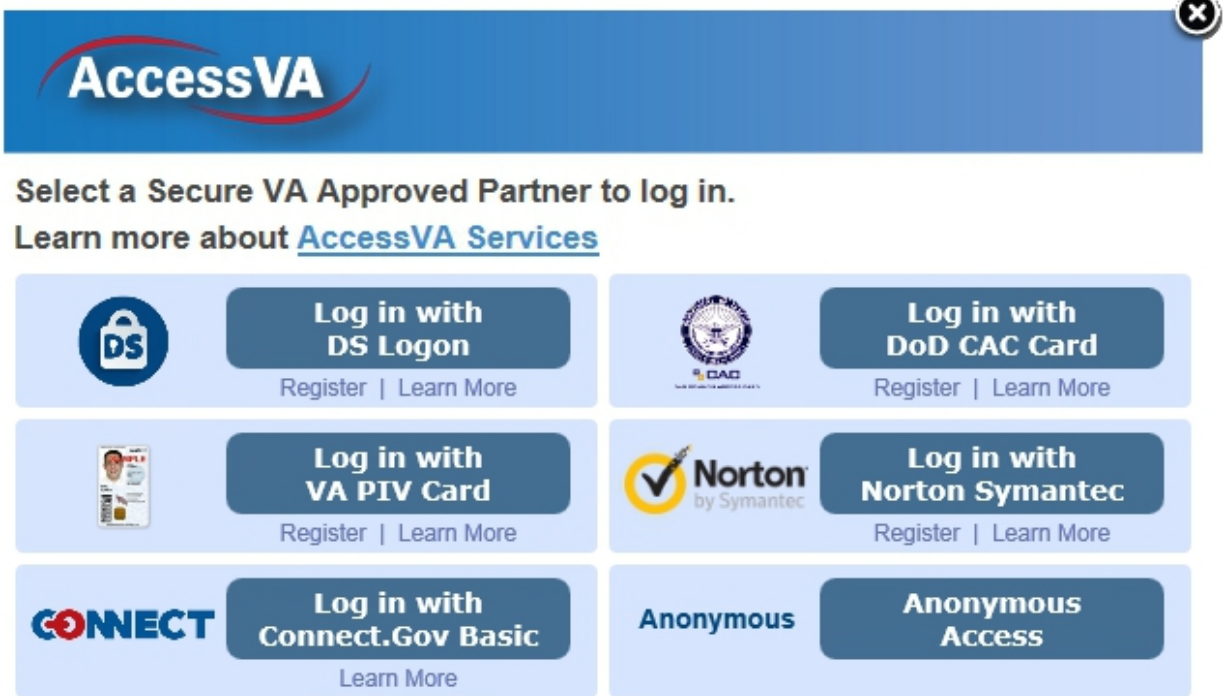


Figure 59: AccessVA Widget

6.2.10.8. Third Party Onboarding Confirmation Page Sequence – Provisioning Not Available

A web service call goes out to the Provisioning Are-You-There operation. If the provisioning web service is not available, the user is redirected to the target application. If the provisioning web service is available and the user is logging in with a PIV or Symantec credential, the Confirmation page displays with the prefilled form. The user fills in the form and selects **Submit**. The page validates the user-entered data and verifies that the CSP-provided information was not changed. If the user is logging in with a DS Logon or CAC, the confirmation process occurs in the background and the user does not need to interact with a form. The call then goes to the ThirdPartyOnboardingRegistration operation containing the data and its source. On receiving the response, AccessVA logs the user out by calling pkmslogout. If the response is a success and the user is logging in with a PIV or Symantec, the Successful page displays (see Section 8.4.10.1). If the response is a failure and the user is using a PIV or Symantec credential, the Unsuccessful page displays with the Error Code and a TransactionID. An OK button displays on both the Successful and Unsuccessful pages. Selecting the button takes the user to the AccessVA page unauthenticated.

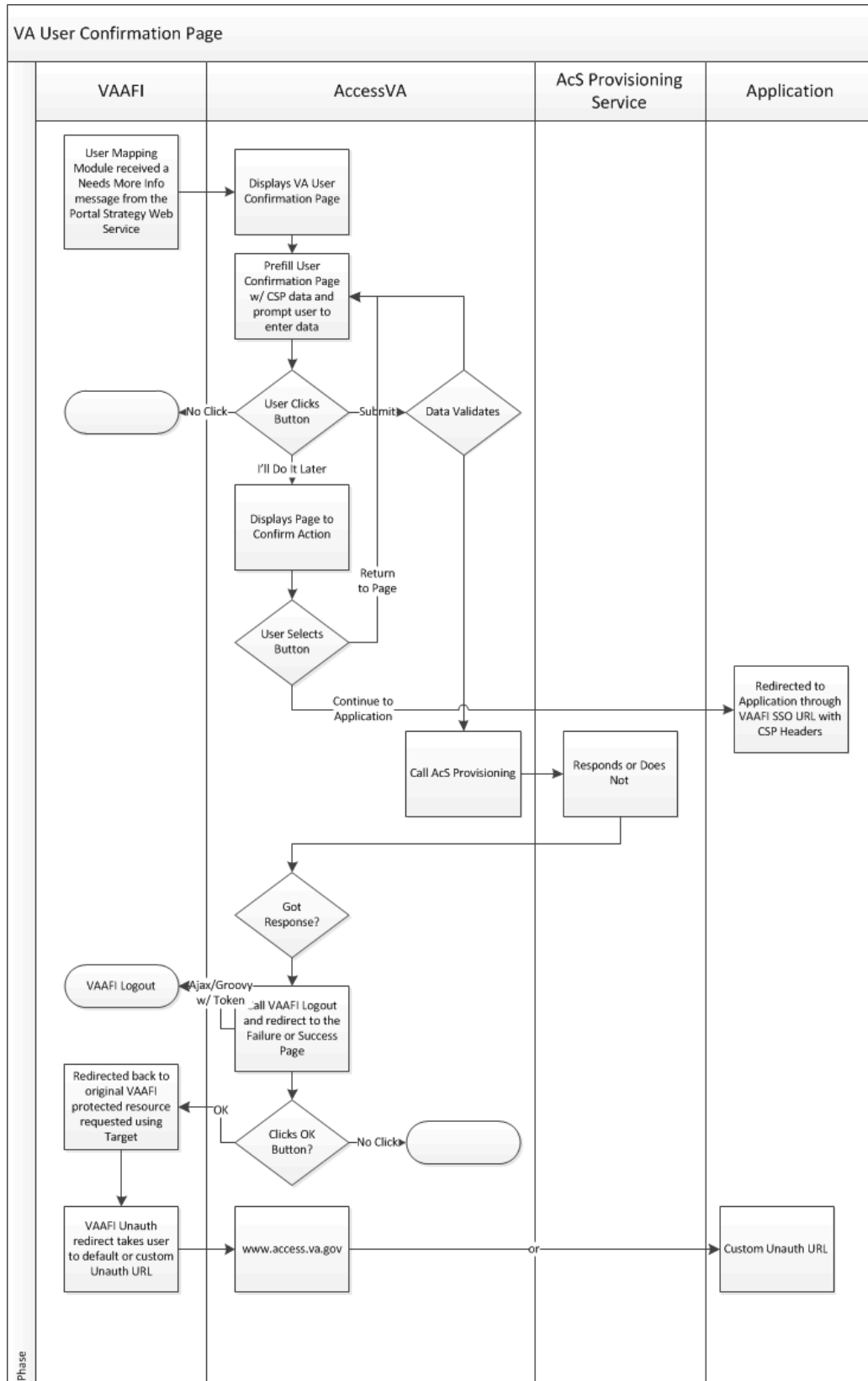


Figure 60: Third-Party Onboarding Confirmation Page Sequence

1. The user enters this flow if the Portal Strategy Web Service returns “More Info Needed.” If the credential being used is a PIV or Symantec, VAAFI sends the user’s browser to AccessVA with the CSP traits and an AppID parameter for the target application the user was attempting to reach through VAAFI. If the credential being used is a DS Logon or CAC, the process continues with Step 0.
2. AccessVA validates the AppID to be sure that it is valid, then sends the user to the VA User Account Confirmation page.
3. The Confirmation page displays with the CSP traits prefilled and uneditable in the form. The user completes the fields. See Section 8.4.10 for sample Confirmation pages.
4. AccessVA waits for the user to select **Submit** or **I’ll Do It Later**.
5. If the user selects **I’ll Do It Later**, VAAFI sends the user’s browser to a confirmation page to ensure that the user wants to delay processing. On this page, the user can select to Return to Confirmation Page or Continue on AccessVA with the CSP traits and an AppID parameter for the target application the user was attempting to reach through VAAFI.
6. If the user selects Submit, the data on the form is client-side validated with the following criteria:
 - The SSN field is a nine-character numeric field.
 - The gender is Male or Female.
 - The phone number is text up to 15 characters.
 - The Country will be selected from a drop down list.
 - The address line 1 is filled in.
 - The ZIP Code is a required field and limited to 10 characters.
7. If the entered data does not pass validation, the page is updated to show which fields failed validation.
8. Provisioning ThirdPartyOnBoardingRegistration is called, the CSP-provided fields from the form are verified—server-side—that they were not changed.
9. CSP traits prefilled the data and the source of the data, either CSP-provided or self-entered. While this occurs, a dynamic graphic spins and a message that the user’s information is being processed displays for those logging with a PIV or Symantec credential.
10. The web service will respond or not.
11. Whether the web service responds or not, the PIV or Symantec user is sent to an AccessVA page that displays the Successful or Unsuccessful message. See Section 8.4.10.1 for the sample Provisioning Successful page. See Section 8.4.10.2 for the sample Provisioning Unsuccessful page.
12. VAAFI calls the VAAFI logout function that kills the user’s session.

NOTE: Once Logout is selected, a session can be automatically recreated by a CSP that is maintaining a session or a browser that has cached the PKI credential.

13. VAAFI waits for the user to select the OK button.

14. Once the user selects **OK**, the user's browser is sent to the authenticated link for the application originally targeted.
15. Because the session ended at VAAFI, the process follows the target application's unauthenticated behavior.
 - a. The default unauthenticated behavior sends the user to AccessVA.
 - b. If the application has elected to follow a custom behavior rather than the default behavior, the application follows that custom behavior. For example, VAAFI redirects eBenefits consumers to DS Logon, not to AccessVA (default).

6.2.10.9. E-Sig Attestation Page

The e-Sig Attestation page provides applications a place to send users to re-authenticate before signing a document with the AcS e-Sig service. AccessVA hosts the page and builds the appropriate links that users follow to re-authenticate at the CSP where they initially authenticated, or for Federal Cloud Credential Exchange (FCCX), to the CSP broker. It then returns the user to the application with a “refreshed” authentication. This process will create a new session for the user and the application is responsible for returning users to their previous session within the application. Figure 61 is a mockup of the e-Sig Attestation page.

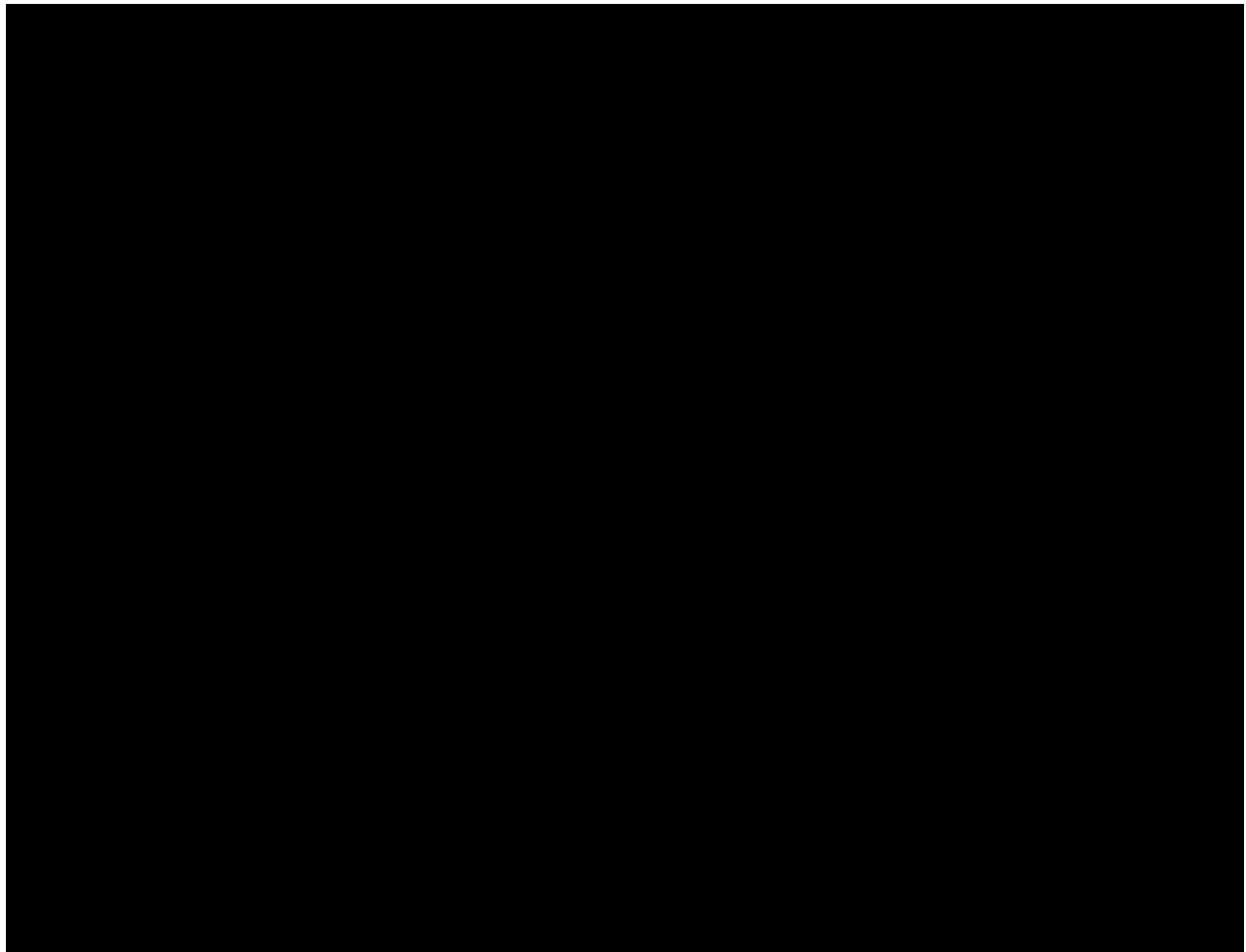


Figure 61: e-Sig Attestation Page

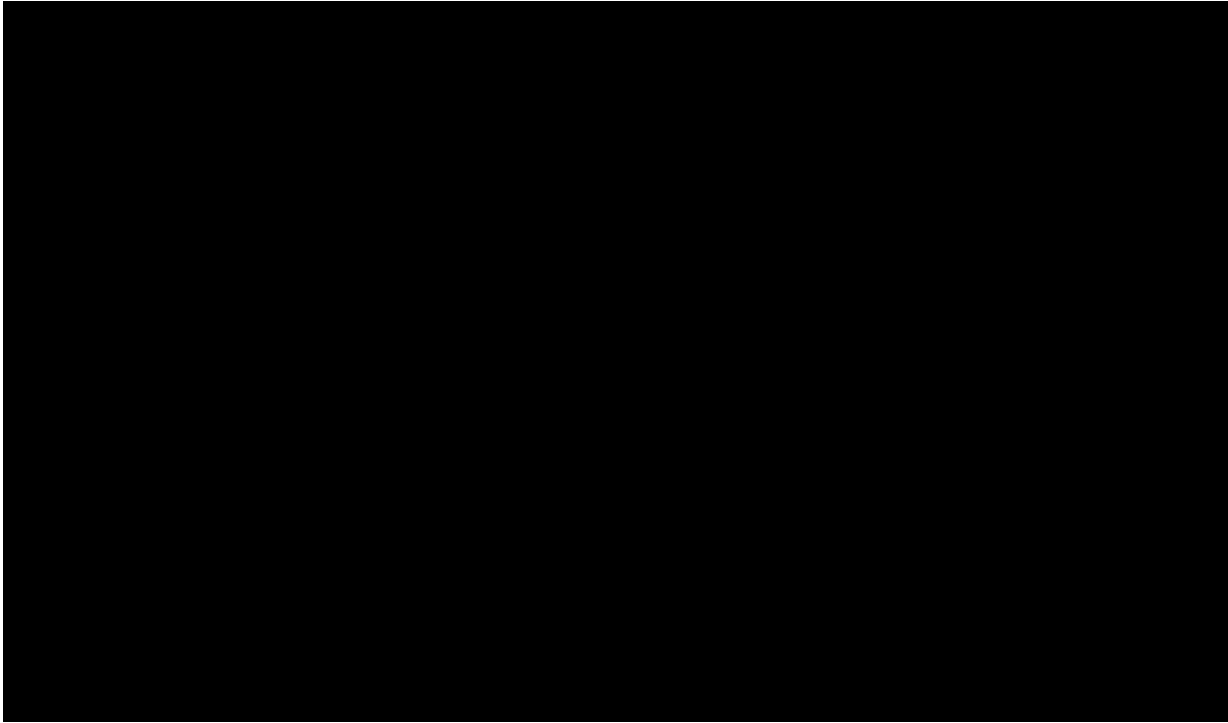


Figure 62: e-Sig Attestation Widget

Applications using this service send authenticated users to the e-Sig Attestation page with one parameter: the URL to which to send the user after re-authenticating if the user selects **Sign**. The following figure illustrates this flow.

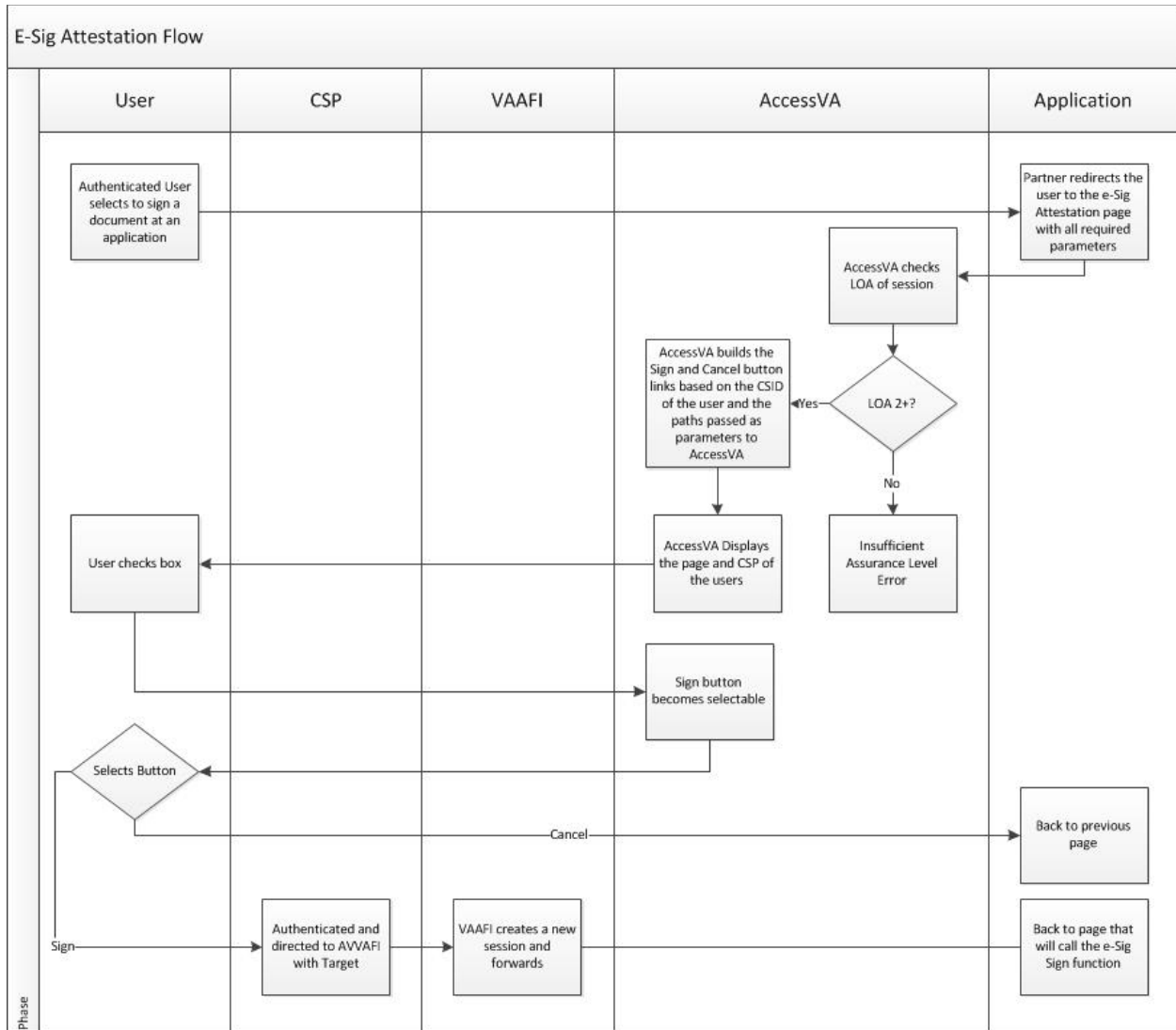


Figure 63: e-Sig Attestation Flow

6.2.11. OAuth Detail Design

The VAAFI OAuth solution provides the following three main customizations to the out-of-the-box TFIM OAuth implementation:

- Client Configuration Provider;
- External Token Cache; and
- Trusted Client Manager.

These customizations are implemented through the TFIM's Open Service Gateway Initiative (OSGI) based pluggable modules. This architecture provides extension points to the base TFIM OAuth implementation and enables customizations. The rest of this Section provides detail design artifacts for the three custom components listed above and the web application. Each section provides the class diagram and screenshot of the plugin configuration file. The plugin

configuration file defines the state information necessary for TFIM to load and execute the plugin.

6.2.11.1. Client Configuration Provider

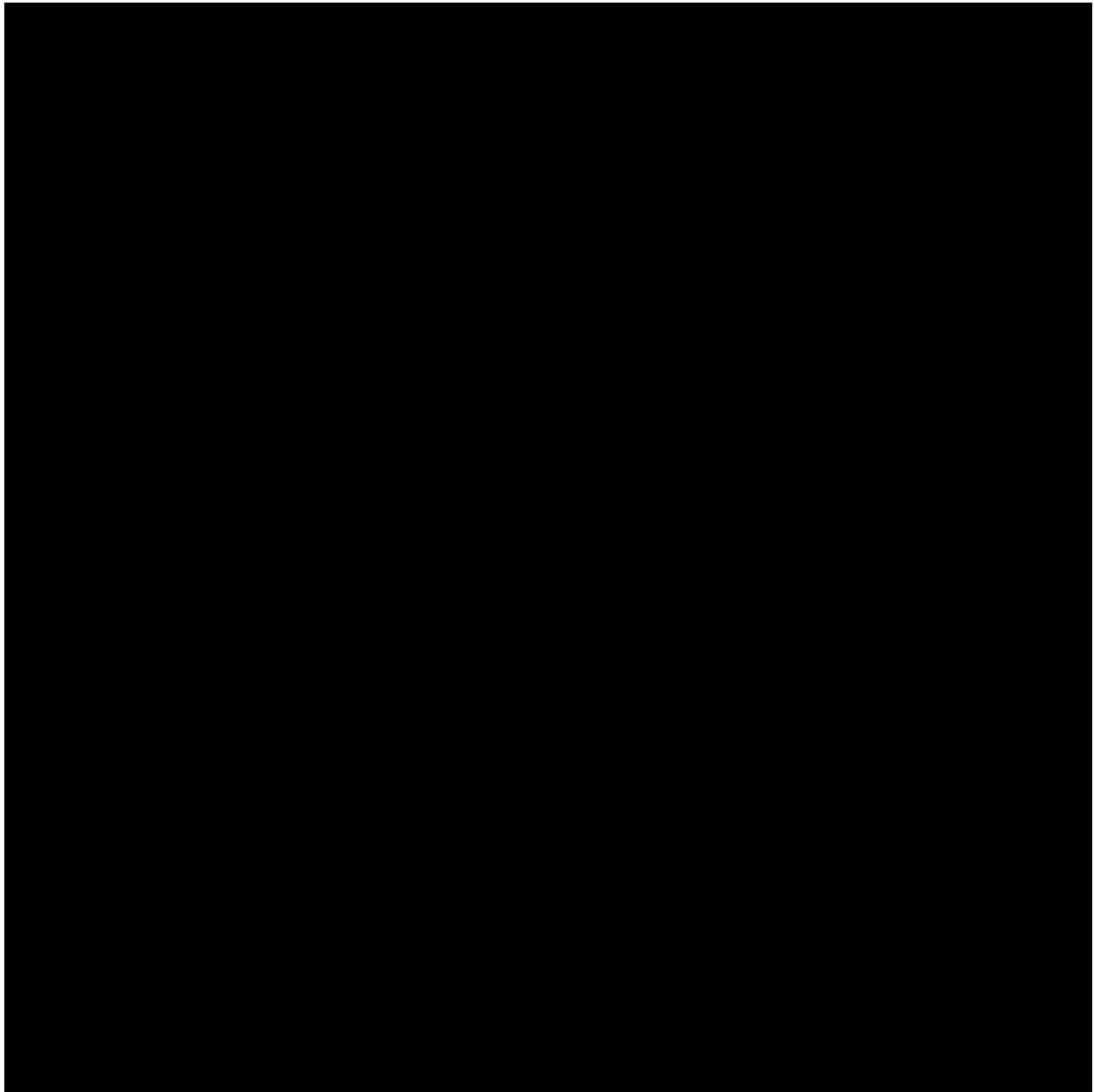


Figure 65: Client Configuration Provider Class and Package Diagram

6.2.11.2. Token Cache Provider

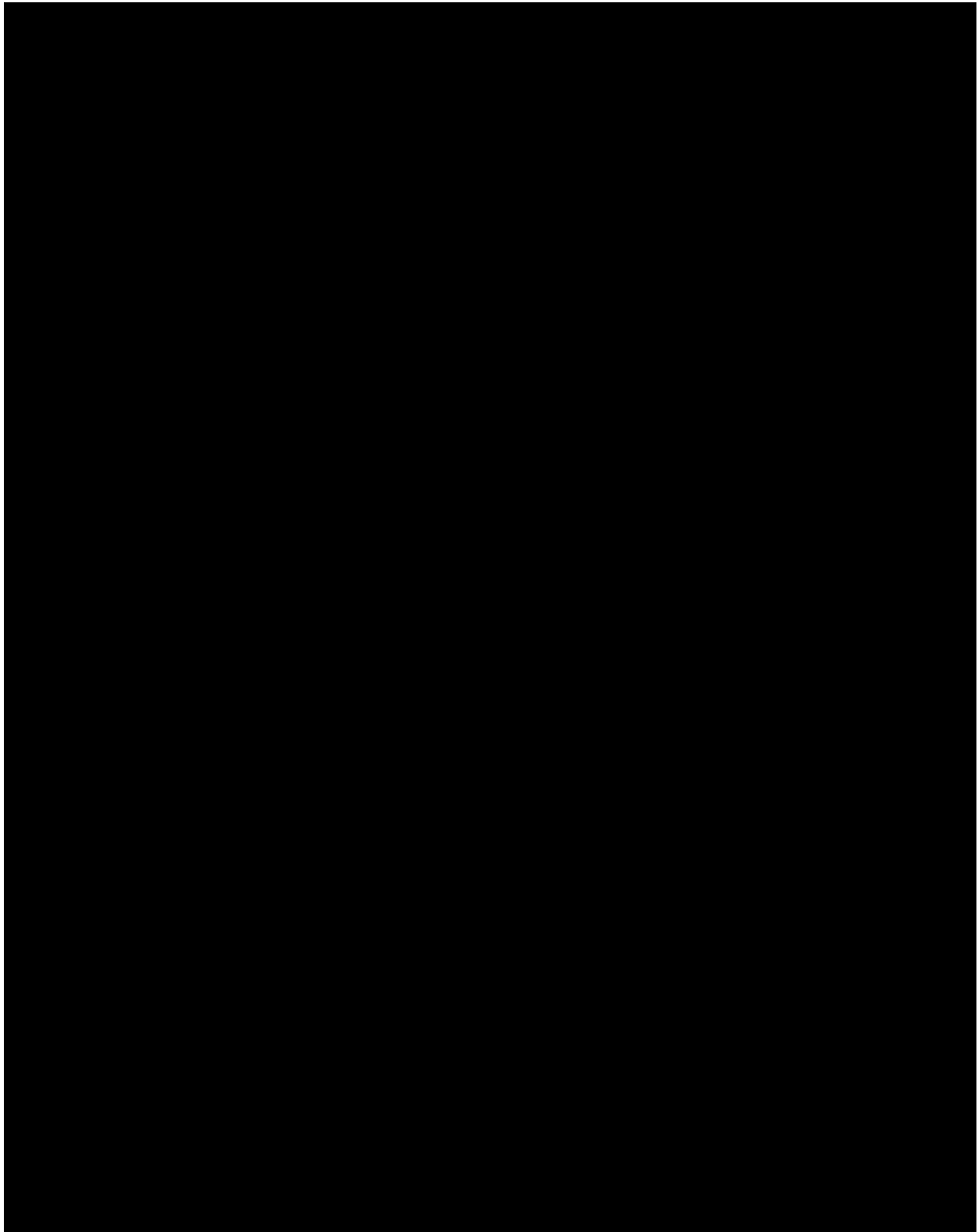


Figure 67: Token Cache Provider Class and Package Diagram

6.2.11.3. Trusted Client Manager



Figure 68: Trusted Client Manager Plugin Configuration File

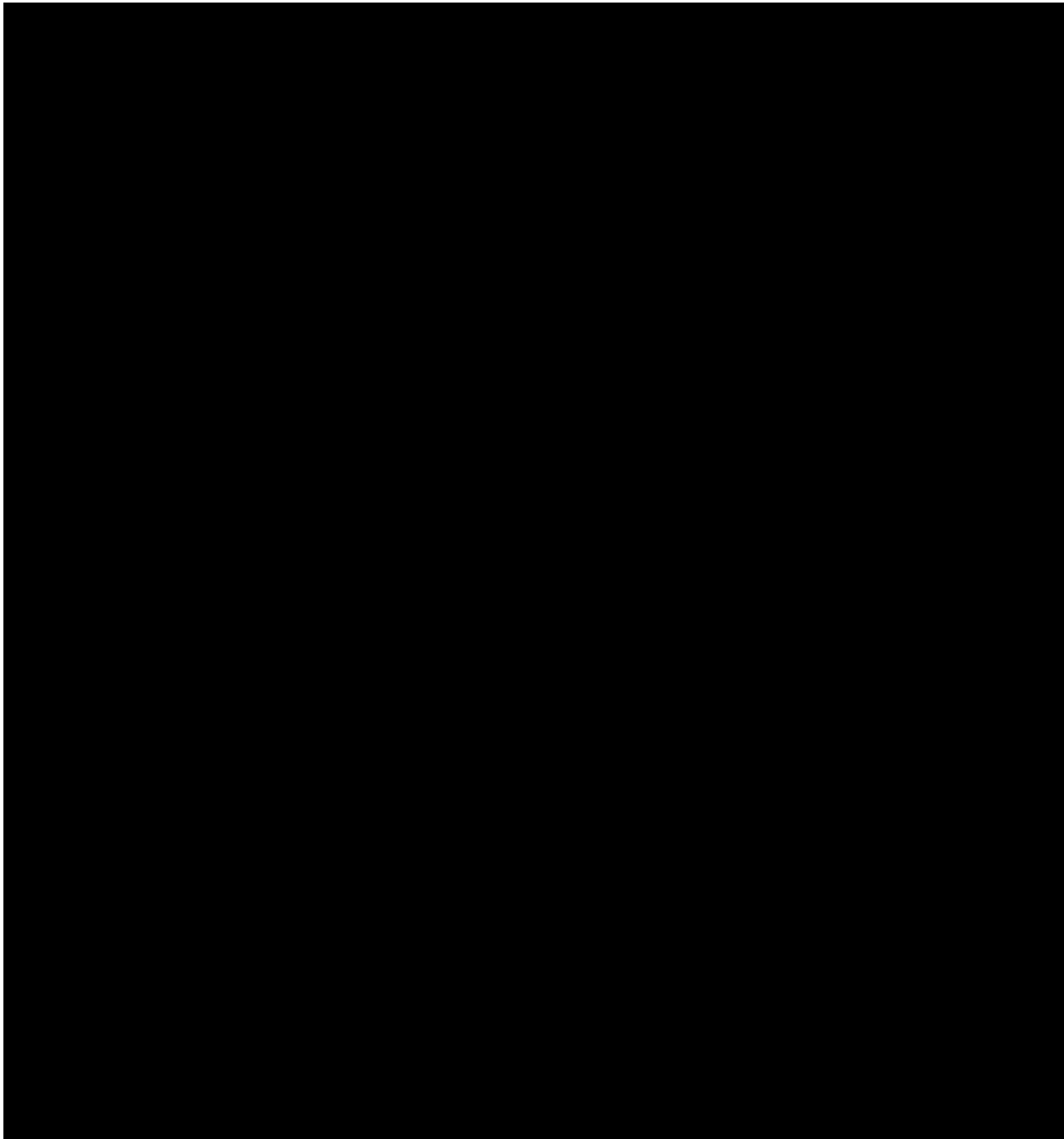


Figure 69: Trusted Client Manager Class and Package Diagram

6.2.11.4. OAuth Web Application

This section provides detail design for the custom web application for OAuth. The following is the package diagram for the OAuth web application.

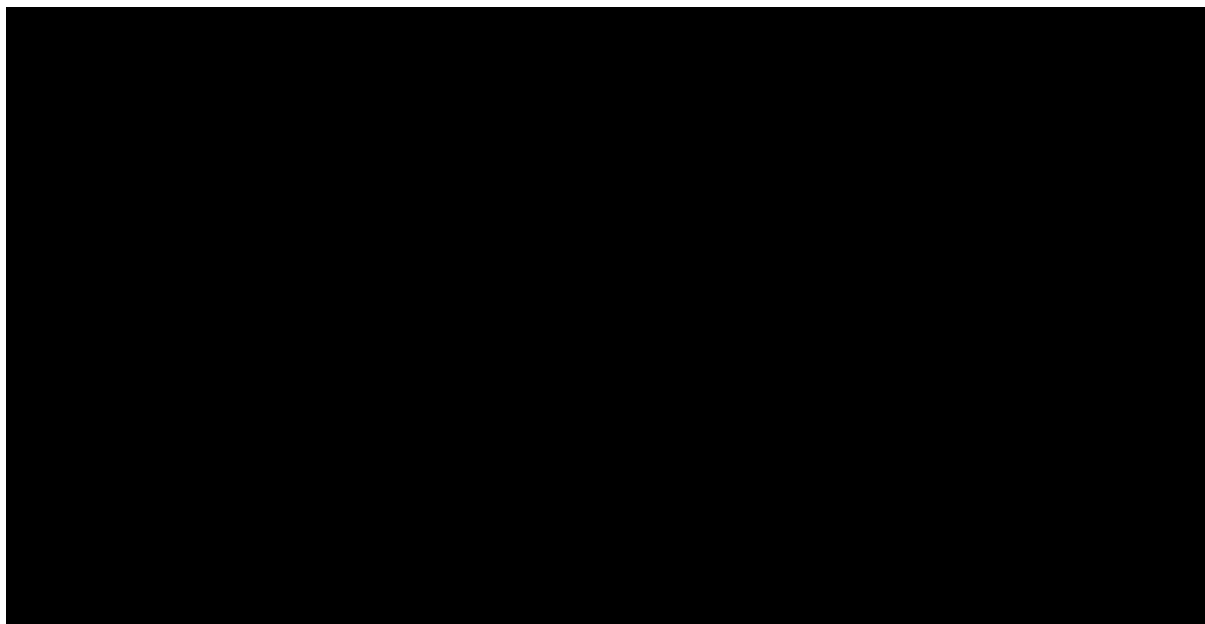


Figure 70: OAuth Web Application Package Diagram

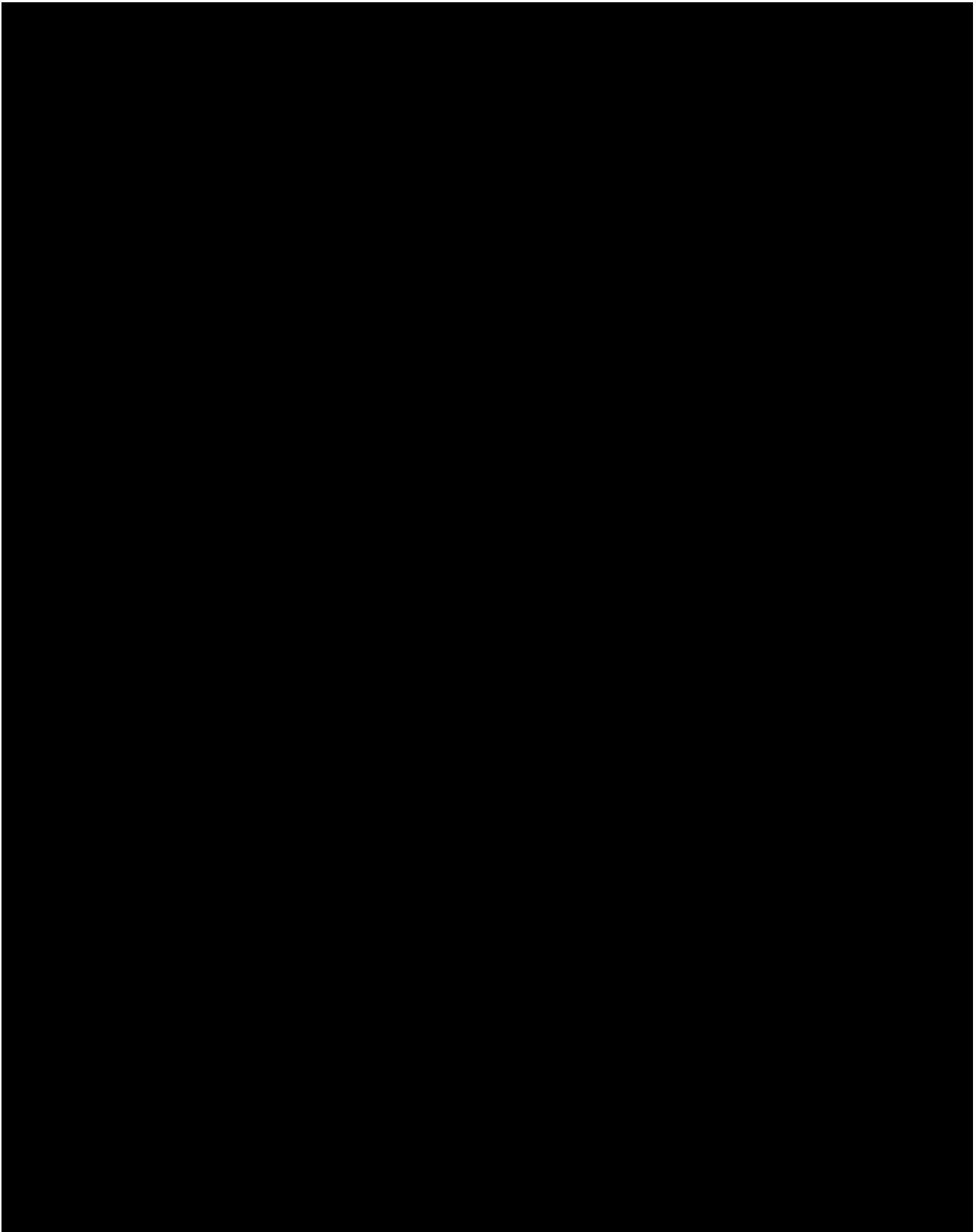


Figure 71: OAuth Web Application Data Access Object (DAO) Class and Package Diagram

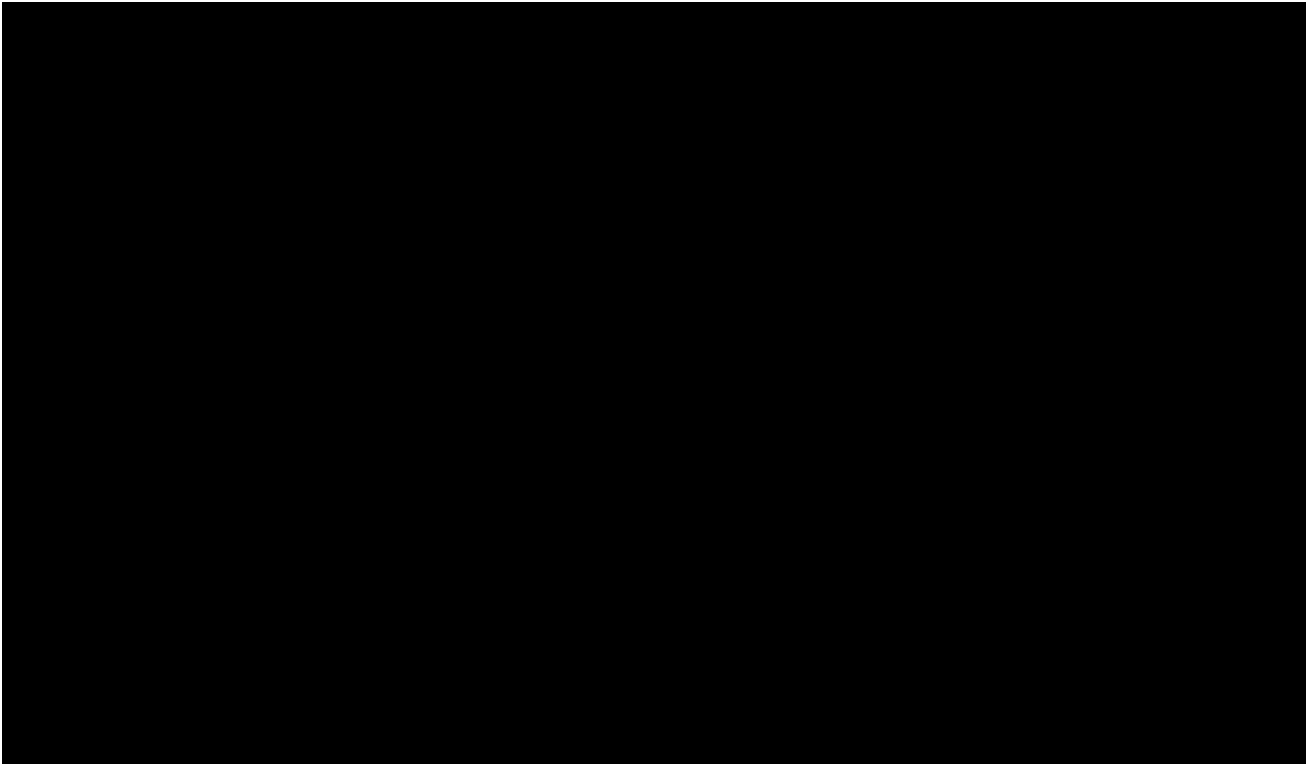


Figure 72: OAuth Web Application Entity Class and Package Diagram



6.2.11.5. User Session Data Storage

This section describes changes to the VAAFI OAuth solution to support storing user session data in association with Access Token issuance. By storing session data, the VAAFI OAuth solution can inject user session data (authentication identifiers and traits) into the OAuth token validation process. One primary benefit of this approach is to provide systems and applications serving OAuth content to serve content based on the user session context. The following are the high-level steps in the end-to-end OAuth flow:

1. Client requests consent to access user's resource.
2. User authenticates and provides consent to the client's request.
3. An Authorization Grant is issued. The authorization grant process constructs and stores the user's authentication identifiers and traits, which link to the corresponding instance of the Authorization Grant entity.

4. Client requests access to the user's resource to which it obtained consent from the OAuth PEP. Client includes the Access Token that was issued.
5. OAuth PEP validates the token. For a valid token, the PEP retrieves the user session data associated with the token and includes it in the response.
6. Client's request is authorized and the user session data provided to the system serving the OAuth resource.
7. Construction and persistence of session data occurs during the Authorization Grant phase. Session data augmentation occurs during the token validation Phase.

6.2.11.5.1. Authorization Grant Process

This section illustrates the changes required to construct and retain the user session data during the Authorization Grant Phase, Authorization Code Grant and Implicit Grant. This process is similar to other two authorization grant types.

Authorization Code Grant

Authorization Code Grant type (also known as Three-Legged Flow) is used to obtain both authorization codes and access tokens.

A client receives an authorization code after the user consents to the access request. The client subsequently requests an access token by submitting the authorization code. The following illustrates the issuance of the authorization code and the access token.

Authorization Code Issuance Flow

The following figure illustrates the OAuth flow for Authorization Grant.

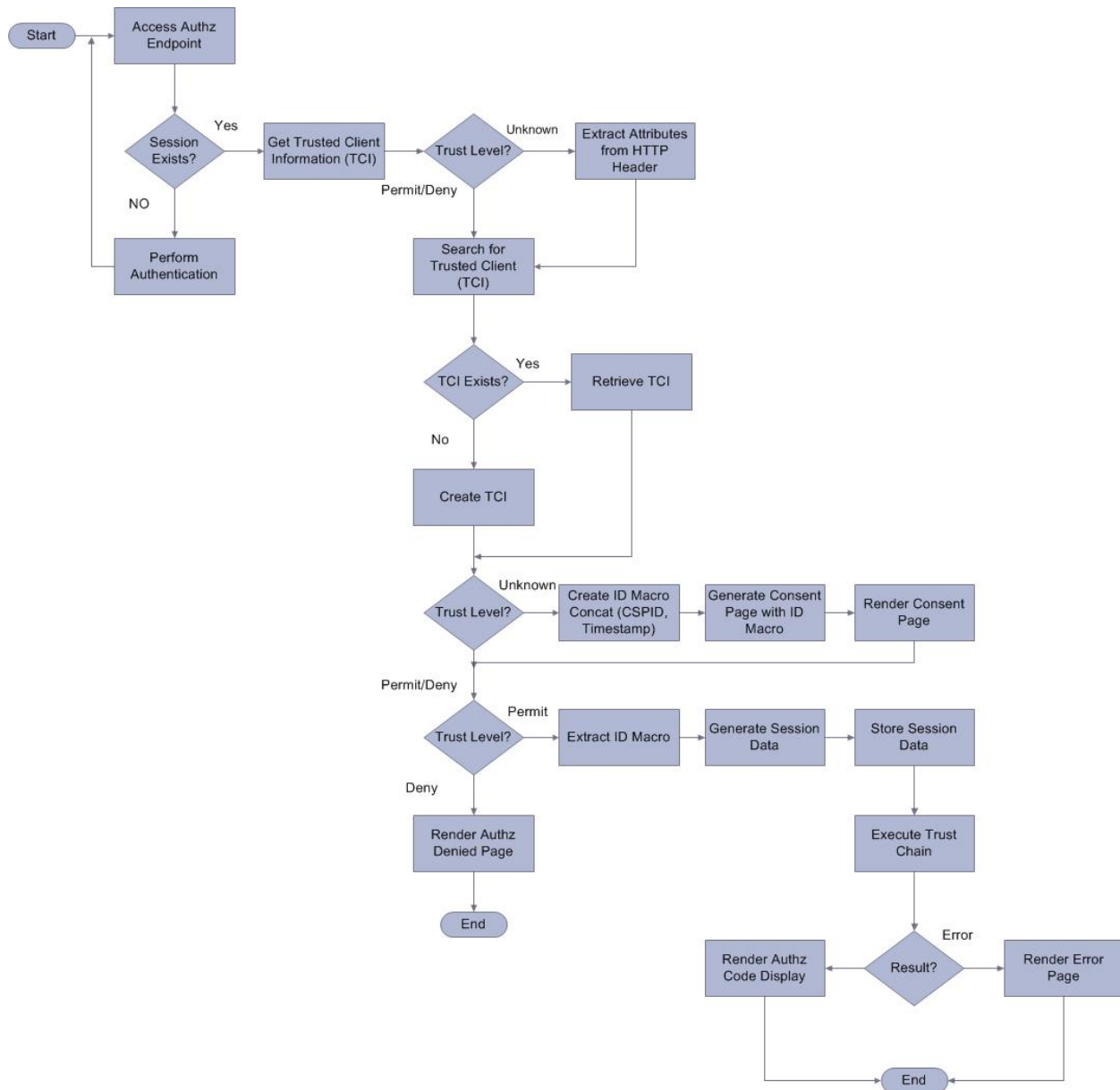


Figure 74: Authorization Code Issuance Flow

1. Client application accesses the Authorization (Authz) Endpoint using a browser.
2. An authenticated session check is performed for this request.
 - a. If an authenticated session does not exist, a redirection occurs. The redirection takes the user to the AccessVA CSP selection page to perform authentication.
 - b. If the session exists, then the flow continues with Step 3.
3. Authz Endpoint invokes the TCM.
4. If the Trust Level is UNKNOWN:
 - a. Attributes are extracted from the HTTP Header: va_eauth_csid (CSP Name), va_eauth_uid (user id used to authenticate at the CSP).

- b. The flow continues with Step 6.
- c. If the Trust Level is Permit or Deny, the flow continues with Step 6.
- d. TCM searches the Trusted Client Data Store for an existing Trusted Client Information (TCI).

NOTE: A CSP uniquely identifies the TCI within the TCM. The CSP ID is defined as a concatenation of the CSP name (va_eauth_csid) and the user ID (va_eauth_uid) separated by an underscore; for example, "DSLogon_John.Doe."

- a. If the TCI exists, the TCI is constructed with the existing attributes.
 - b. If TCI does not exist then a new TCI is instantiated.
5. If the Trust Level is UNKNOWN:
- a. TCM creates a custom macro message that is a concatenation of CSP ID and the Timestamp (the ID Macro).
 - b. Authz Endpoint generates the consent page and embeds the ID Macro.
 - c. Authz Endpoint renders the consent page to the user. The flow continues with Step 9 when the user submits the consent page.

NOTE: The consent page consists of an HTML form and includes the custom macro message as a hidden input field.

6. If the Trust Level is Permit or Deny, the flow continues from Step 9.
7. User has an option to Permit or Deny the authorization request.
- a. If the user Denies the request, Authz Endpoint renders an Authorization Denied page and the process ends.
 - b. If the user Permits the request, the flow continues with Step 10.
8. Authz Endpoint invokes the TCM, which extracts the ID macro from the hidden form field.
9. The TCM extracts user session data from the HTTP Header fields and constructs a succinct representation of this data. TCM stores this data in a data store in conjunction with the ID macro.

NOTE: The ID Macro serves as the key to reference the session data.

10. Authz Endpoint executes the Trust Chain.
11. The Trust Chain returns the execution result.
- a. If the result is an error, the flow ends.
 - b. If the result is an Authorization Code, the client receives and the flow terminates.

Access Token Issuance Flow

The following figure illustrates the Access Token Issuance in the Authorization Code Grant type.

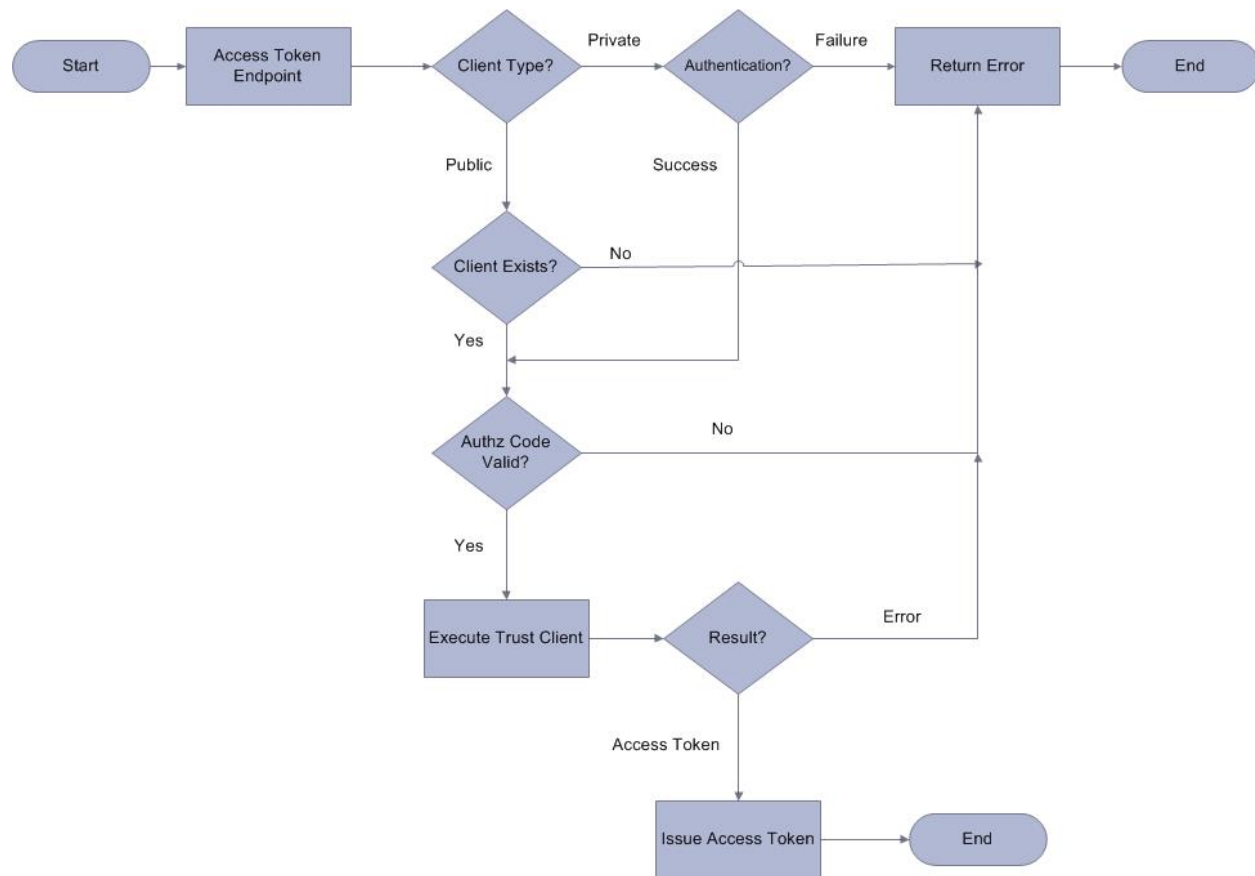


Figure 75: Access Token Issuance Flow

NOTE: The client_id and client_secret parameters in the Token Endpoint request perform client authentication for confidential clients.

1. Client accesses the Token Endpoint.
2. If the client type is Private, the Token Endpoint authenticates the client.
 - a. If the authentication fails, an error generates and the process ends.
 - b. If the authentication succeeds, the flow proceeds to Step 3.
3. If the client type is Public, the Token Endpoint checks if the client is registered.
 - a. If the client does not exist, an error generates and the process ends.
 - b. If the Client registration exists, the flow proceeds to Step 3
4. Token Endpoint validates the Authorization Code provided by the client.
 - a. If the Authorization Code is valid, the Token Endpoint executes the Trust Chain

NOTE: This Trust Chain is the same as the one executed in the Authorization Code Issuance Flow. However, this Trust Chain follows a different execution path.

- b. If the Authorization Code is not valid, an error returns and the process ends
- 5. The Trust Chain returns the execution result.
 - a. If the result is an error, an error generates and the flow ends.
 - b. If the result is an Access Token, the token is returned to the client and the flow terminates.

6.2.11.5.2. Implicit Grant Flow

The implicit grant type is used to obtain access tokens immediately after the access consent. The client receives an access token as the result of the authorization request. The following figure describes the Implicit Grant flow.

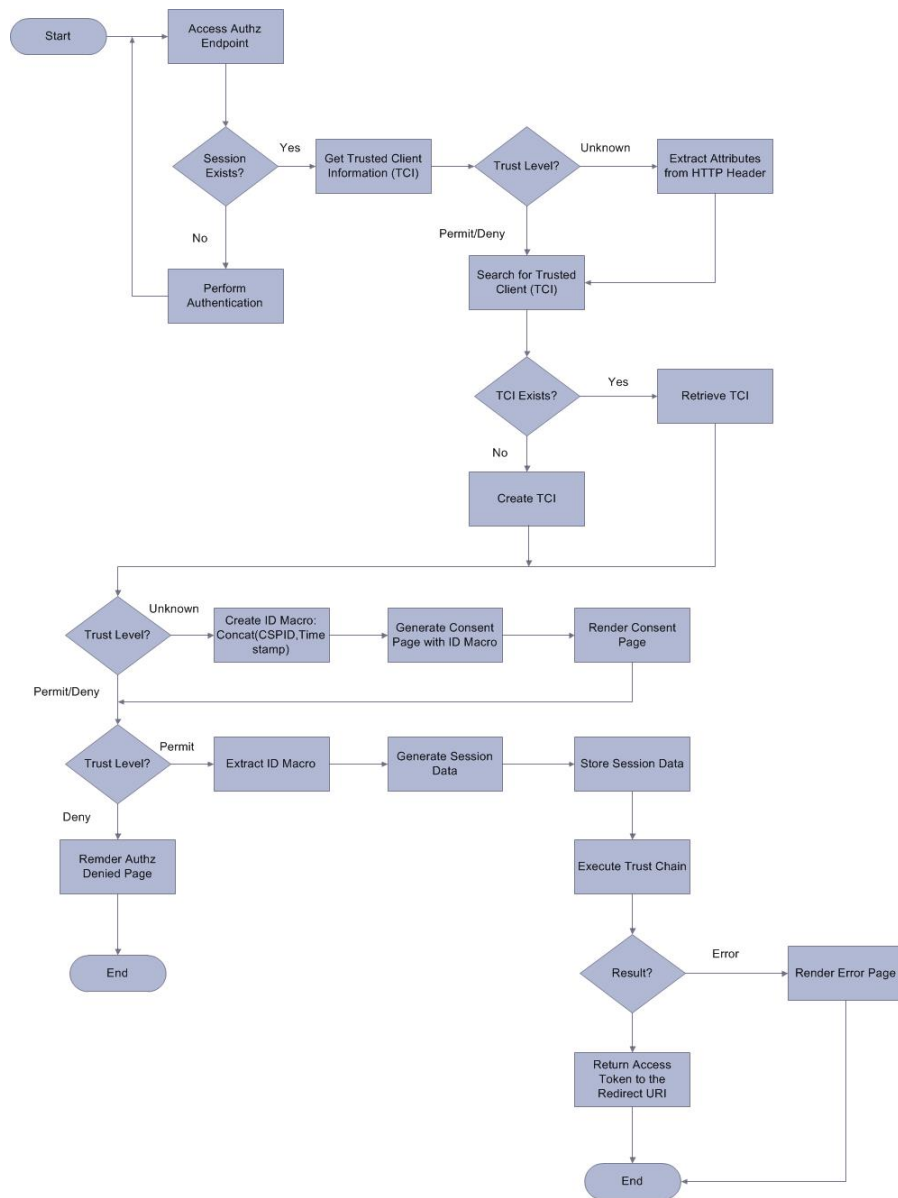


Figure 76: Implicit Grant Flow

1. Client application accesses the Authorization (Authz) Endpoint using a browser.
2. This request triggers an authenticated session check.
 - a. If an authenticated session does not exist, a redirection occurs. The redirection takes the user to the AccessVA CSP selection page to perform authentication.
 - b. If the session exists, the flow continues to Step 3.
3. Authz Endpoint invokes the TCM.
 - a. If the Trust Level is UNKNOWN, attributes are extracted from the HTTP header: va_eauth_csid (CSP Name), va_eauth_uid (user id used to authenticate at the CSP), and the flow continues to Step 4.
 - b. If the 'Trust Level' is Permit or Deny, the flow continues from Step 4.

4. TCM searches the Trusted Client Data Store for an existing TCI.

NOTE: A CSP ID uniquely identifies The TCI within the TCM.

5. If the TCI exists, the TCI is constructed with the existing attributes.
6. If TCI does not exist, a new TCI is instantiated.
7. If the Trust Level is UNKNOWN:
 - a. TCM creates a Custom Macro Message that is a concatenation of CSP ID and the Timestamp (the ID Macro).
 - b. Authz Endpoint generates the consent page and embeds the ID Macro.
 - c. Authz Endpoint renders the consent page to the user. The flow proceeds to Step 9 when the user submits the consent page.

NOTE: The consent page consists of a HTML Form. The Custom Macro Message is included as a hidden input field.

8. If the Trust Level is Permit or Deny, then the flow continues to Step 9.
9. User has an option to Permit or Deny the authorization request.
 - a. If the user Denies the request, the Authz Endpoint renders an Authorization Denied page and the process ends.
 - b. If the user Permits the request, the flow continues to Step 10
10. Authz Endpoint invokes the TCM,) which extracts the ID Macro from the hidden form field.
11. TCM extracts user session data from the HTTP header fields and constructs a succinct representation of this data. TCM stores this data in a data store in conjunction with the ID Macro.

NOTE: The ID Macro serves as the key to reference the session data.

12. Authz Endpoint executes the Trust Chain. Appendix A provides a detailed description of the Trust Chain.
13. The Trust Chain returns the execution result.
 - a. If the result is an error, an error generates and the flow ends.
 - b. If the result is an Authorization Code, the code is returned to the client and the flow terminates.

6.2.11.5.3. OAuth PEP Flow

After a client obtains Access Tokens, it follows up with a request to access the resource. The OAuth PEP is responsible for validating the Access Token and permits or denies based on the validation result. One of the major customizations of the OAuth implementation is to provide the user session data that was associated with the Access Token during the Authorization Grant phase. The following describes OAuth PEP flow.

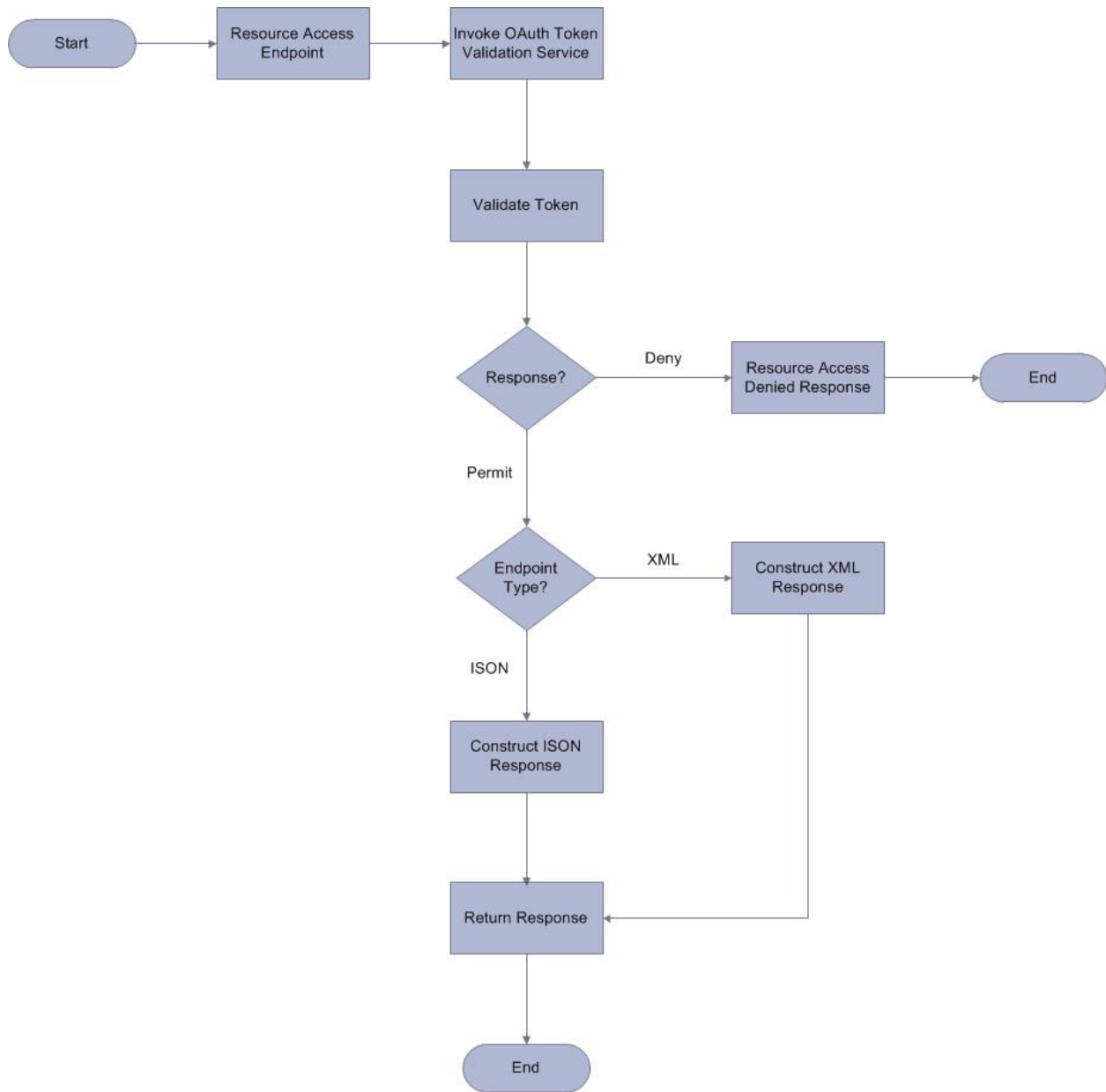


Figure 77: OAuth PEP Flow

1. Client requests access to an OAuth resource from the OAuth PEP using previously issued Access Token.
2. The OAuth PEP constructs a WS-Trust request and invokes the OAuth Token Validation Service endpoint. The request includes the Access Token and various context parameters as per the PEP configuration.
3. The OAuth Token Validation Service invokes and executes the token validation process
 - a. If the Access Token validation fails, the response contains an Authorization Denied message and the flow ends.
 - b. If the Token validation succeeds, STS service returns user specific session data.

4. OAuth PEP constructs the response (user session data) based on the endpoint type:
 - a. If the endpoint is XML, then an XML response is returned.
 - b. If the endpoint is JSON, then a JSON response is returned.
5. The response is returned to the PEP.

6.2.12. IAM STS WS Detailed Design

The IBM XI52 DataPower Appliance (XI52) functionality will be used as an enterprise service bus for IAM STS WS. This will provide transformations from inbound requests, central processing, and lookups to STS data sources. Using WS-* security mechanisms and transports, the authorizations and claims appropriate for the client will be extended into the back-end processing flows, ensuring protected data is only provided for an active claim.

6.2.12.1. IAM STS WS Multi-Protocol Gateway (MPGW)

The IAM STS WS MPGW (MPGW) is an XI52 feature. It will be the consumer interface for IAM STS WS. The MPGW will support STS requests over REST and SOAP, and respond with SAML or JSON tokens.

Details TBD until Sprint 5 is complete.

6.2.12.2. IAM STS WS Application Firewall (AFW)

The IAM STS WS AFW (AFW) is an XI52 feature. It will be the processing core of the STS service. It will accept and process STS requests from the MPGW. It will then contact appropriate datasource services, and obtain security tokens or identity data. Finally, it will process and assemble tokens and identity data, and return the appropriate token(s).

Details TBD until Sprint 5 completes.

6.2.12.3. IAM STS WS Proxy (WSP)

The IAM STS WS WSP (WSP) is an XI52 feature. It will be utilized as datasource services for the producers of STS tokens, as well as additional identity stores. Each instance will provide specific STS or identity data to the AFW processing core.

Details TBD until Sprint 5 completes.

6.2.12.4. VAAFI STS Web Service Process Flow

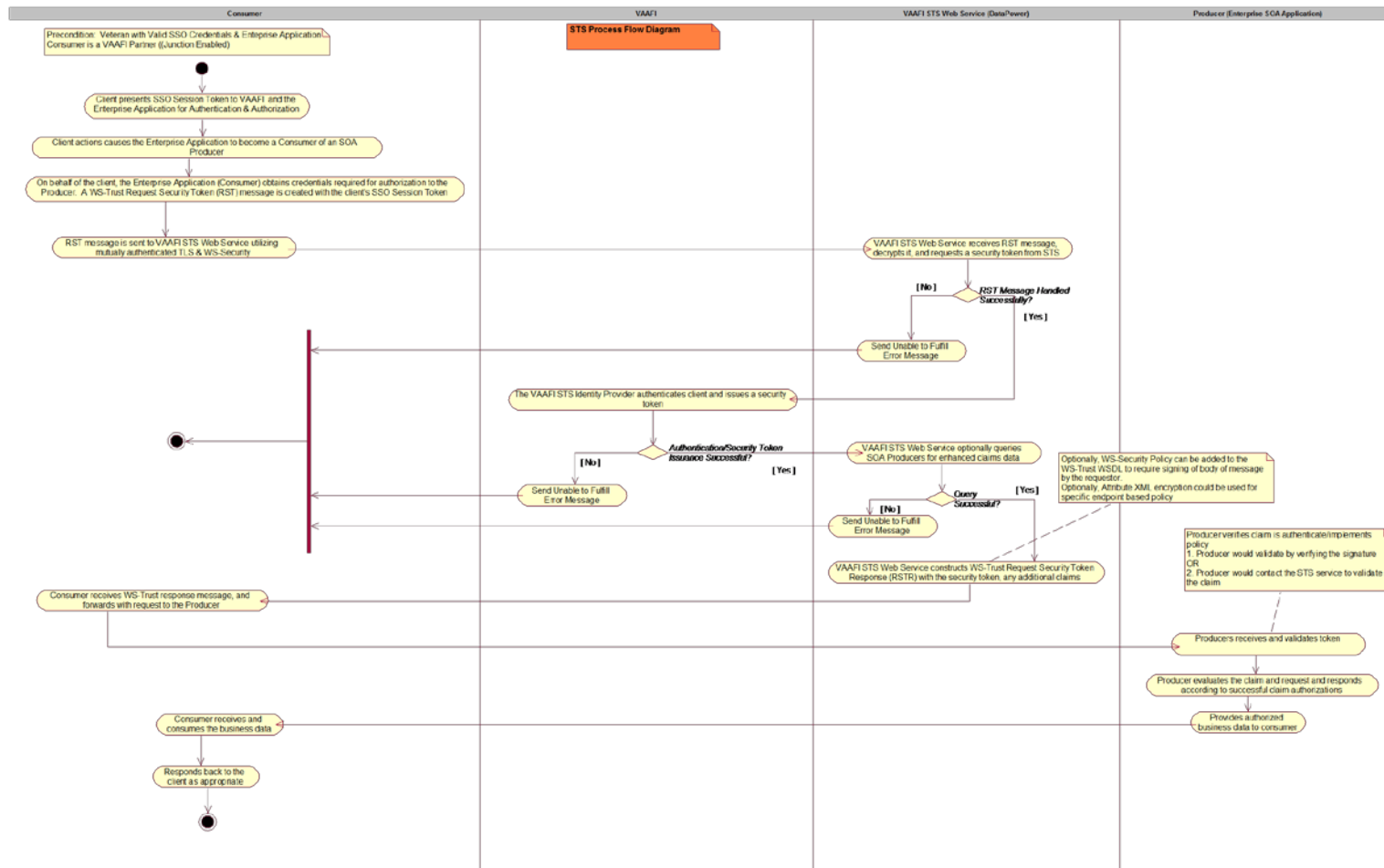


Figure 78: VAAFI STS Web Service

The VAAFI STS Web Service workflow follows:

Assumption: Veteran (client) has established an active security token with VAAFI.

1. A Veteran user (client) selects a VAAFI enabled Enterprise Application (EA).
2. The client presents an SSO session token to VAAFI and the EA for authentication and authorization.
3. Client actions within the application cause the EA to need to consume additional resources within the VA. The EA becomes a consumer of an SOA Producer.
4. On behalf of the client, the consumer obtains required authorization credentials to the Producer. It the client's SSO Session Token to create a WS-Trust Request Security Token (RST) message.
5. The WS-Trust RST message is sent to the VAAFI STS Web Service utilizing SOAP over a mutually authenticated TLS and WS-Security protected connection.
6. VAAFI STS Web Service receives the RST message, decrypts it, and requests a security token from VAAFI STS Identity Provider.
7. VAAFI STS Identity Provider authenticates the client, and issues a new security token or claim back to the STS Web Service.
8. VAAFI STS Web Service optionally queries AcS Provisioning for enhanced claims data (e.g., VDS).
9. VAAFI STS Web Service responds with the requested security token in a WS-Trust RSTR message or an "Unable to Fulfil" error message to the consumer if any of the lookups in steps 6-8 fail for any reason.
10. The consumer receives the WS-Trust RSTR message, and forwards the required claim(s) along with a data request to the Producer.
11. The producer receives the claim and request, and performs token validation based on business rules, which may include validation of issuer, calling application binding, conditional and digital signature, and encryption.
12. Once it validates, the Producer evaluates the claim and request, and responds with business data as appropriate for the successful claim authorization for that client.
13. The Producer provides the authorized business data to the consumer.
14. The consumer receives the business data, and responds back to the client as appropriate.

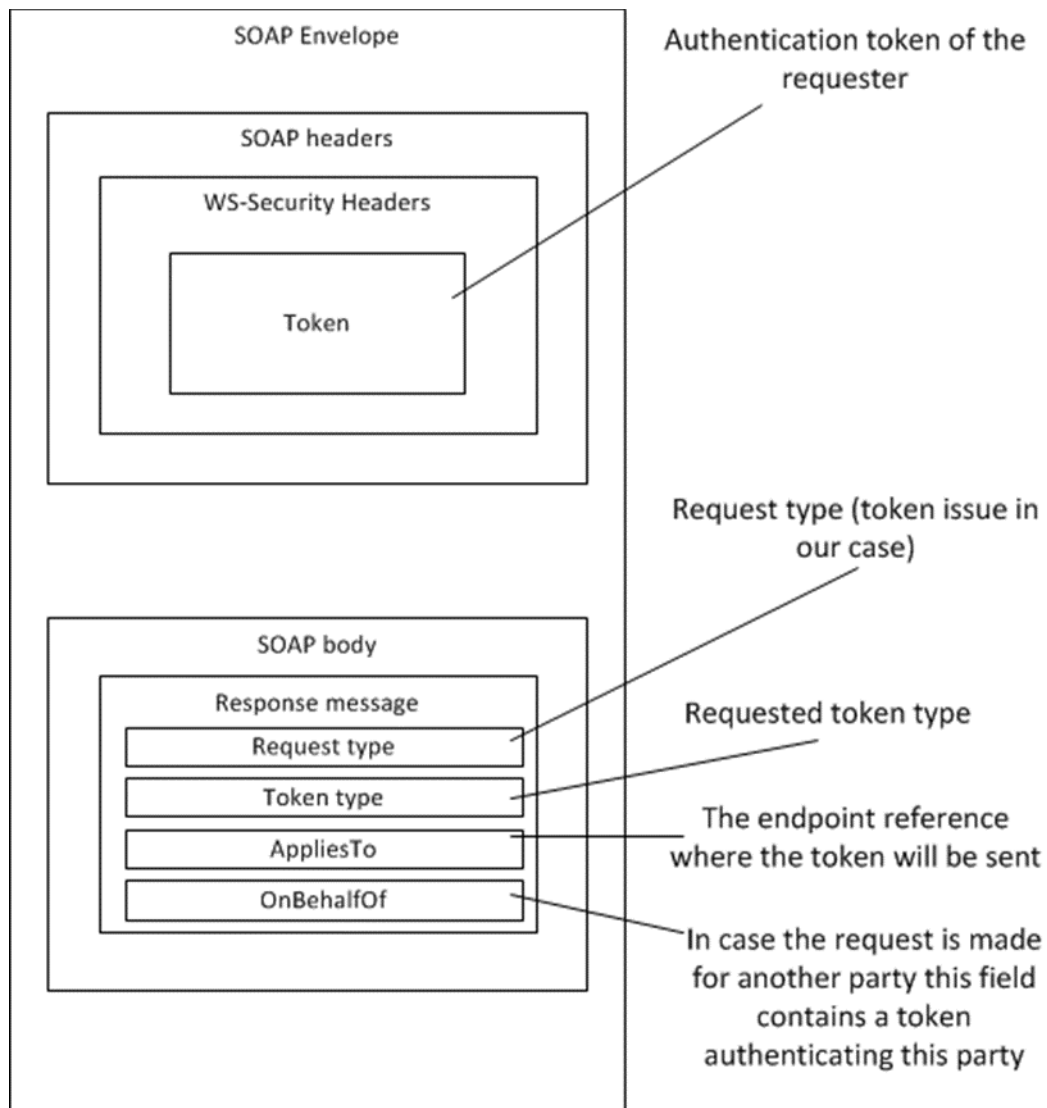


Figure 79: VAAFI Request Security Token (RST) Message

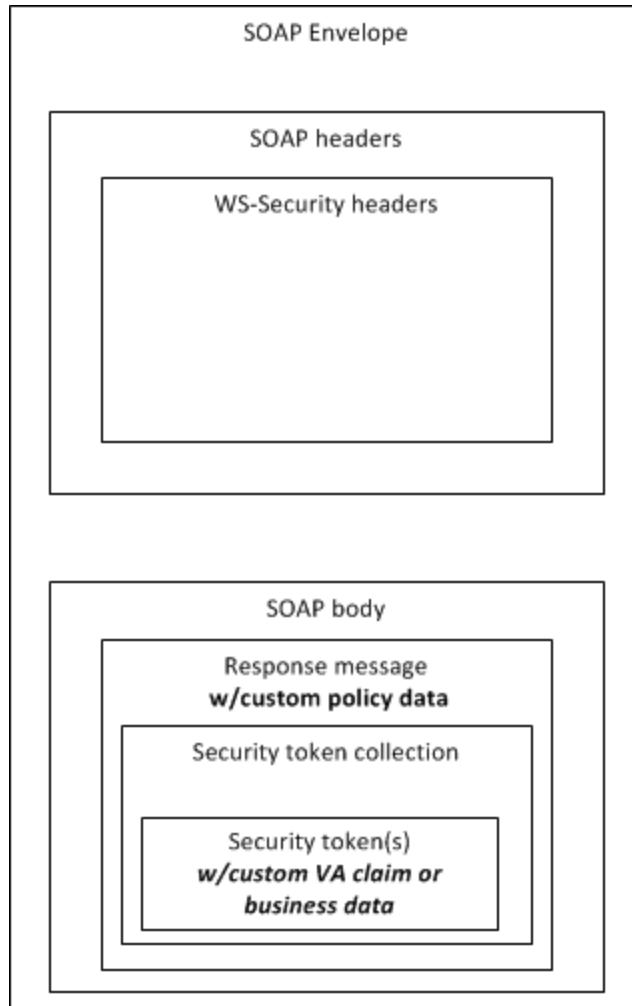


Figure 80: VAAFI Request Security Token Response (RSTR) Message

6.3. Network Detailed Design

Detailed Network Design will be complete by Sprint 4.

6.4. Service Oriented Architecture/ESS Detailed Design

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.1. Service Description for <Consumed Service Name>

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2. Service Design for <Provided Service Name>

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.1. Introduction

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.1.1. Purpose and Scope of Service

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.1.2. Links to Other Documents

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.2. Service Details

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.2.1. Service Identification

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.2.2. Service Versions

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.2.3. Summary of Design and Platform Details

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.2.3.1. SOA Pattern(s) Implemented

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.2.3.2. COTS Platform Vendor Names and Versions for Hosting Platform

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.3. Dependencies

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.4. Service Design Details

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.4.1. Interface Technical Specs

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.4.1.1. Service Invocation Type

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.4.1.2. Service Interface Type

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.4.1.3. Service Name

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.4.1.4. Interface

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.4.1.5. End Points

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.4.1.6. Operations or Methods

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.4.1.7. Message Schemas

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.4.2. Information Model

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.4.2.1. Class Diagram and Description of Entities Involved

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.4.2.2. Mappings from ELDM to Standards Based Schemas

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.4.3. Behavior Model (AKA Use Case Realization)

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.4.3.1. Use Cases (Use Case Model)

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.4.3.2. Interaction Diagrams

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.5. Gap Analysis

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.5.1. Variances from Enterprise Target Architecture

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.5.2. Variances from SLDs

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.5.3. Variances from Standards and Policies

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

6.4.2.5.4. Justification for Exceptions and Mitigation

All of the services will be identified by Sprint 3, and the details of each service by Sprint 6.

7. External Interface Design

VAAFI is a security framework that provides authentication and related services to partner systems. As such, VAAFI is continually adding new external interfaces. Interfacing systems may be CSPs, Web Service Consumers, Web Service Producers, and Web Portals.

When new partners are joining the federation, the VAAFI Team sends them the VAAFI Integration Guide, which describes the various integration patterns. Once the integration is complete, the partner information is added below.

VAAFI Design and the VAAFI SDD capture the AccessVA external interface. AccessVA is integrated into VAAFI via SSO and uses VAAFI SSO to link users between applications.

In addition to partners that use VAAFI services, VAAFI has additional external interfaces, including:

- IBM Integrated Solution Console, which allows administration and configuration of the Tivoli products as described throughout this document;
- WhatsUp Gold, which monitors services, server performance metrics, certificates, and other items; and
- Splunk, which aggregates server and VAAFI logs for ease of searching and reporting.

7.1. Interface Architecture

Each VAAFI interface is customized to work with a specific partner. The VAAFI Security Patterns document describes the process for these integrations.

Figure 16 diagrams the interface architecture.

The VAAFI Interface design forms the basis for the AccessVA Interface design. AccessVA uses VAAFI to provide services and there are no interfaces to AccessVA other than the VAAFI URLs.

7.2. Interface Detailed Design

As a security framework, VAAFI currently has the following types of partner connections:

Table 44: Partner Types

Enablement Type	Description
Web Proxy Consumer	In this scenario, a consuming application has a need to invoke a web service exposed by another application and proxied by VAAFI. In this case, configuration is performed within SSOe/VAAFI to provide the consuming application with access to the web service. The consuming application then invokes the web service through a URL provided by VAAFI (secured by shared SSL certificates), and then VAAFI redirects the request to the producing application.

Enablement Type	Description
Consumption an additional web service between systems where a web proxy producer/consumer relationship has already been established	In this scenario, Application A is already configured as a consumer of one or more web services provided by Application B, and it needs to consume another web service provided by Application B. Configuration of this new web service consumer relationship is less costly and time consuming than the original configuration because much of the overhead for core configuration such as sharing and registration of security certificates has already been completed.
Web Proxy Producer	In this scenario, an application exposes a web service for other applications to use. Configuration of a web proxy producer allows the VAAFI system to secure the web service , so that all access requests must go through VAAFI. This provides centralization of access to all exposed web services, implementation of enterprise-standard security practices, detailed logging/reporting, and alleviation of the burden of implementing detailed security mechanisms by the partner application development team. Exposure of Master Veteran Index (MVI) web services through VAAFI provides an example of this type of enablement.
New Web Service from Existing Web Proxy Producer	In this scenario, an application that was previously configured as a Web Proxy Producer is deploying an additional web service, and this new web service must also be proxied by VAAFI. Configuration of this new web service is less costly and time consuming than the original Web Proxy Producer configuration because much of the overhead for core configuration, such as sharing and registration of security certificates, has already been completed.
Credential Service Provider (CSP)	In this scenario, the SSOe/VAAFI system is configured to recognize external security credentials such as the Defense Manpower Data Center (DMDC) DS Logon credential. This allows users to use the external credential to gain access to designated VA applications. SSOe/VAAFI essentially “binds” the user’s identity browser session with the Credential Service Provider identity and with the same user’s identity in one or more applications. (VAAFI never actually stores an identity; it keeps the attributes for an authenticated browser session in memory, to share appropriately with applications that are accessed through VAAFI.)

Enablement Type	Description
Security Assertion Markup Language (SAML) Reassertion	Under this enablement type, a user who has already authenticated to an enabled CSP (such as DS Logon) is reasserted to a non-VAAFI VA system. Usually these systems are off the VA network and belong to partners such as DoD or Veteran-focused web sites for job searches. In this scenario, VAAFI sends a SAML reassertion to the partner. This SAML reassertion serves as a security token containing the user's authentication information, and confirming to the partner application that the user has already been authenticated and can be granted access.
Standard Junction	In this enablement type, an authenticated user accesses a VA application protected by VAAFI. VAAFI sits between the user and the application, passing the application attributes about the user and the user's session, such as which CSP the user authenticated to; what level of assurance the CSP has; whether the user is who the user claims to be; and personal attributes about the user such as SSN, first name, or date of birth. VAAFI is restricted at this time to providing only attributes that the CSP provides to VAAFI. Some CSPs provide very little information while others provide 40 or more attributes, allowing applications to present a highly customized experience to the user. In the future, VAAFI will query other VA systems (such as PSIM) to provide a greater assortment of attributes to applications despite which CSP the user is authenticated with.
AccessVA Web Site Update	In this enablement type, the AccessVA web site is updated to integrate the new credential service provider or consuming application into the AccessVA logic and functionality.
OAuth	In this enablement type, mobile clients must follow a registration process to consume and use the OAuth services.
STS	In this enablement type, STS WebService consumers must utilize and SML gateway to generate a security token that identifies the appropriate attributes of the trusted VAAFI exchange.

Integration	Description
VDS	VAAFI interfaces with the VDS to collect data from Third Party Onboarding to pass in the headers to the consuming application.
Provisioning	Need Verbiage

The following table identifies the current interfaces both internal and external as well as the enablement type. For more detail related to each interface please reference the complete list of ICDS on the VA SharePoint site.

Table 45: Current VAAFI Interfaces

Interface	Application	Enablement Type
Prudential	VGLI	SAML Reassertion
iAS	Tricare Online	SAML Reassertion
	MHS Learn	SAML Reassertion
EVSS	eBenefits Portal	Standard Junction (/ebenefits-portal)
		Web Service Consumer (PSIM, BEP, Chapter33)
	eBenefits Admin Portal	Standard Junction (/ebenefits-admin-portal)
	SEP	Standard Junction (/sep)
	VDC (aka VONAPP2)	Standard Junction
	VDC	Web Service Consumers
	WSSWEB (VAPii)	Standard Junction (/wssweb)
	ebenefits	Standard Junction
	Resume Bank	SAML Reassertion
	Chat and Co-Browse	SAML Reassertion
	eBenefits Perf and Demo	Standard Junction (/ebenefits-demo and ebenefits-perf)
MHV	My HealtheVet	Standard Junction (/mhv-portal-web)
VONAPP	VONAPP	Standard Junction (/VONAPP)
LGY	Certificate of Eligibility	Standard Junction (/LGY)
	Configuration for MeB	Web Service Consumer
		Web Service Producer
	RPSC Web Services	Web Service Consumer
		Web Service Producer
WAVE	WAVE	Standard Junction (/WAVE)
CRM	TBI Toolbox CRM	Web Service Consumer (PSIM)
BEP	AddressUpdate	Web Service Producer
	BenefitClaimStatus	
	PaymentHistory	
	TrackedItemService	

Interface	Application	Enablement Type
	PaymentInformation	
Chapter33	GetClaimant	Web Service Producer
	GetEnrollment	
Center for Vision Excellence	DVEIVRS	Web Service Producer
		Web Service Consumer
DMDC DS Logon	DS Logon	CSP
DMDC Portlets	Content Manager	Web Service Producer
	Family Health	
	Civilian Employment Information	
	Servicemember Group Life Insurance (SGLI)	
	Other Health Insurance (OHI)	
	Transfer of Education Benefits	
	DVIR	Web Service Producer (DVIR)
		Web Service Producer (PSIM)
	SOES	Web Service Producer (SOE)
	Patient Registration Service	Web Service Producer (PatientReg)
DOD	JANUS	Web Service Consumer (PSIM)
	Patient Discovery	Web Service Consumer
		Web Service Producer
IAM	PSIM	Web Service Producer (VA_IDM_PSIM_HL7v3)
		Web Service Producer (VA_IDM_PSIM)
	AccessVA	Standard Junction (/accessva)
		Standard Junction (/eag)
	VA CSP	CSP
		Web Service Consumer (PSIM)
	PKI CSP	CSP
	IAM Header Page	Standard Junction (/IAM)
	STS Web Service	Web Service Producer

Interface	Application	Enablement Type
	OAuth	Web Service Producer
	Attribute Token Service	Standard Junction (/STS)
		Web Service Producer
	DPLoopback	Web Service Producer
DALC	Remote Order Entry System (ROES)	Standard Junction (/ROES)
		Standard Junction (/ROESPKI)
		Standard Junction (/ROESDMDC)
ESR	ESR	Web Service Consumer
HAPE	VistA Fee 5010 AET	Web Service Consumer
Right Now	IRIS	SAML Reassertion
NCA	BOSS/AMAS	Web Service Consumer
VBH	Veteran Benefits Handbook	Web Service Consumer
		Standard Junction (/vbh)
VHIC	VHIC	Web Service Producer
		Web Service Consumer (PSIM)
	CPRS GUI	Web Service Consumer
Financial Services Center	FSC	Web Service Consumer
HRC	HRC-OSCR	Web Service Consumer
Mobile Health Platform	Mobile Health Platform (formerly CIH)	Standard Junction (/mobilehealthplatform)
		Web Service Consumer
Symantec	Symantec/Norton	CSP
VEIRS	ACA	Web Service Consumer (PSIM)
	D2D	Web Service Consumer (PSIM)
	BENS	Web Service Consumer (PSIM)
HRA	HRA	Web Service Consumer (PSIM)
SDVI	SDVI	Web Service Producer
	VA Person Service	Web Service Producer
PCMMR	PCMMR	Web Service Consumer (PSIM)
BioPoint	BioPoint (Medical Wristbands)	Web Service Consumer (PSIM)
FCMT	FCMT	Web Service Consumer (PSIM)

Interface	Application	Enablement Type
FBSR	FBSR (formerly BFFS)	Web Service Consumer (PSIM)
VOA	1010EZ	Standard Junction
		Web Service Consumer (PSIM)
CRMe	CRMe	Web Service Consumer
VLER	Health Exchange	Web Service Consumer
ESI	Replace ESI Objects	Web Service Consumer
N. Chicago	N. Chicago	Web Service Consumer
VDC	VDC	Web Service Consumer
Connect.gov	Connect.Gov	CSP
CRS	Converged Registries Solution	Web Service Consumer
BizFlow	BizFlow	Web Service Consumer
DSM	Direct Secure Messaging	Web Service Consumer
eHMP	eHMP	Web Service Consumer

8. Human-Machine Interface

VAAFI has a limited amount of its functionality directly exposed to the end user. The only functionality that is exposed to the end user is the following:

- VAAFI web site: A web site within the VA Internet site that presents VAAFI information to end users and helps answer questions and directs them to VAAFI protected resources (for legacy reasons until all partners use AccessVA).
- AccessVA
- AccessVA Widget
- E-Signature Widget
- OAuth

AccessVA is a web-based application. AccessVA consists of a number of dynamically generated HTML pages to provide the switchboard functionality between various Credential Service Providers and VAAFI protected applications. To consolidate all user-facing functionality to a single place, the PKI CSP web pages for registration and error messages and other VAAFI error messages will eventually move to the AccessVA site. It will also eventually replace the Eauth branch of the VA web site, but until all partners use AccessVA, a user that becomes unauthenticated will get redirected to the Select A CSP pages on that site. To do otherwise would send them to AccessVA with no links to reauthenticate and continue use of a non-AccessVA application.

8.1. Interface Design Rules

8.1.1. Section 508 Compliance

The VAAFI design satisfies the following accessibility requirements from 36 CFR part 1194.22, Web-based Intranet and Internet Information and Applications, <http://www.gpo.gov/fdsys/pkg/CFR-2012-title36-vol3/pdf/CFR-2012-title36-vol3-sec1194-22.pdf>:

- 1194.22 (a) A text equivalent for every nontext element shall be provided. VAAFI provides a
- 1194.22 (d) Documents shall be organized so they are readable without requiring an associated style sheet. The structure and organization of the HTML source presents visual elements in a logical sequence in a browser when the style sheet is not available.
- 1194.22 (o) A method shall be provided that permits users to skip repetitive navigation links. The first element of a page in VAAFI is a link to a “content” anchor just above the Main Page content, with text that reads “skip to page content.” It is invisible to most users but very useful to users with screen readers

Additionally, the VAAFI design incorporates the following guidelines for creating accessible web pages:

- Cascading Style Sheets (CSS) should be used for visual representation. A CSS is used to specify borders, background, and font colors, as well as font styles and sizes.

- Tables should not be used for content layout. The page uses div tags in conjunction with cascading style sheets to position the various visual elements.
- Any of the links can be accessed and activated using only keyboard input. The Tab key accesses any of the links on the page and the Enter key can be used to activate it.
- Redundant navigation links are provided at the bottom of the page.

The content is readable on client machines, with larger than normal font size settings. This allows individuals who are visually impaired to easily view the side menus and the main content area.

8.1.2. Other Design Guidelines

VAAFI adheres to VA Enterprise Web Infrastructure Support Compliance Guidelines at:

8.2. Inputs

Section 8.4 identifies inputs.

8.3. Outputs

Section 8.4 identifies outputs.

8.4. Navigation Hierarchy

Within VAAFI, the interactive subsystem that interfaces with end users is becoming AccessVA. Pages from the PKI CSP will continue to be end user facing, but eventually these pages will move to AccessVA, as well. The human interface of other subsystems is only for system administration and management. Therefore, this section focuses on AccessVA.

AccessVA is a web-based application with URLs that are accessible to the public (such as the unauthenticated landing page) and other URLs that are secure and protected by VAAFI. The Input screens can be grouped into the following categories:

- Unauthenticated pages;
- Authenticated landing pages; and
- Application-specific pages.

The outputs are URLs that VAAFI protects. The outputs are secure redirects and SAML redirects that VAAFI handles.

8.4.1. AccessVA Switchboard Functionality

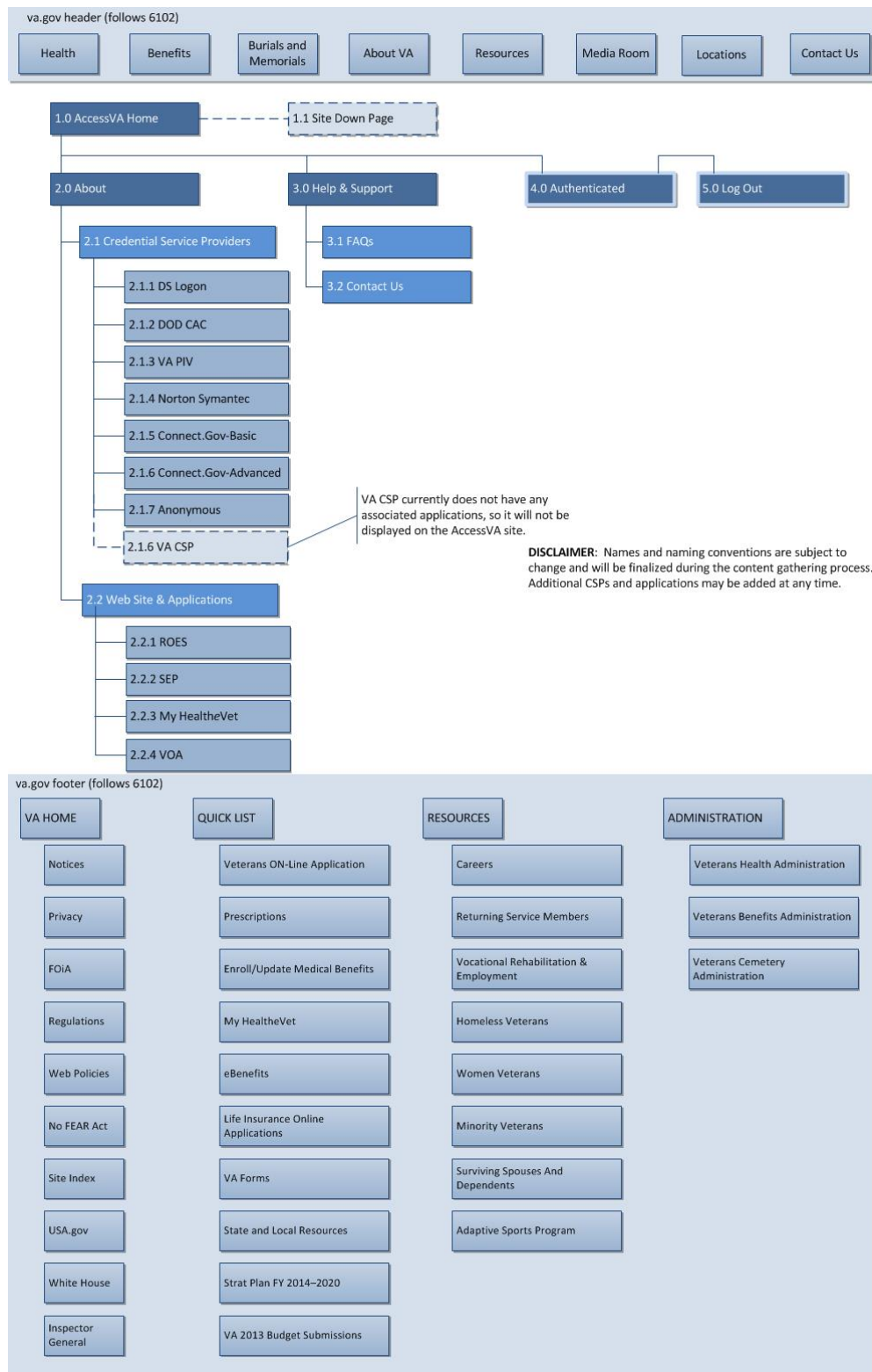


Figure 81: AccessVA Site Map

8.4.1.1. AccessVA Home Page – Unauthenticated

The VAAFI team is redesigning AccessVA to support a new more user-friendly look and feel. The screens that will display will continue to resize based on the screen resolution of the device accessing the page in support of mobile devices. The following figures are screen mockups representing the new AccessVA Unauthenticated Home Page.

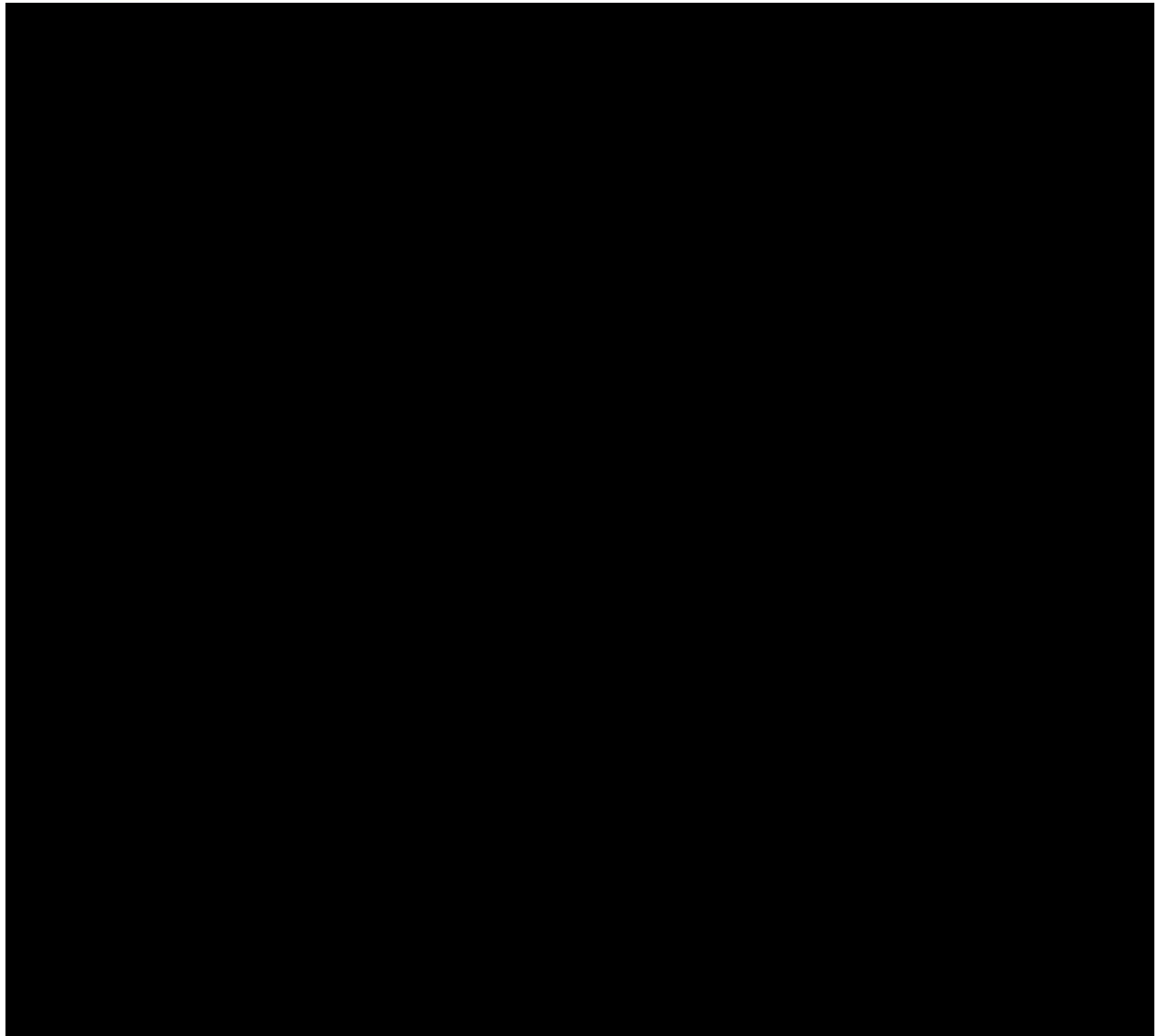


Figure 82: Unauthenticated AccessVA Home Page

8.4.1.2. Unauthenticated – Application Selected: My HealtheVet Example

This page shows the CSPs that a user can select to log into the My HealtheVet application. Each CSP button is a specially crafted link that will take the user to the CSP where they can authenticate, then sends them to My HealtheVet once authentication is complete. The Select Another Application button returns the user to the Unauthenticated AccessVA Home Page. Similar pages present the CSP available for every application that AccessVA supports.

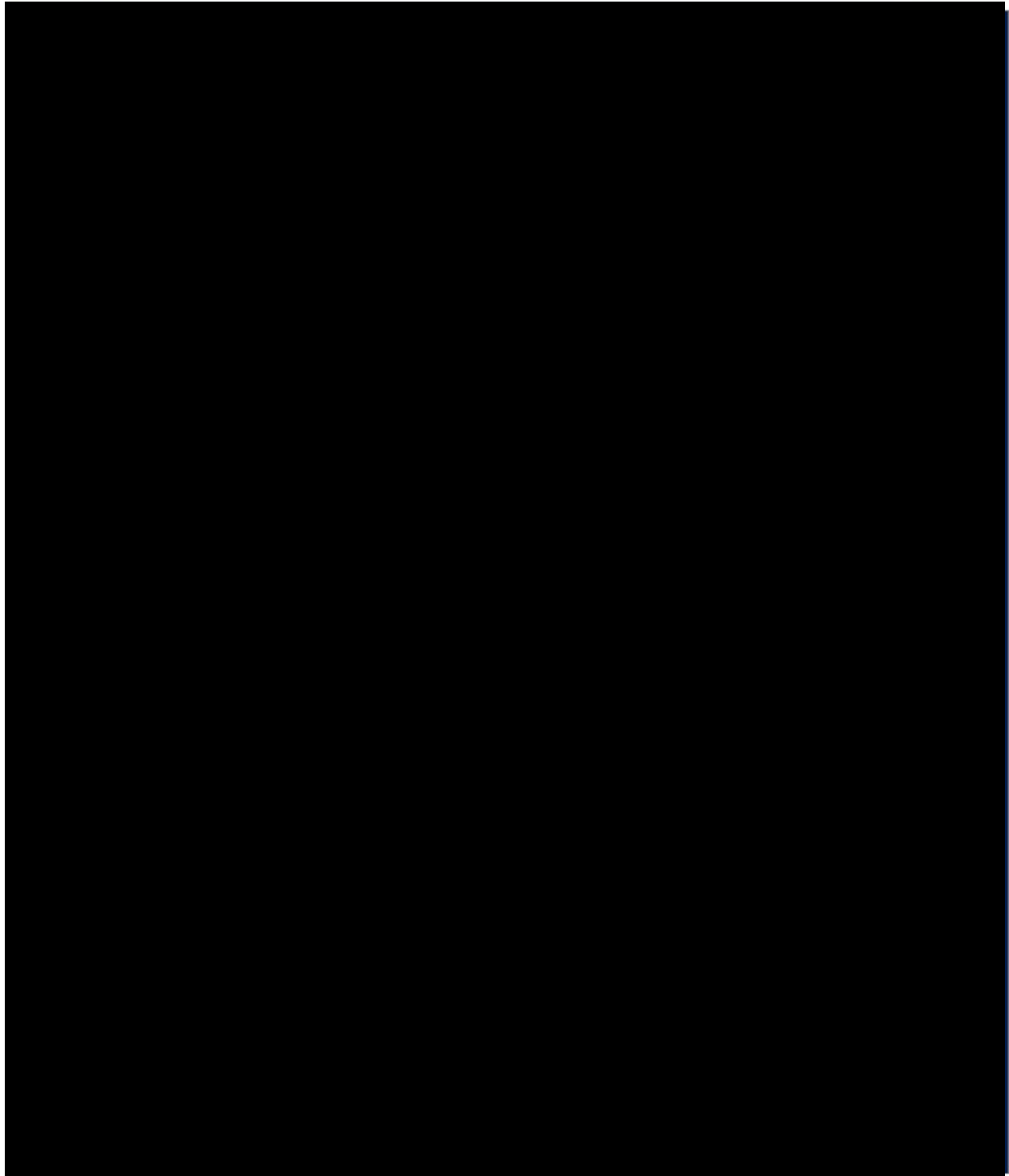


Figure 83: Unauthenticated - Application: MyHealthVet Selected

8.4.1.3. Authenticated: DS Logon Example

This page shows the applications that a user with an Assurance Level 2 DS Logon credential can access. AccessVA has a similar page for each possible CSP – Level of Assurance combination. However, because this increment removed the radio button to display CSPs for authentication, an authenticated user will rarely to be in AccessVA. No link on AccessVA brings a user to this page.

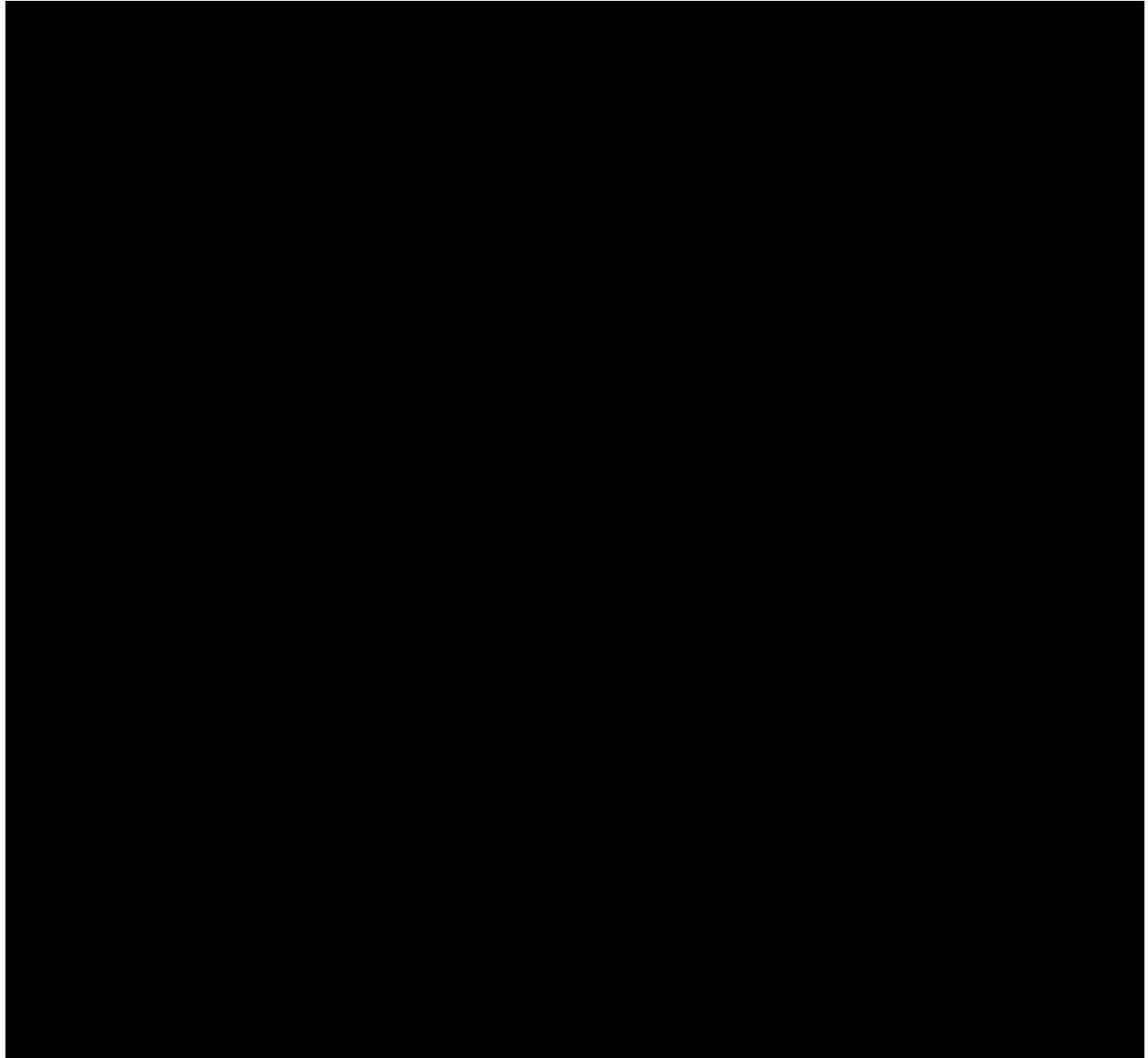


Figure 84: Authenticated Through DS Logon

8.4.2. AccessVA Widget

The widget design allows the application to offer a link to login to AccessVA without navigating to the AccessVA Home page. This is a two-column presentation of the CSP login options for one application.



Figure 85: AccessVA Widget

8.4.3. About Page

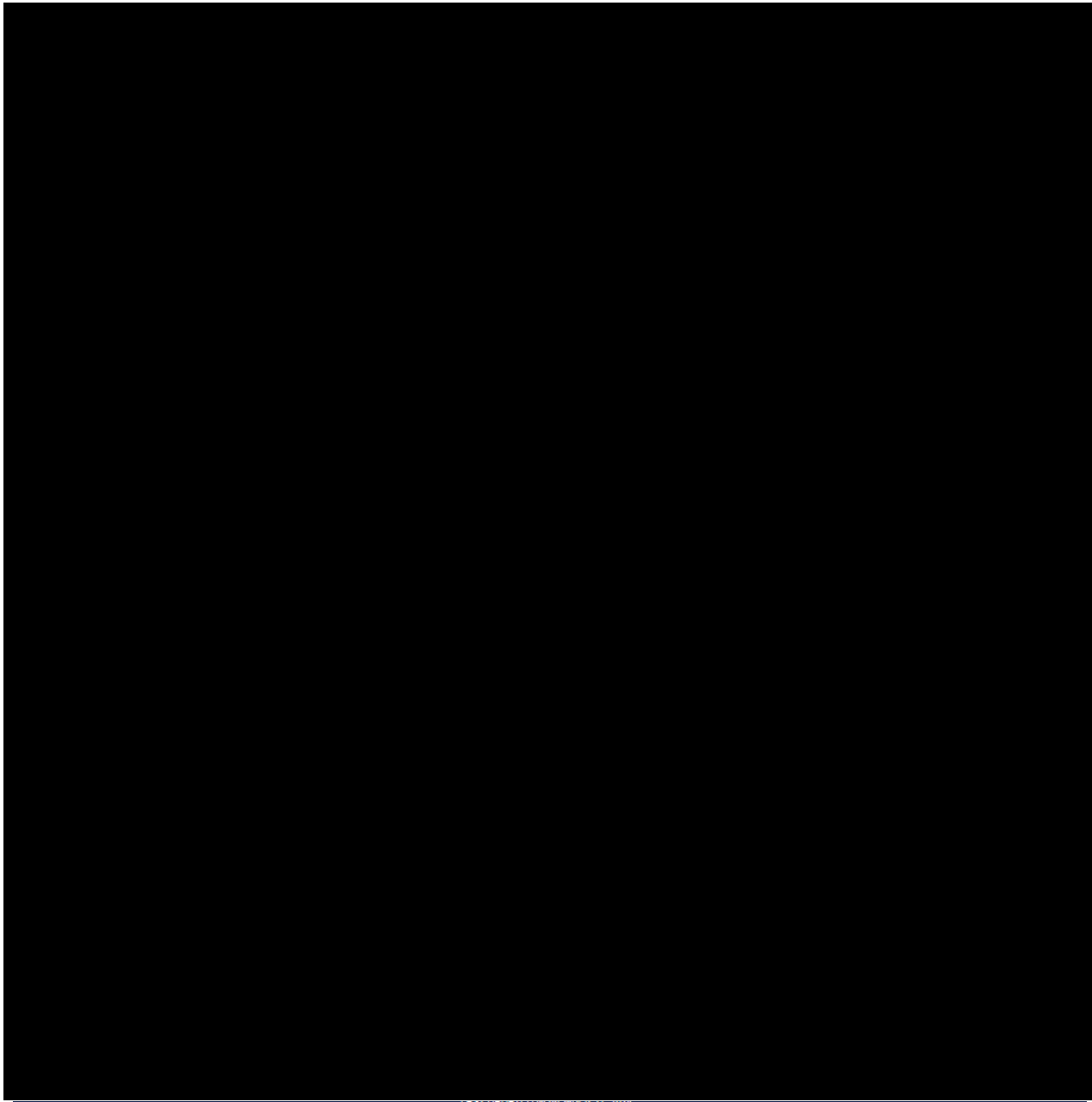


Figure 86: About

The screens in this section have the same panes as Figure 86; however, the text in the center pane differs with each screen. Sections 8.4.3.1 through 8.4.3.2.4 contain the text for each screen.

8.4.3.1. Credential Service Providers (CSPs)

(Refer to Figure 86.)

Center pane text:

As it relates to AccessVA, a Credential Service Provider (CSP) is a trusted source for registration and issuance of logon credentials to access VA resources.

Make a selection from the left menu to learn more about the CSPs available on AccessVA.

8.4.3.1.1. DS Logon

(Refer to Figure 86.)

Center pane text:

The Department of Defense Self-Service Logon (DS Logon) is a secure, self-service logon ID that allows individual to access several websites using a single username and password. DS Logon is available to DoD and VA Service Members and Patients (Active Duty, Guard/Reservists, Retirees), Veterans, Spouses, Eligible Family Members (18 and over), and Civilian Retirees.

If you do not already have a DS Logon you can register for an account online. Questions will be asked so that your record can be located in the Defense Enrollment Eligibility Reporting System (DEERS). You must be enrolled in DEERS to obtain a DS Logon.

For additional information on DS Logon visit the DS Logon Help Center at:
<https://www.dmdc.osd.mil/identitymanagement/help.do?execution=e2s1>

8.4.3.1.2. DOD CAC

(Refer to Figure 86.)

Center pane text:

The Department of Defense (DoD) Common Access Card (CAC) is the standard identification card for Active-Duty Military Personnel, Selected Reserve, DoD Civilian Employees, and Eligible Contractor Personnel. It is also the primary card used to access DoD buildings and it provides access to DoD computer networks and systems.

For additional information and steps to obtain a CAC, please visit the DOD ID Card Reference Center at: <http://www.cac.mil/>

8.4.3.1.3. VA PIV

(Refer to Figure 86.)

Center pane text:

The Personal Identity Verification (PIV) card is issued by a Federal agency and is the standard identification card for all Federal employees and contractors. The PIV card is issued for secure access to Federal and other facilities and access to Federal computer networks and systems.

For additional information about the VA PIV card visit the PIV Card Project page at:



8.4.3.1.4. Norton Symantec LOA 2

(Refer to Figure 86.)

Center pane text:

The Department of Veterans Affairs (VA) now offers you Norton Symantec to access VA websites and applications.

To register for a Norton Symantec account, you will need the following:

- Name and Address
- Working email address
- Date of Birth
- Social Security Number
- Ability to answer Knowledge Based Questions

You will not be charged for a Norton Symantec account. This information is required to verify your identity.

8.4.3.1.5. Norton Symantec LOA 3

(Refer to Figure 86.)

Center pane text:

The Department of Veterans Affairs (VA) now offers you Norton Symantec to access VA websites and applications.

To register for a Norton Symantec account, you will need the following:

- Name and Address
- Credit Card
- Email address
- Phone with ability to receive voice mail or text message

You will not be charged for a Norton Symantec account. This information is required to verify your identity

8.4.3.1.6. Connect.Gov-Basic

(Refer to Figure 86.)

Connect.Gov is a government service that VA has partnered with to make several other log in partners available to you.

Connect.Gov - Basic accounts provide the option to log in with all available Connect.Gov log in partners.

Connect.Gov - Advanced accounts will only display higher security log in accounts which may be required by your VA website.

For additional information about Connect.Gov visit their website at <https://www.connect.gov>.

8.4.3.1.7. Connect.Gov-Advanced

(Refer to Figure 86.)

Connect.Gov is a government service that VA has partnered with to make several other login partners available to you.

Connect.Gov - Basic accounts provide the option to log in with all available Connect.Gov log in partners.

Connect.Gov - Advanced accounts will only display higher security log in accounts which may be required by your VA website.

For additional information about Connect.Gov visit their website at <https://www.connect.gov>.

8.4.3.1.8. Anonymous

(Refer to Figure 86.)

Anonymous access can be used for generic situations which do not require personal identification. To access your website without logging in simply select the button for Anonymous Access.

8.4.3.2. VA Web Sites & Applications

(Refer to Figure 86.)

Center pane text:

AccessVA provides a single starting point and secure log in process to Veterans and other VA customers and business partners to access VA websites and applications.

Select from the left menu to learn more about the VA Websites & Applications available through AccessVA.

8.4.3.2.1. ROES

(Refer to Figure 86.)

Center pane text:

VA's Remote Order Entry System (ROES) application allows Veterans to place orders online for products and/or services available through the Denver Acquisition & Logistics Center (DALC). ROES provides a convenient, secure means of using the Internet to place orders for products available through the DALC. Veterans who currently receive VA care for designated medical/physical conditions can use ROES to request hearing aid batteries and prosthetic socks online. Previous methods of requesting replacement batteries (mail-in battery request card, e-mail request, phone request, etc.) will still remain in place.

If you are a Veteran and want to order products and/or services through the DALC you must have a level two credential. It is strongly recommended that you obtain a DS Logon Premium (level two) credential which will allow you to access not only ROES online ordering capabilities, but also a broad array of VA and Department of Defense (DoD) online resources. If you have any questions or wish to obtain more information regarding this access, please contact the ROES Customer Service Section at dalc.css@va.gov.

A number of other government agency (OGA) clinical professionals also currently place orders for designated products through the DALC. These clinicians, many of whom practice within the DoD healthcare system, are already familiar with the DALC as a convenient source for designated medical products. DoD clinicians can now place orders through ROES, giving them access to an ordering system similar to that enjoyed by VA clinicians. Most notably, DoD

audiologists already ordering hearing aids through VA's government-wide contracts can use ROES to do so, increasing the efficiency and accuracy of hearing aid and accessory orders.

DoD audiologists and associated procurement staff can use their DoD Common Access Card (CAC) to access ROES. For information, contact the DALC Customer Service Section at 303-273-6200 or by email at dalc.css@va.gov.

8.4.3.2.2. SEP

(Refer to Figure 86.)

Center pane text:

The Stakeholder Enterprise Portal (SEP) is a single, secure entry portal that provides VA partner organizations and external stakeholders access to the web-based systems they need to assist Veterans, Reservists, members of the National Guard, and their dependents.

The SEP combines previously autonomous VA partner organizations such as Compensation Service (CS), Vocational Rehabilitation and Employment (VR&E), Health Administration Center (HAC) and Education Service (ES) into a seamless, secure and consistent service.

8.4.3.2.3. My HealtheVet

(Refer to Figure 86.)

Center pane text:

My HealtheVet is the Department of Veterans Affairs (VA) Personal Health Record for Veterans. The website is designed for Veterans and their families to better understand and manage their health. With My HealtheVet, your health information is kept safely and securely online. The website provides tools to partner with your health care team to manage your health care. Use My HealtheVet to record diet, exercise and health history. Send a message to your VA health care team with non-urgent questions, or refill VA medications. Keep track of your VA appointments and review notes from your last clinic visit. New tools and enhancements are added frequently. Features currently include:

- Secure Messaging: Communicate non-urgent concerns or questions to your VA health care team*
- VA Prescription Refills: Refill your VA medication(s) online
- VA Appointments: View detailed information about your VA clinic appointments*
- VA Health Record: View copies of key extracts from your VA electronic health record, including lab and radiology results, Admissions and Discharges, Problem List and VA Notes*
- Military Health History/Military Service Information: Enter and track military health information, and access your Department of Defense (DoD) Military Service Information* (if eligible)
- VA Blue Button: View, print or download your personal health information with the VA Blue Button
- VA CCD: View, print or download a summary of essential health information*
- Veterans Health Library: Medical information specifically targeting Veteran health issues

- My Goals: an online tool to help you set and achieve goals

**Requires a Premium account.*

If you are a Veteran and want a My HealtheVet Premium account, you must register on My HealtheVet and have a Level 2 credential. A DS Logon Premium (Level 2) credential will allow you to access not only all of My HealtheVet features, but also a broad array of VA and Department of Defense (DoD) online resources. The Department of Defense offers more information on how to obtain a DS Logon.

8.4.3.2.4. VOA

(Refer to Figure 121.)

Center pane text:

The Veterans Online Application for Health Benefits (VOA) is a simple to use, secure, online forms gateway that allows Veterans to provide initial and updated enrollment and eligibility information. VOA currently supports the Application for Health Benefits (VA Form 10-10EZ) and the Health Benefits Update Form (VA Form 10-10EZR) and allows the Veteran to choose the appropriate form in order to enroll in VHA health benefits (VA Form 10-10EZ) or update information needed to stay enrolled for VHA health benefits.

VOA's goal is to streamline Veterans' access to VA health care enrollment applications, facilitate their ability to review and manage their personal information, and provide effective communication regarding their enrollment processes. Using VOA to enroll or update information allows the Veteran to submit data directly to the Enrollment System. This reduces the risk of errors in data entry and in the time required to determine eligibility.

Veterans who authenticate through AccessVA will have forms pre-filled with the data the Enrollment System has about them. But whether accessing VOA with or without authenticating, VOA provides Veterans with an enhanced capability to manage their VHA health benefit information (address, military service, insurance), provide supporting information (attachments and documents), and update preferences.

8.4.4. AccessVA Login Button

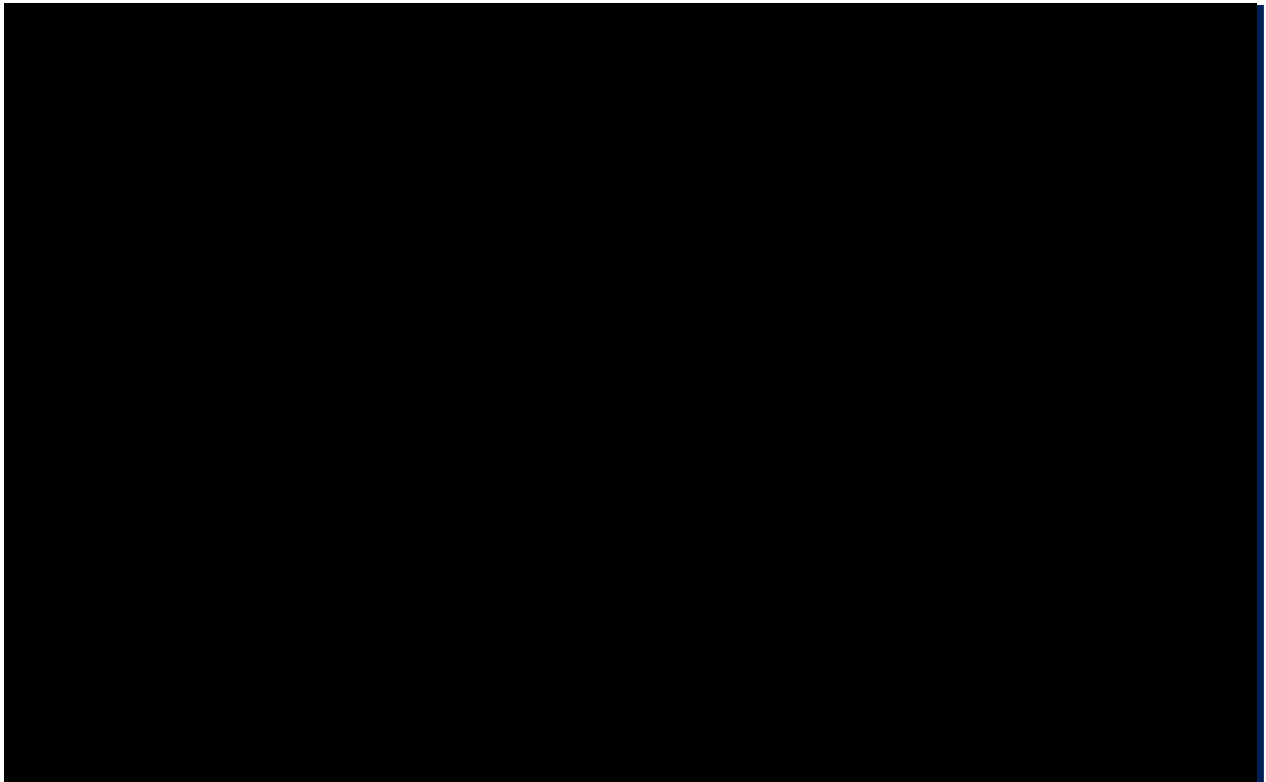


Figure 87: AccessVA Login Button

8.4.4.1. AccessVA User Type Selector Pop-Up Widget

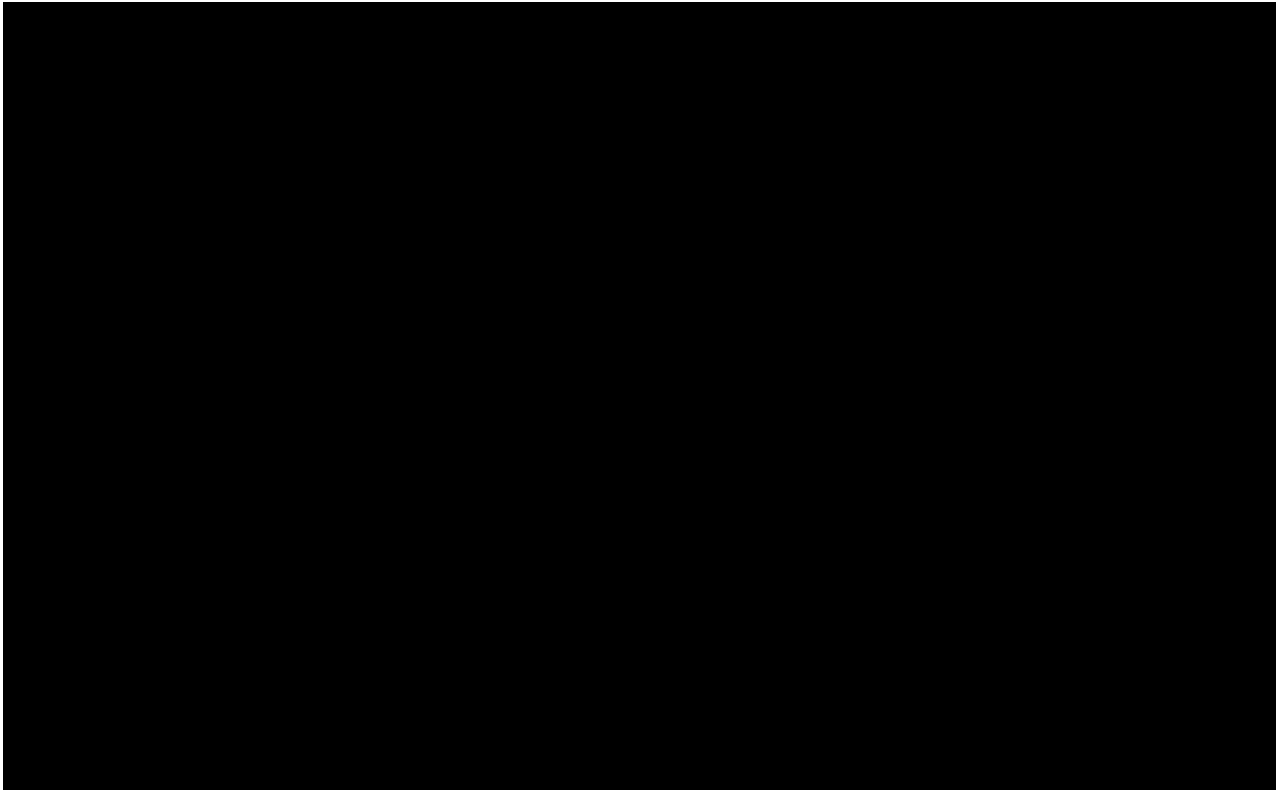


Figure 88: AccessVA User Type Selector Pop-Up Widget

8.4.4.2. AccessVA CSP Selector User Type Pop-Up Widget

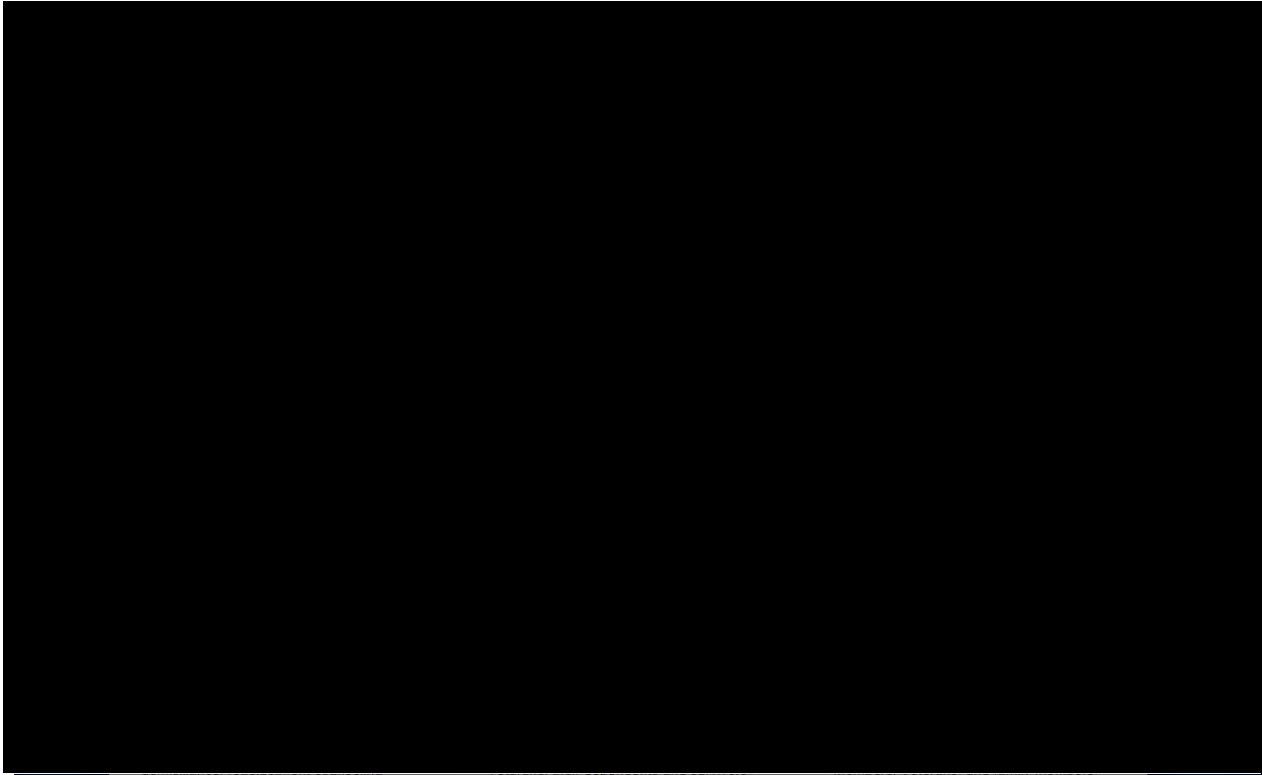


Figure 89: AccessVA User Type Selector Pop-Up Widget – Expanded

8.4.4.3. AccessVA CSP Selector Pop-Up Widget

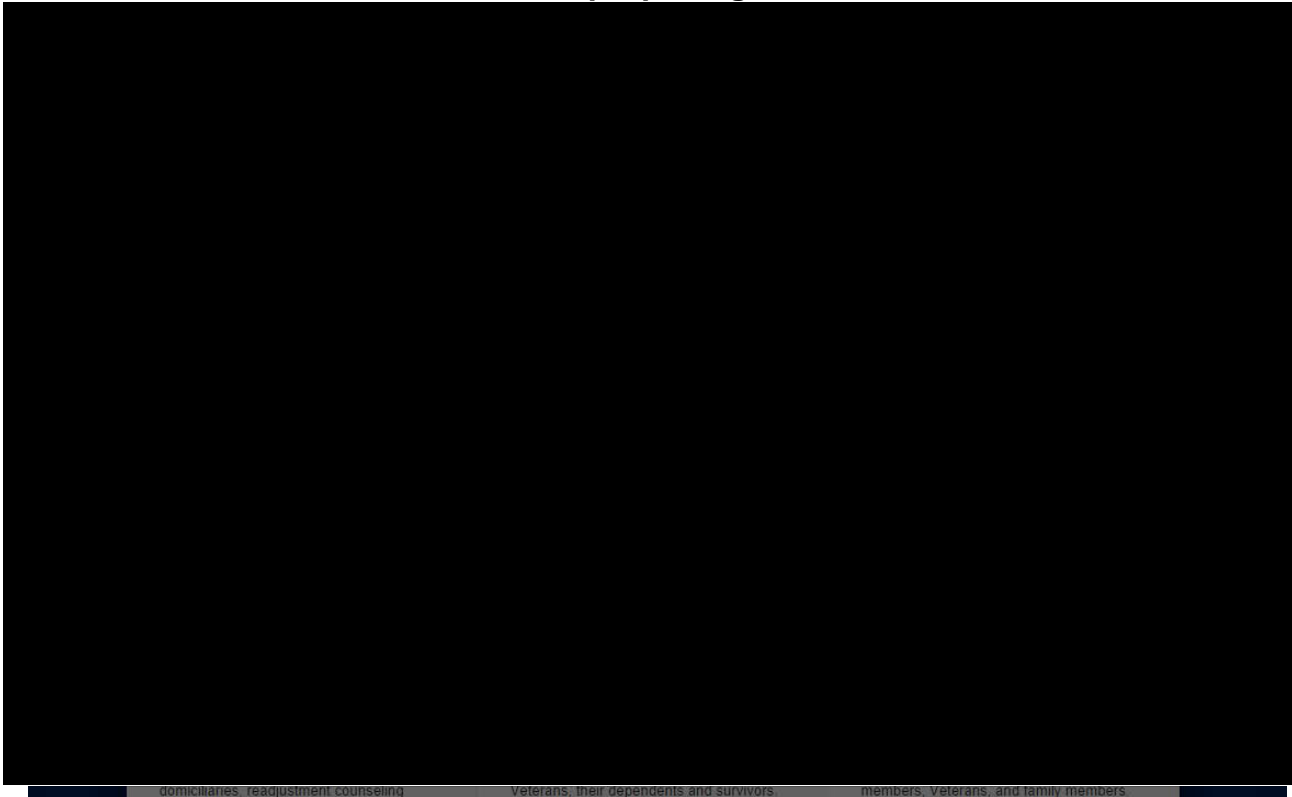


Figure 90: AccessVA CSP Selector Pop-Up Widget

8.4.5. Help & Support

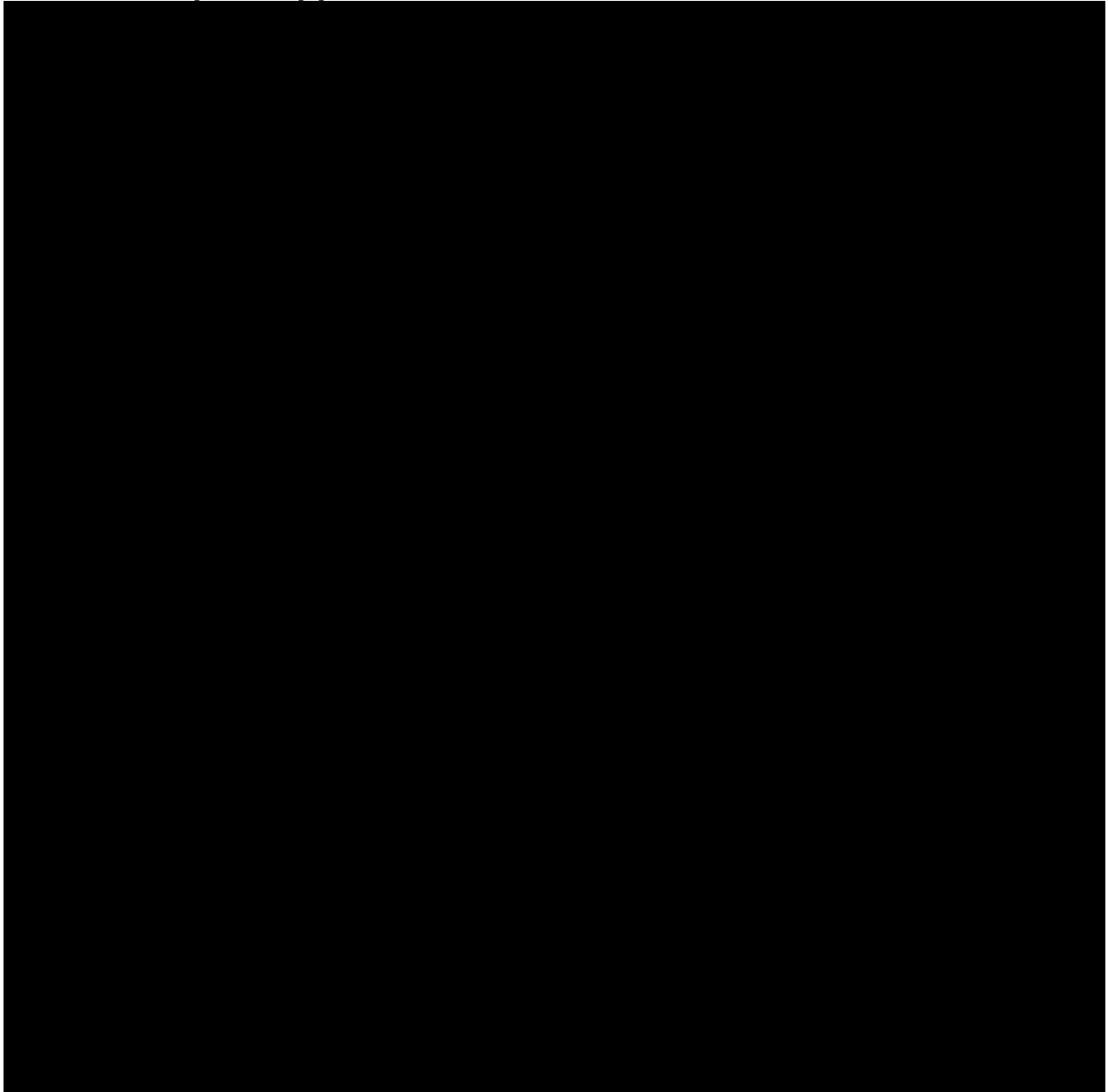


Figure 91: Help & Support

The screens in this section have the same panes as Figure 91; however, the text in the center pane differs with each screen. Sections 8.4.5.1 and 8.4.5.1 contain the text for each screen.

Center pane text:

Choose from the Frequently Asked Questions (FAQs) to learn more about AccessVA, or select the Contact Us link to learn how to contact an AccessVA Service Representative.

8.4.5.1. Frequently Asked Questions (FAQs)

(Refer to Figure 91.)

Center pane text:

- What is AccessVA?
- What's in it for our Veterans?
- Where is AccessVA today?
- What does AccessVA do?
- Do I need to be a Veteran to use AccessVA?
- What is a Credential?
- How do I update my information for a Credential?
- How do I know what Level of Credential I need?
- Who is participating with AccessVA?
- Why should I use AccessVA?
- Can I have more than one AccessVA Credential?
- Do I need more than one AccessVA Credential?
- What is a Level of Assurance?
- What is Identity Proofing?
- How do I receive an AccessVA Credential from the DoD DS Access?
- What is an AccessVA error 403?
- What is an AccessVA error 404?
- What is an AccessVA error 500?
- What is a VAAFI error 403?
- What is a VAAFI error 404?
- What is a VAAFI error 500?
- What if I lose or forget my Password for my DS Logon Credential?
- How do I contact the VA to ask Questions, Submit Compliments, Complaints and Suggestions regarding AccessVA?

What is AccessVA?

Simply put, AccessVA is a single-entry point online where Veterans, beneficiaries, and affiliates log on and are provided with access to multiple government websites and applications. AccessVA allows users to log in once with an issued identification credential (such as username and password combination) and gain access to all AccessVA-enabled websites and applications instantly.

Remembering different login information for each website and application will no longer be required. One set of credentials issued by a participating AccessVA credential provider will allow users access to online resources from the Department of Veterans Affairs and Department of Defense and the ability to conduct business with many applications and websites provided by these government agencies.

What's in it for our Veterans?

The Department of Veterans Affairs exists to serve our Veterans. Veterans will realize substantial benefits by using AccessVA for online transactions including:

1. **Secure and Easy Identity Validations** - AccessVA provides a single identification credential and logon process that Veterans can use for a number of federal information systems across participating agencies. Once an account is established, the Veteran will be spared the burden of having to keep track of multiple sets of identification credentials.
2. **Reduces the Wait for Service and Increases Public Trust** - AccessVA provides improved customer service by enabling a more streamlined record keeping system that allows responsive and timely service for the Veteran while increasing the public's confidence in online business transactions with the Federal Government by preventing potential fraud.
3. **Saves Taxpayer Dollars** - AccessVA promotes efficiencies and cost savings by establishing a unified authentication system that can be interoperable among various agencies that service our Veterans and citizens. By adopting a single system, we save taxpayer dollars.

Where is AccessVA today?

AccessVA is now in an established state. There are multiple VA websites and applications enabled with AccessVA, and there are multiple AccessVA credential providers available for Veterans to use to receive logon credentials. As time moves on, more than twenty VA websites and applications will use the system as well as many Department of Defense websites popular with Veterans.

What does AccessVA do?

In the past, each web site or application you visited wanted to provide you with a username and password for use on that system only. This forced you to keep up with many, often different, usernames and passwords for different web sites. In AccessVA, a username and password that is issued to you (one example of an online credential) will be used at multiple participating VA applications. AccessVA allows participating members to take advantage of work already accomplished by other federation members. As the federation grows, so will the number of VA applications that will accept your credential. One of the main goals is to simply the process for users to do business with the VA electronically.

Do I need to be a Veteran to use AccessVA?

At the current time you need to be a Veteran, Service Member or eligible dependent to acquire AccessVA's most popular credential, DS Logon. Furthermore, you need to be registered in the Defense Enrollment Eligibility Reporting System (DEERS). Most Veterans are already enrolled in this system, spouses and dependents are also often enrolled. At this time there is no AccessVA credential available to the general public but such a credential is planned for the future.

What is a Credential?

The simplest example of an online credential is a username and password pair. You are probably familiar with entering usernames and passwords, and this is all you need for most web sites. However, there are different levels of credentials that are issued based on the security requirements of the application you are accessing. For example, a doctor who is

accessing the medical records of several Veterans across the Internet will need a stronger credential than a simple username and password pair, such as a digital certificate or smartcard. There are different levels of credentials that are issued based on the security requirements of the application you are accessing. The National Institute of Standards and Technology (NIST) has defined four assurance levels. Levels 1 and 2 are user IDs and passwords, while levels 3 and 4 require additional security measures.

How do I update my information for a Credential?

If your personal information, such as last name or address, has changed, you may need to update your Credential information with your Credential provider. Your Credential provider is the organization who assigned and maintains your credential. We are not able to update your credential through AccessVA. You will need to contact your Credential provider to make these updates. The following provides contact information for the CSPs AccessVA accepts, which will help you get started making your information updates:

Credential Provider	Contact Information
[REDACTED]	[REDACTED] [REDACTED] [REDACTED]
[REDACTED]	[REDACTED] [REDACTED] [REDACTED]
[REDACTED]	[REDACTED] [REDACTED]
[REDACTED]	[REDACTED] [REDACTED]

8.4.5.2. Contact Us

(Refer to Figure 91.)

Center pane text:

If you have a question for AccessVA please contact the Health Resource Center (HRC) at 1-877-327-0022. Hours of Operation are Monday - Friday, 8:00 a.m. - 8:00 p.m. (eastern time).

8.4.6. Sign-In Partners Page

This page shows the Sign-In Partners that are available through AccessVA. When a user selects a Sign-In Partner, information about that CSP will display in the About Sign-In Partners section of the screen. This section will display who qualifies for the Sign-In Partner, which applications are accessible after logging in with the Sign-In Partner, and the links to log on or register for the selected Sign-In Partner.

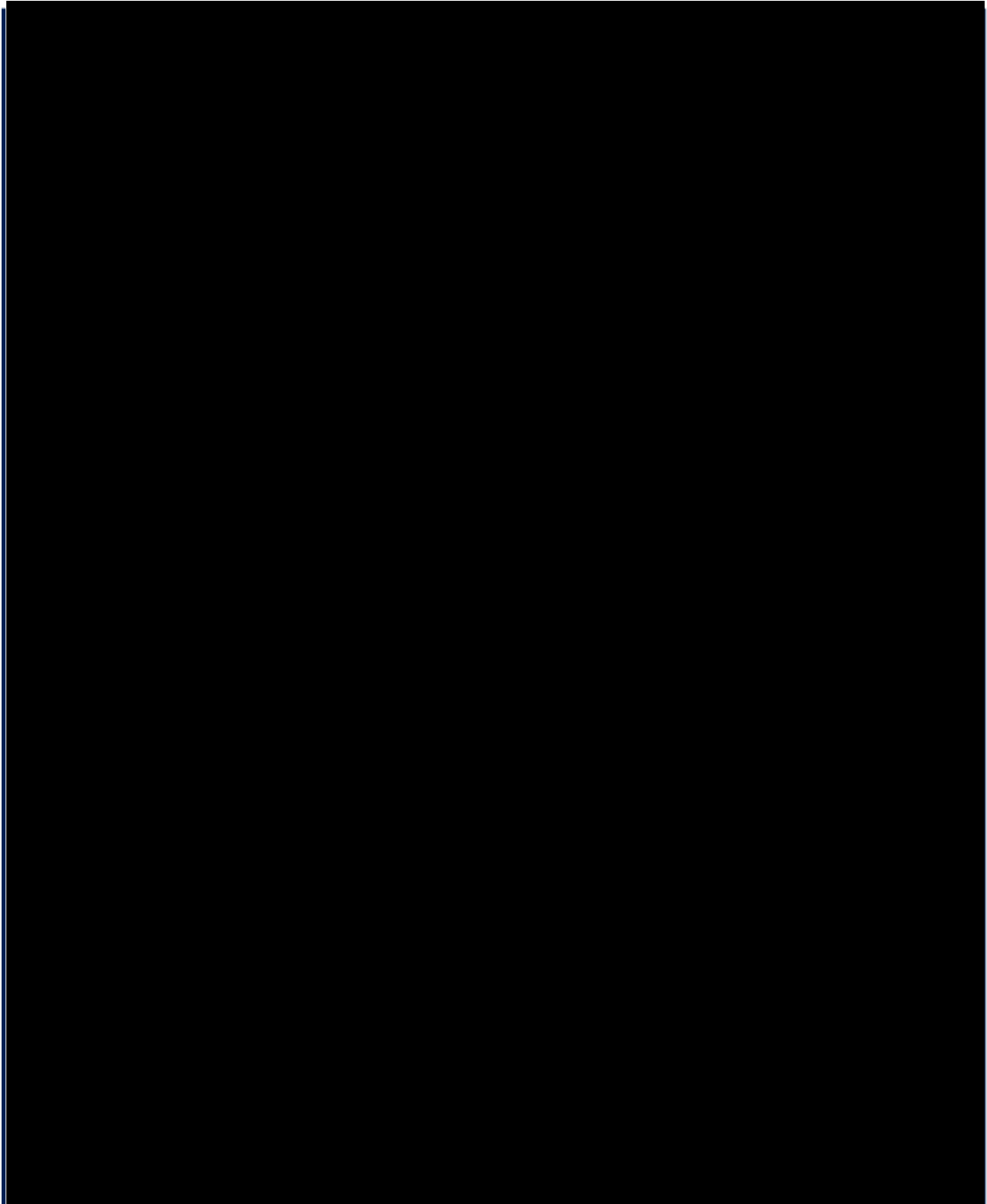


Figure 92: AccessVA Sign-in Partners Page

8.4.7. Application Preselected: MHV Example

Rather than display SSO links for each CSP, applications are encouraged to direct users to AccessVA using the Application Preselected feature. Each application receives an application code. The application will have a logon button that directs the user to AccessVA using that code in the URL in this format: [REDACTED]. For example, the URL to get to AccessVA with MHV preselected is:

[REDACTED] The CSPs that display on this page are fluid, and will display the CSPs that are available for any given application.

8.4.8. Contact Us

(Refer to Figure 91.)

Center pane text:

If you have a question for AccessVA please contact the Health Resource Center (HRC) at [REDACTED]. Hours of Operation are Monday – Friday, 8:00 a.m. – 8:00 p.m. (eastern time).

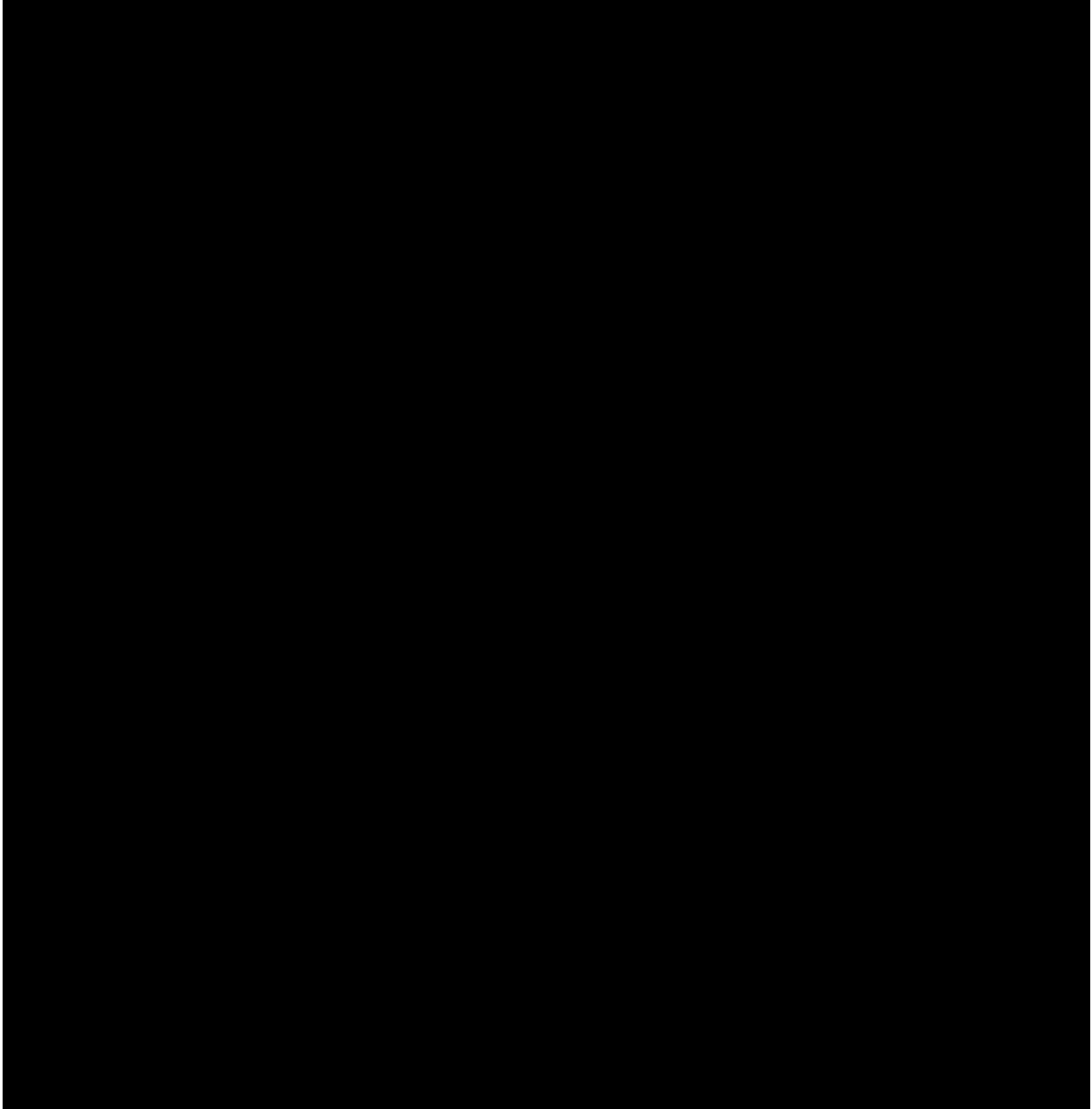


Figure 93: Application Preselected Landing from MHV

8.4.9. Site Down

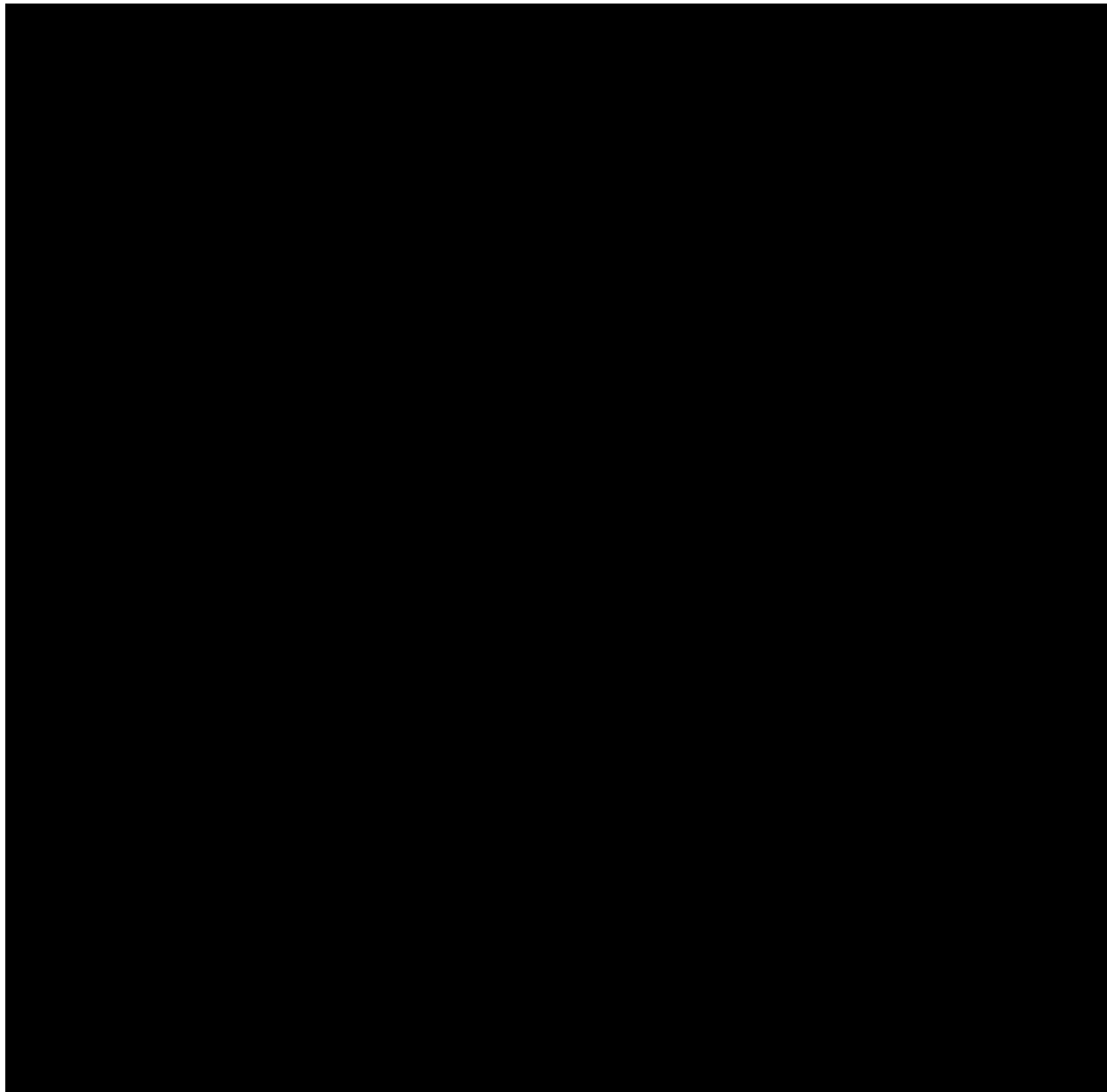


Figure 94: Site Down

8.4.10. Third Party Onboarding Confirmation

As part of the Portal Strategy, each new credential that an individual uses through VAAFI is provisioned. Attributes of the individual map a user to an existing identity or create a new identity in the Master Veteran Index (MVI) and in the Virtual Directory Service. When possible the attributes A CSP sends do this without having the user provide any information. When there is insufficient information to find or create a user, the Third Party Onboarding Confirmation page displays. Section 6.2.6.11 describes the user flows that lead to this page. Section 6.2.10.7 describes the design behind this page. When a user lands on the page, the fields the CSP provides are filled with that data. The user must complete the mandatory fields.

VA User Account Confirmation

We need help with confirming your NOT_FOUND information with our records.

- The information below is what is provided by NOT_FOUND log in.
- To complete the process, we need you to provide the additional required information.
- This process will improve your VA user experience and security while accessing information on the intended website.
- You can complete this later. All users will have to complete confirmation by May 1st 2015.

Confirming your account is only required once for each different credential you use.

Name	Person
First Name*	Gender*
<input type="text"/>	<input type="text"/>
Middle Name	Home or Cell Phone #*
<input type="text"/>	<input type="text"/>
Last Name*	Email*
<input type="text"/>	<input type="text"/>
	Date of Birth*
	<input type="text"/>

Home Address	Identification
Address 1*	SSN*
<input type="text"/>	<input type="text"/>
Address 2	
<input type="text"/>	
Address 3	
<input type="text"/>	
City*	
<input type="text"/>	
State*	
<input type="text"/>	
Zip Code*	
<input type="text"/>	
Country*	
<input type="text"/>	

Fields marked with an asterisk (*) are required.

The fields that are not editable were provided by NOT_FOUND. Please contact NOT_FOUND if you desire to update the non-editable information. For more information on updating NOT_FOUND information, please [Click Here](#).

[I'll do it later](#)

[Submit](#)

[Related Links](#). [Our Privacy Policy](#).

Figure 95: Third Party Onboarding Confirmation –USA

VA User Account Confirmation

We need help with confirming your NOT_FOUND information with our records

- The information below is what is provided by NOT_FOUND log in.
- To complete the process we need you to provide the additional required information.
- This process will improve your VA user experience and security while accessing information on the intended website.
- You can complete this later. All users will have to complete confirmation by May 1st 2015.

Confirming your account is only required once for each different credential you use.

Name	Person
First Name* <input type="text"/>	Gender* <input type="text"/>
Middle Name <input type="text"/>	Home or Cell Phone #* <input type="text"/>
Last Name* <input type="text"/>	Email* <input type="text"/>
	Date of Birth* <input type="text" value="mm/dd/yyyy"/>

Home Address	Identification
Street Address* <input type="text"/>	SSN* <input type="text"/>
City* <input type="text"/>	
Province/Region* <input type="text"/>	
Postal Code* <input type="text"/>	
Country* <input type="text" value="United Kingdom"/>	

Fields marked with an asterisk (*) are required.

The fields that are not editable were provided by NOT_FOUND. Please contact NOT_FOUND if you desire to update the non-editable information.
For more information on updating NOT_FOUND information please [Click Here](#)

Related Links: [Our Privacy Policy](#)

Figure 96: Third Party Onboarding Confirmation – Foreign

8.4.10.1. Successful Confirmation

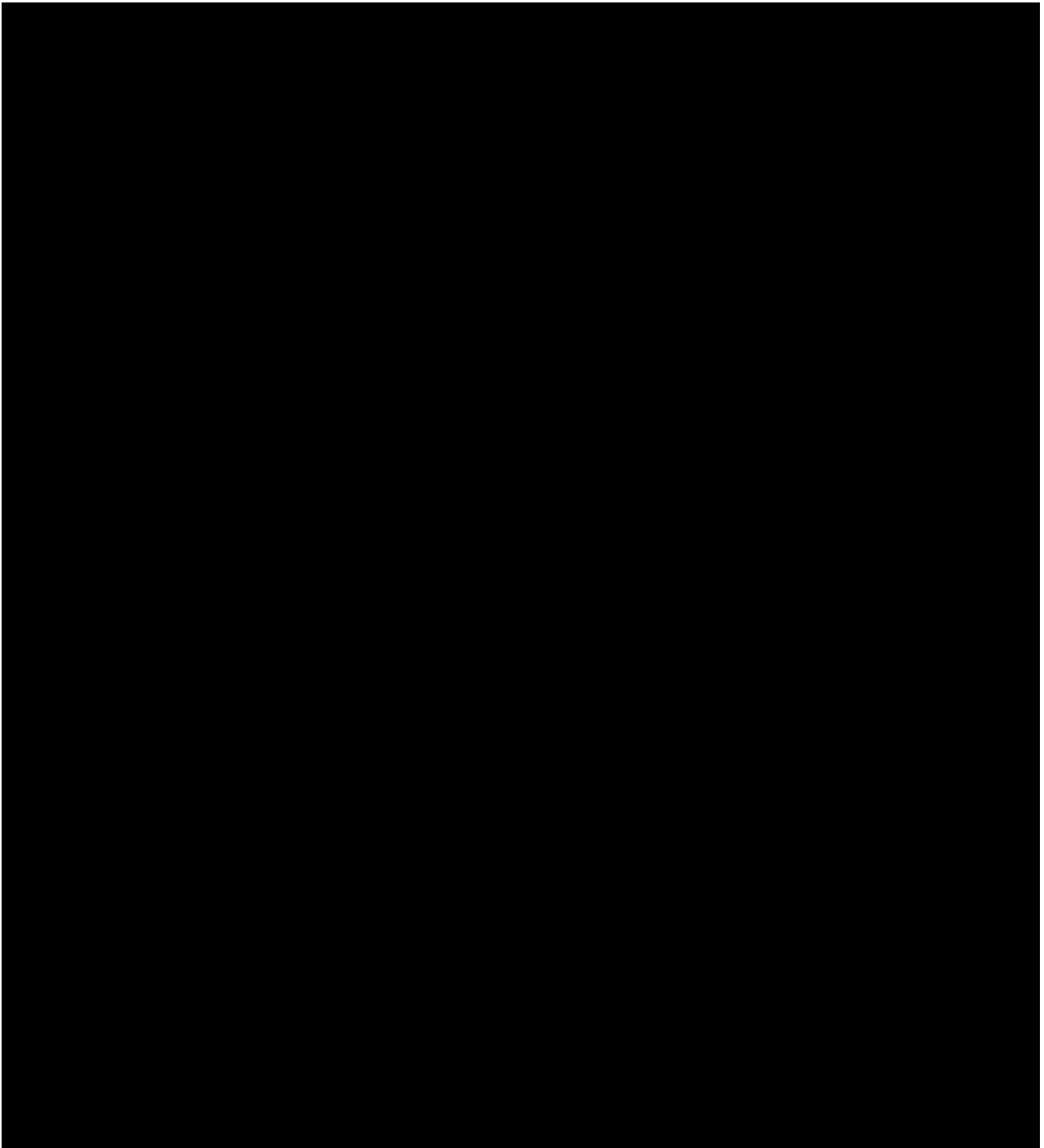


Figure 97: Successful Confirmation

8.4.10.2. Unsuccessful Confirmation

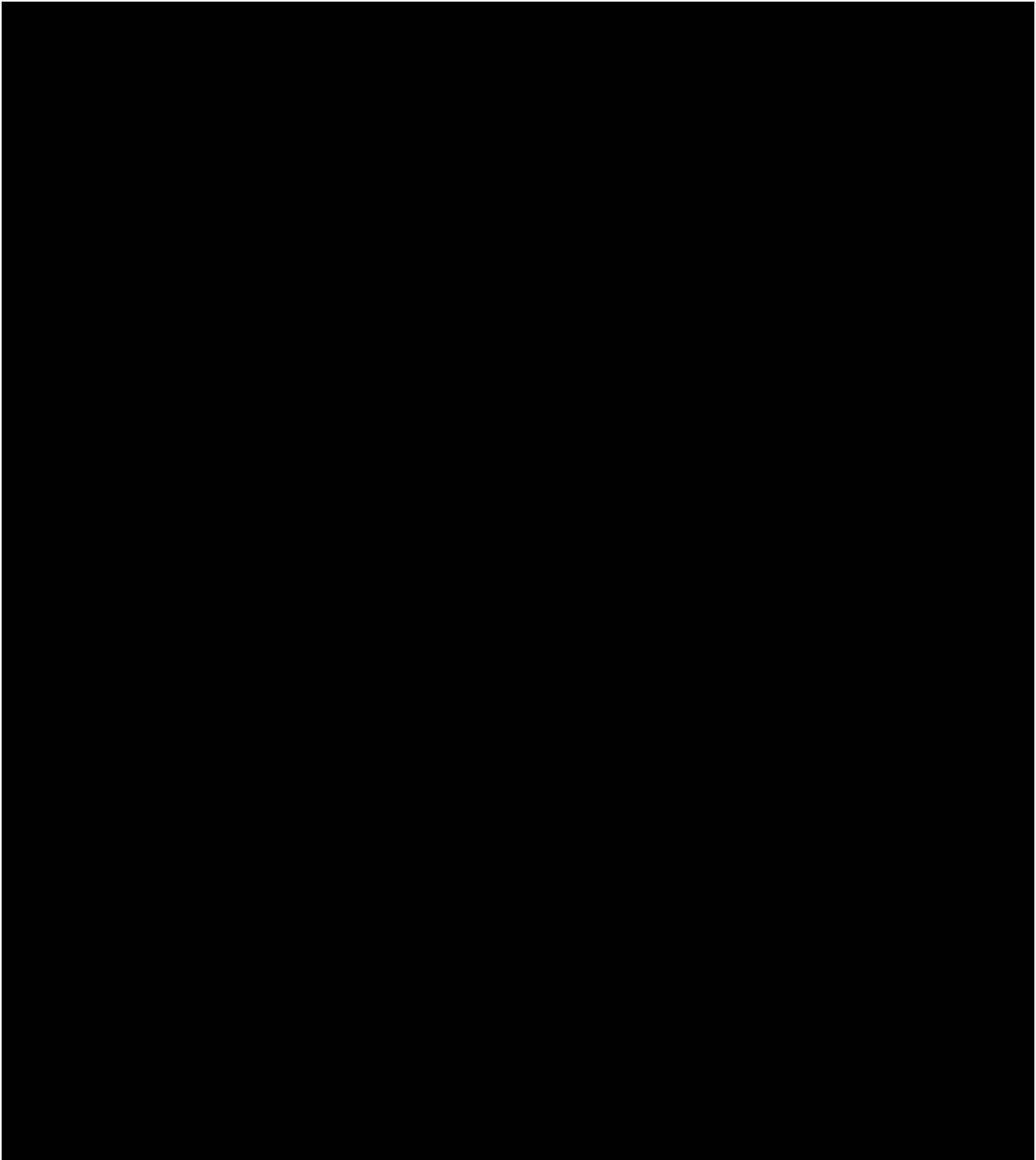


Figure 98: Unsuccessful Confirmation

8.4.10.3. Cancel Confirmation

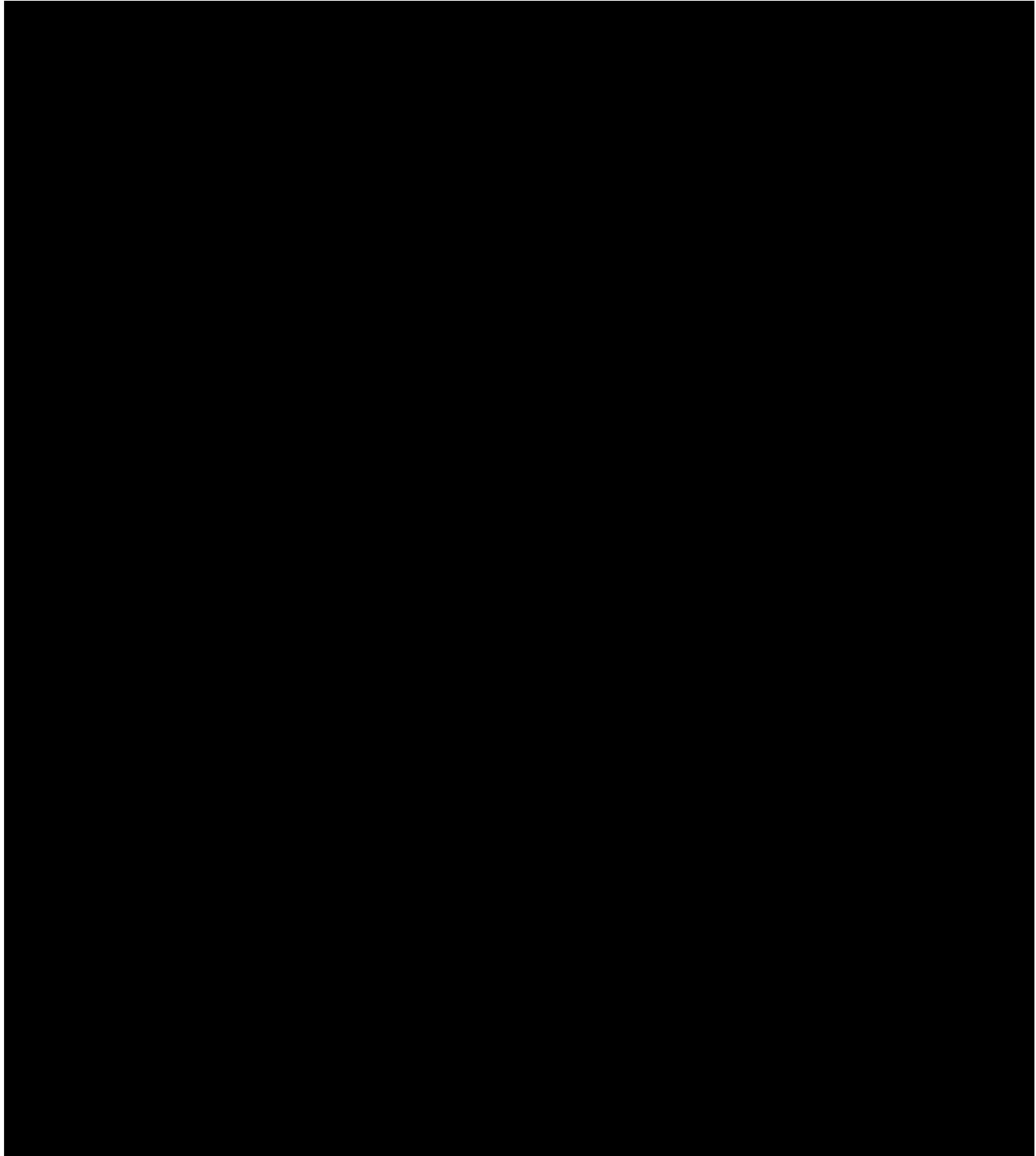


Figure 99: Cancel Confirmation

8.4.11. Global Deny

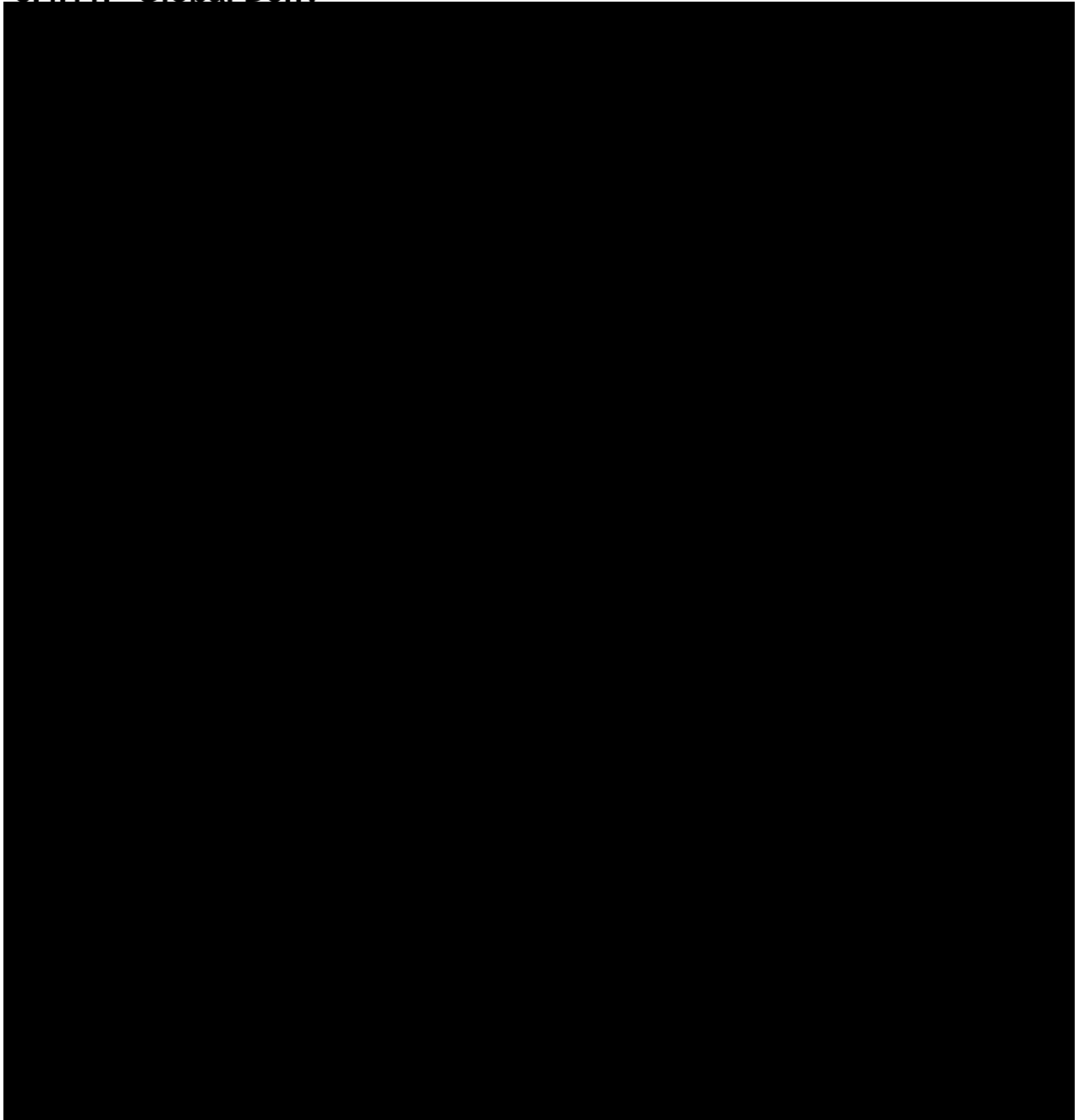


Figure 100 Global Deny Page

8.4.12. VA External Link Notification Dialog

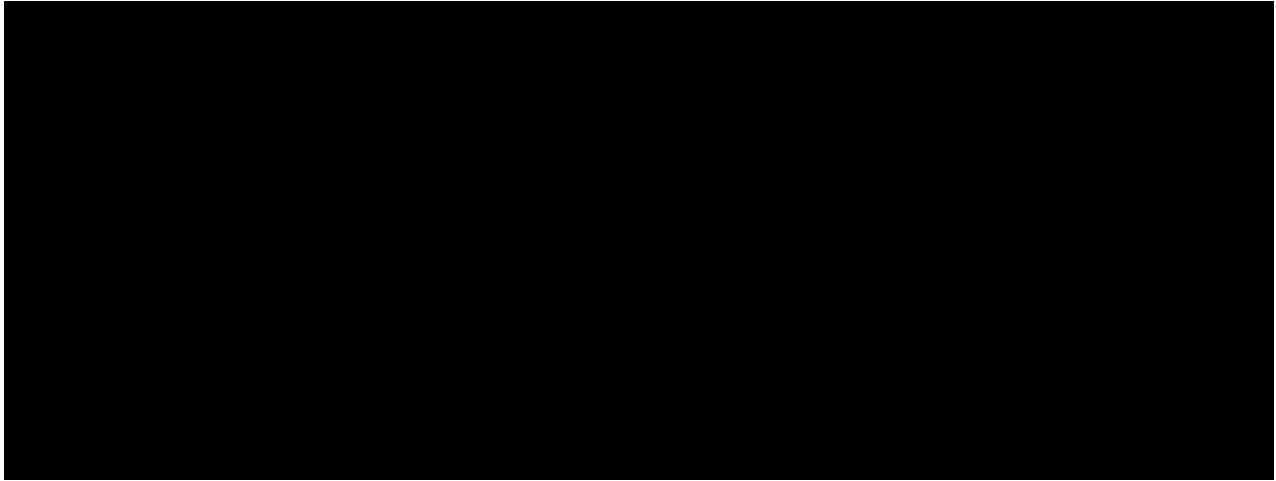


Figure 101: VA External Site Notification

This modal dialog displays when the user is redirected to a CSP login or external link. It includes the ability to cancel the navigation and provides both a warning that the user is being redirected and the approved warning text from VA Directive 6500.

The wording in the dialog will dynamically generate based on the destination.

8.4.13. Error pages

8.4.13.1. VAAFI Insufficient Assurance Level Error

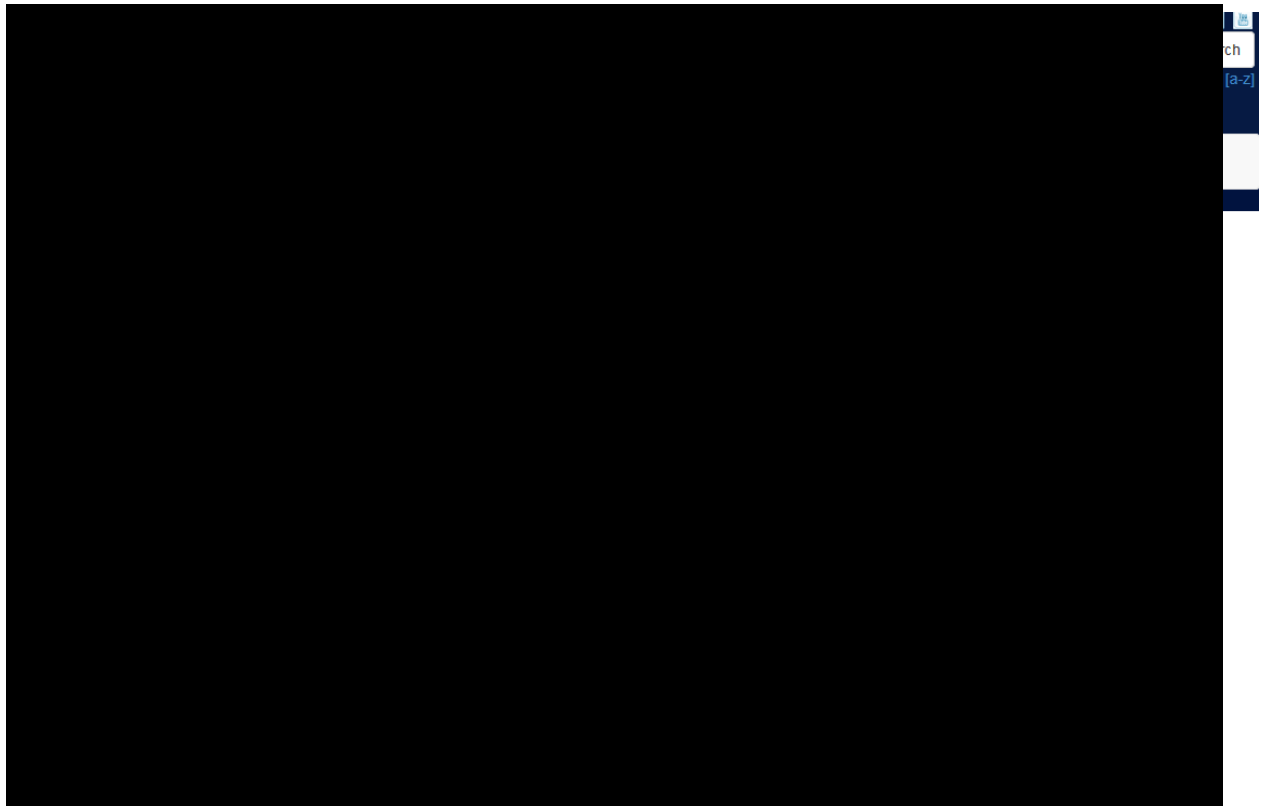


Figure 102: VAAFI Insufficient Assurance Level Error

8.4.13.2. VAAFI Error 404

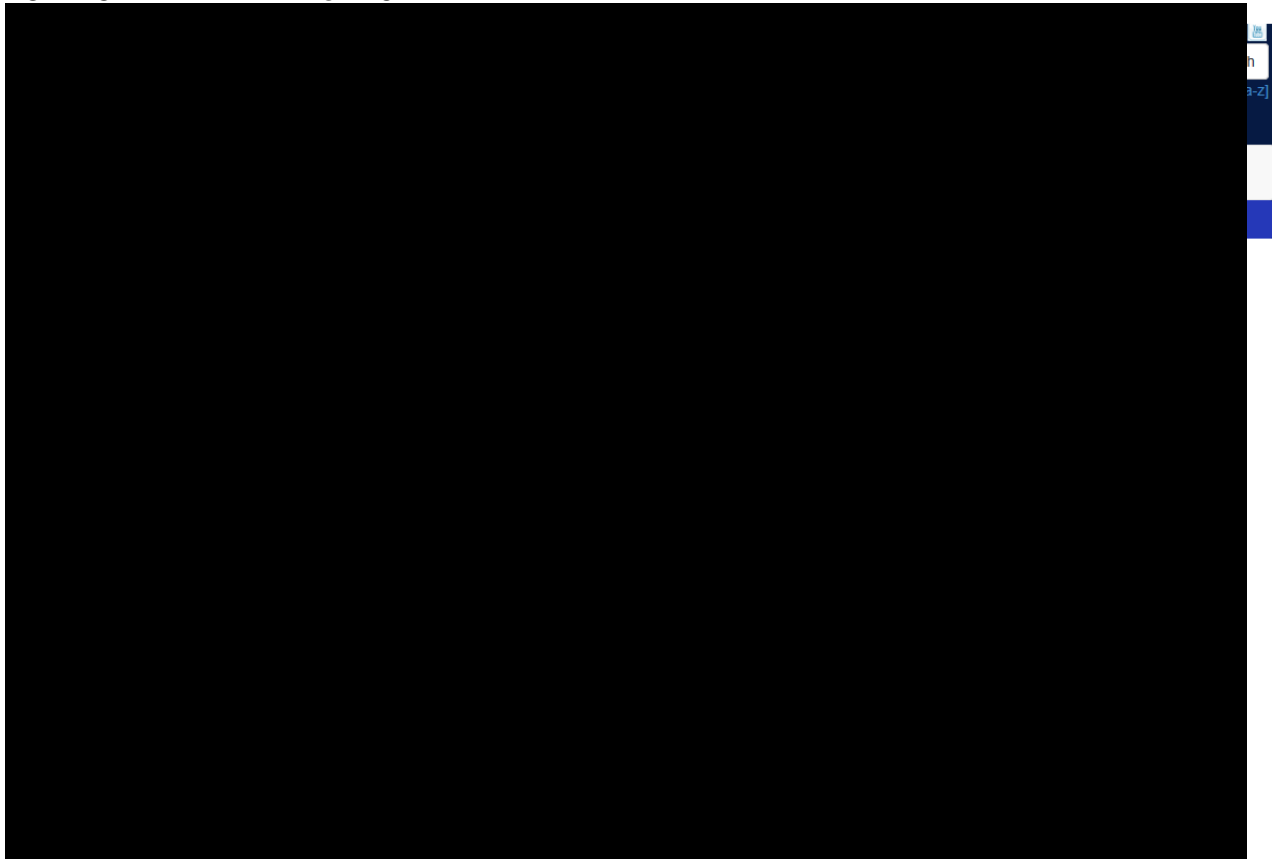


Figure 103: VAAFI Error 404

The remaining error screens in Section 8.4.13 have the same panes as Figure 103; however, the text in the center pane differs with each error screen. Sections 8.4.13.3 through 8.4.13.5 contain the text for each screen.

8.4.13.3. VAAFI Error 403

This is the same as Figure 103 but with text for the HTTP 403 Error Code.

8.4.13.4. VAAFI Error 500

This is the same as Figure 103 but with text for the HTTP 500 Error Code.

8.4.13.5. VAAFI Error 401

This is the same as Figure 103 but with text for the HTTP 401 Error Code.

8.4.14. VAAFI No PKI Error

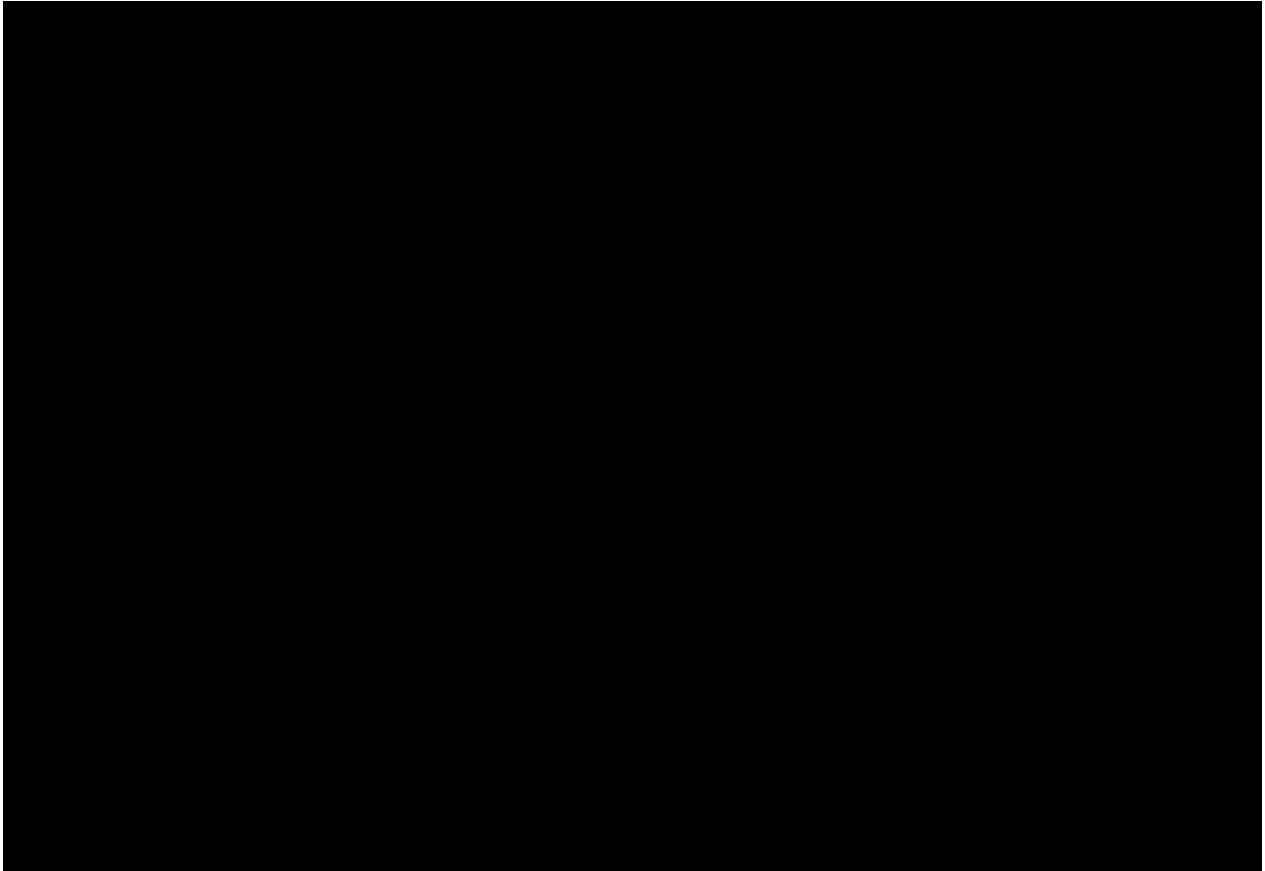


Figure 104: VAAFI No PKI Error

8.4.15. Timeout Pop-up

VA policy requires that an authenticated user's session must disconnect if it is left idle. If an authenticated user does not access an authenticated page for an extended time period, VAAFI session management will terminate their authentication. If a user attempts to access any URL requiring an authenticated session after their session has been terminated, the user will be redirected to AccessVA and a pop-up message will be displayed explaining that the user's session has timed out. After the user dismisses the pop-up dialog, the user will see the application "pre-select" page if the application previously in use could be determined, or the main AccessVA page otherwise

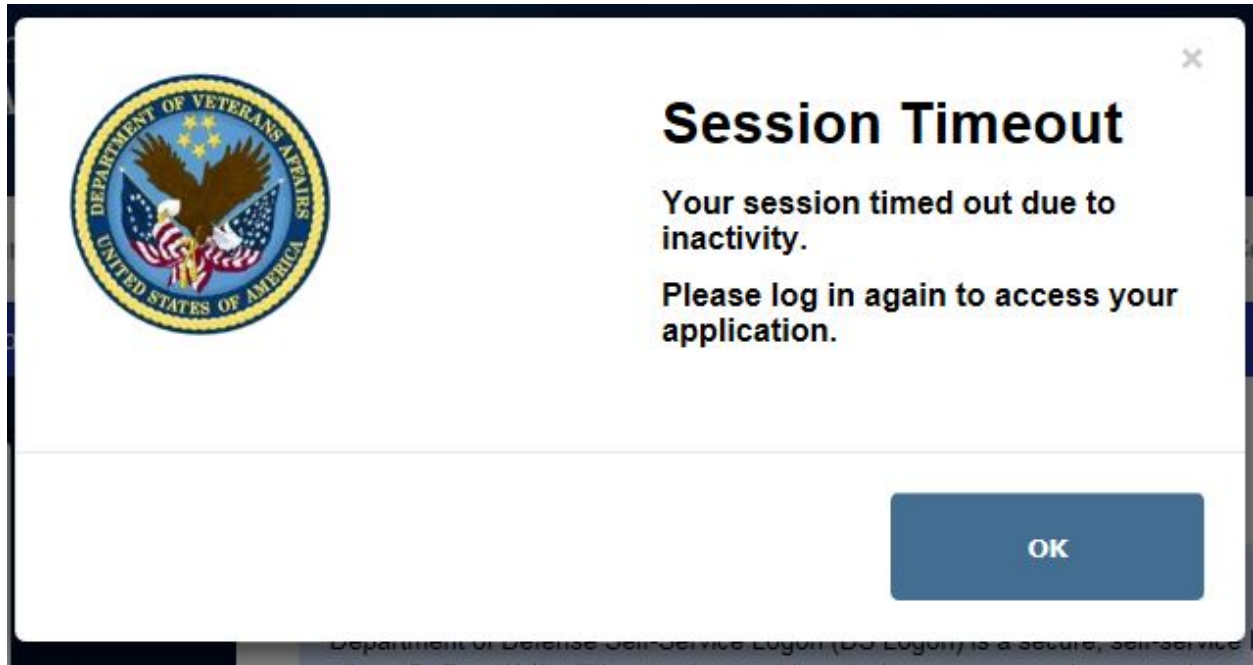


Figure 105: Timeout Pop-up

8.5. PKI CSP Registration

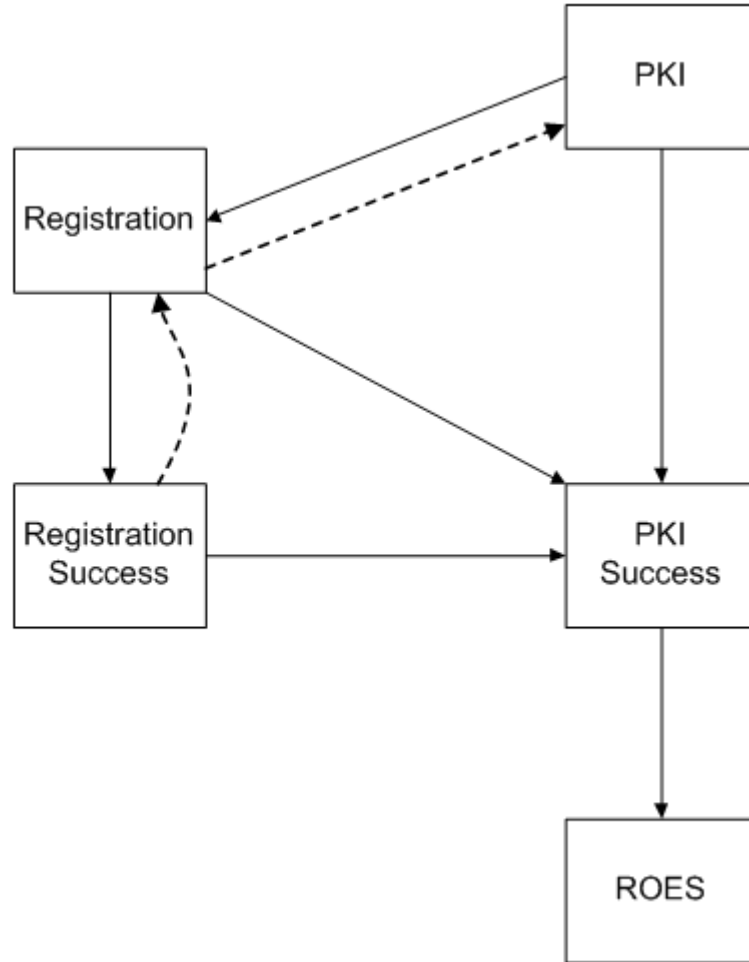


Figure 106: Example of PKI CSP Navigation Hierarchy

CACs and PIVs are registered with the VAAFI CSP when the browser reads the Distinguished Name (DN) from the PKI certificate contained on the card, presents it to the CSP, and the CSP creates an IBM Tivoli Access Manager (ITAM) user with that DN in the IBM Tivoli Directory Server (ITDS) LDAP directory. The ITAM WebSEAL then reads that same DN from the smart card and successfully authenticates ITAM users found in the ITDS LDAP with identical DN to those read from the smart card used to authenticate to the VAAFI CSP. Both the VAAFI PKI Registration Application and VAAFI WebSEAL maintain identical key databases containing the public key chains of trusted Certificate Authorities (CAs). Only PKI credentials that these CAs sign will be allowed to register with and use the VAAFI CSP. The following describes the flow of a user from registration to authentication to accessing VAAFI protected application ROES.

8.5.1. PKI Registration Screen

The user accesses <https://register.eauth.va.gov> with a DoD CAC or VA PIV card in the card reader, and the Active Client software prompts the user for a password. The user enters smart card password and selects OK.

The user is prompted for a certificate to authenticate to [REDACTED]. The user chooses the identity certificate from the user's smart card.

The user is authenticated to the VAAFI PKI Registration application because the Certificate Authority (CA) that signed the user's identity certificate on the smart card (DoD CAC or VA PIV) is a valid CA, trusted by the VAAFI PKI Registration application.

The user then selects the **Register Smart Card** button. Upon successful registration, the user is redirected to AccessVA. Registration failures will be sent to error-specific error pages within AccessVA.

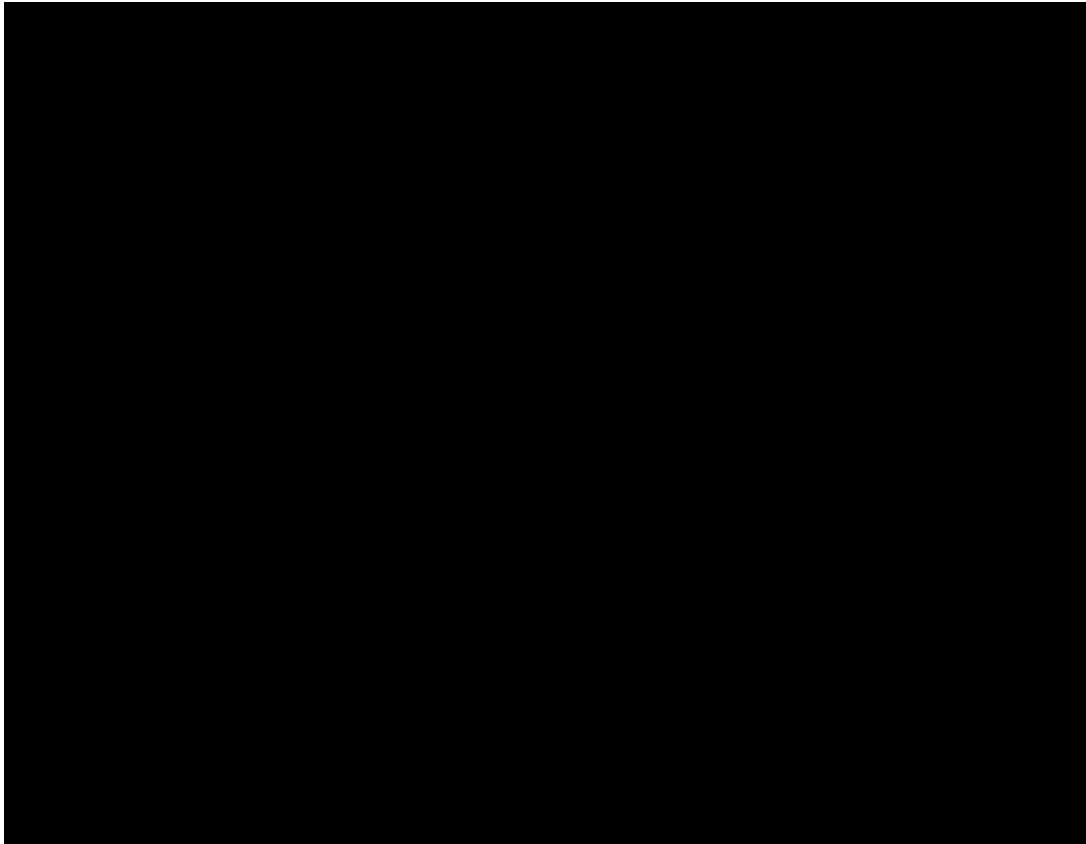


Figure 107: PKI Registration Screen

8.5.2. 401 Error Screen

The user has attempted to access a page the user is not authorized to view at the PKI Registration application and receives an Error 401 page.

8.5.3. 403 Error Screen

The user has attempted to access the PKI Registration application without a PIV or CAC in the card reader and receives an Error 403 page.



Figure 108: PKI Registration Error

8.5.4. 500 Error Screen

The PKI Registration application encounters an internal error and the user receives an Error 500 page.

8.5.5. 404 Error Screen

The user has attempted to access a nonexistent page at the PKI Registration application and receives an Error 404 page.

8.5.6. No PKI Error

The user has attempted to access <https://pki.eauth.va.gov> without a PIV or CAC in the card reader, or the smart card certificate has expired or has been revoked and the user receives an error page, Figure 104.

The VAAFI web site provides Veterans with links to VA applications and CSPs in a single web page. The portal also provides information that explains Federated Authentication and VA's role with other partners in VAAFI, including a link to a page with Frequently Asked Questions

(FAQs). The VA Web Operation group hosts the portal, which uses templates that VA provides to comply with standard VA header and sidebar formats.

The web site conforms to the VA standards for all its web sites and uses the same interface. The content on the center of the page is customized and several links have been created in the Links item of the navigation bar.

8.6. FIM and WebSEAL Error Pages

The FIM and WebSEAL products include dozens of error pages to assist in properly messaging the user and providing engineers with information for troubleshooting. In this increment, all of these pages are reskinned to match the AccessVA look and feel. Variables defined on the pages will provide Tivoli Error Codes and other data to assist engineers to troubleshoot issues.

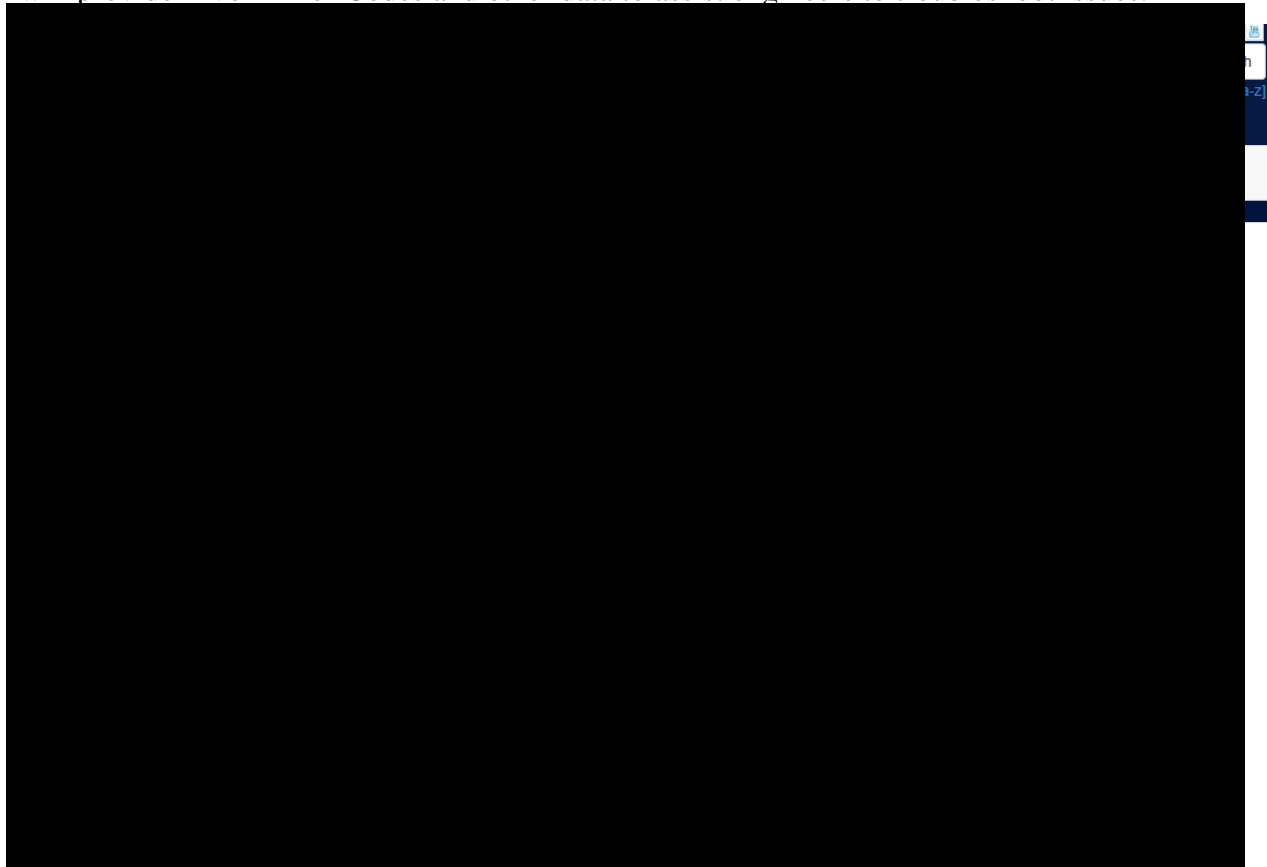


Figure 109: Sample Error Page with a FIM Error Code

In addition to the stock information these products provide, certain pages have added error codes to match the error codes described in the Help Desk Training package and other VAAFI documentation. Descriptions of these error pages are in Section 6.2.6.12.

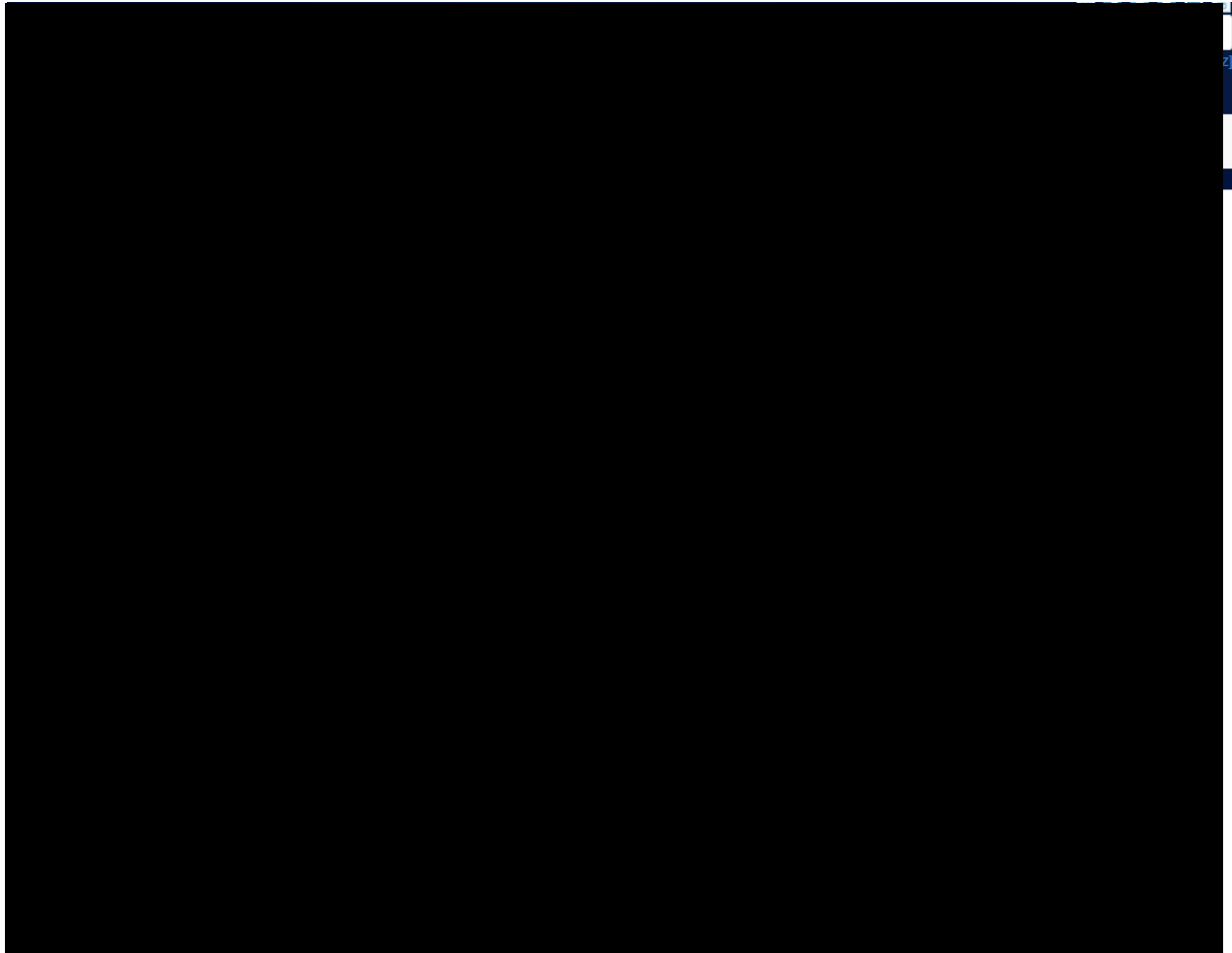


Figure 110: Error Page with a WebSeal Error Code

8.7. OAuth Navigation Hierarchy

The OAuth solution consists of the TFIM OAuth Implementation and the custom web-based application to supplement TFIM's OAuth features. Details will be provided in Sprints 3 through 5.

NOTE: Some of the TFIM OAuth pages are customized to accommodate various screen sizes since OAuth is widely used by mobile devices.

8.7.1. OAuth Consent Management Screen

User can provide his/her consent decision by clicking the radio button (**Permit/Deny**) and clicking the **Submit** button.

UNITED STATES
DEPARTMENT OF VETERANS AFFAIRS



Consent to Authorize

The Client **Public Client**

is requesting access to the following token scope(s)

☒ appointmentProfile

Permit ☒

Deny ☐

Figure 111: OAuth Consent Management

8.7.2. OAuth Response Page

This page displays the Authorization Code/Access Token when a redirection URL has not been assigned to the client.

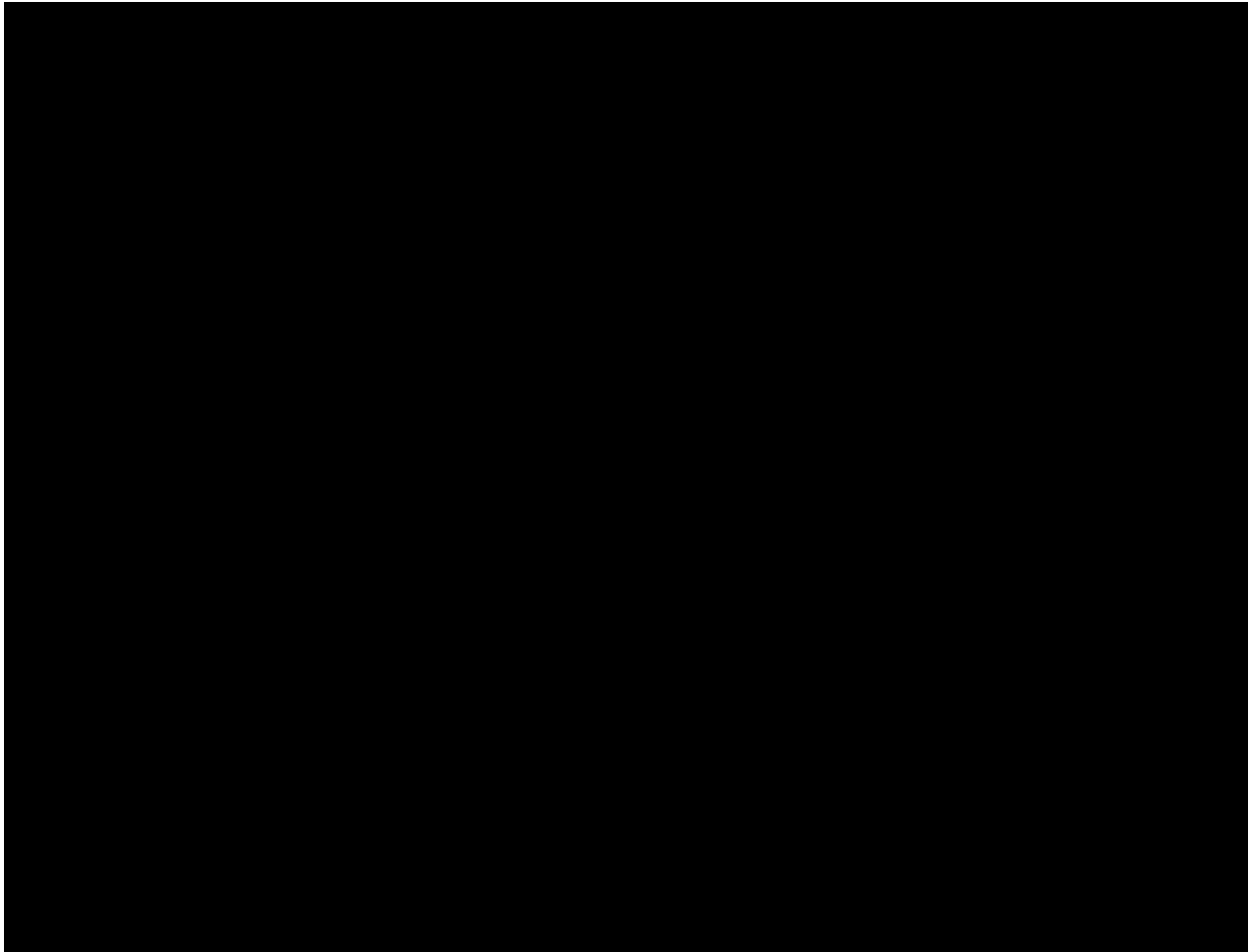


Figure 112: OAuth Response

8.7.3. OAuth Error Page

The TFIM OAuth runtime uses a generic error page. TFIM runtime components dynamically populate Error Code and Error Description based on the underlying error message.

Figure 113: OAuth Error

NOTE: Error Code and Error Description are parameters that the TFIM runtime components populate by based on the underlying errors.

The web pages in 8.7.4, 8.7.5, and 8.7.6 support the web application for supporting client registration, device registration, and consent management.

8.7.4. Client Registration Pages

8.7.4.1. Client Registration - Main

An administrator can manage OAuth client registrations. The administrator can view, enable, disable and delete OAuth Clients and initiate a new client registration.

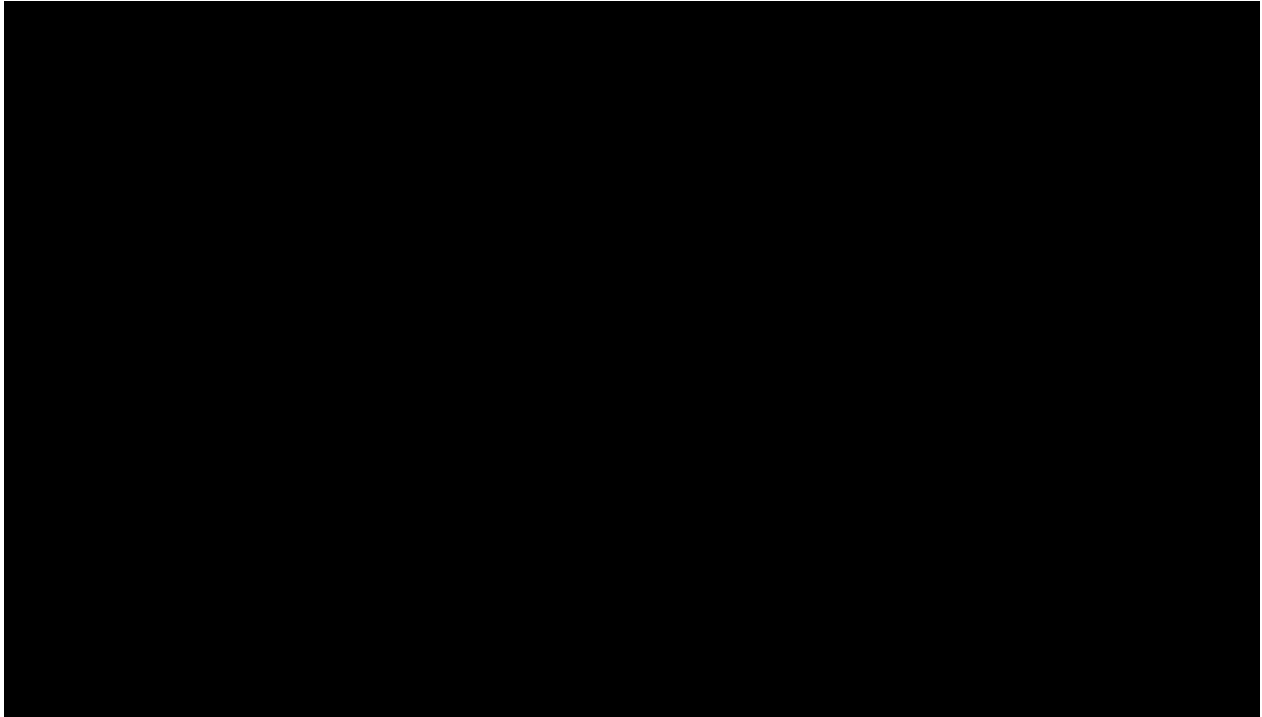


Figure 114: Client Registration – Main

8.7.4.2. Client Registration – New Client Registration

The administrator can register a new OAuth client.



U.S. Department
of Veterans Affairs

[AccessVA Home](#) | [About](#) | [Help & Support](#)

Client Registration

Client Identifier * :

Friendly Application Name :

Redirect URI :

☒ Public
☐ Confidential

[Return to view the Registered Clients](#)

Figure 115: Client Registration – New Client Registration

8.7.5. Device Registration Page

Users can register their devices and provide authorization for the device to participate in the OAuth flow.

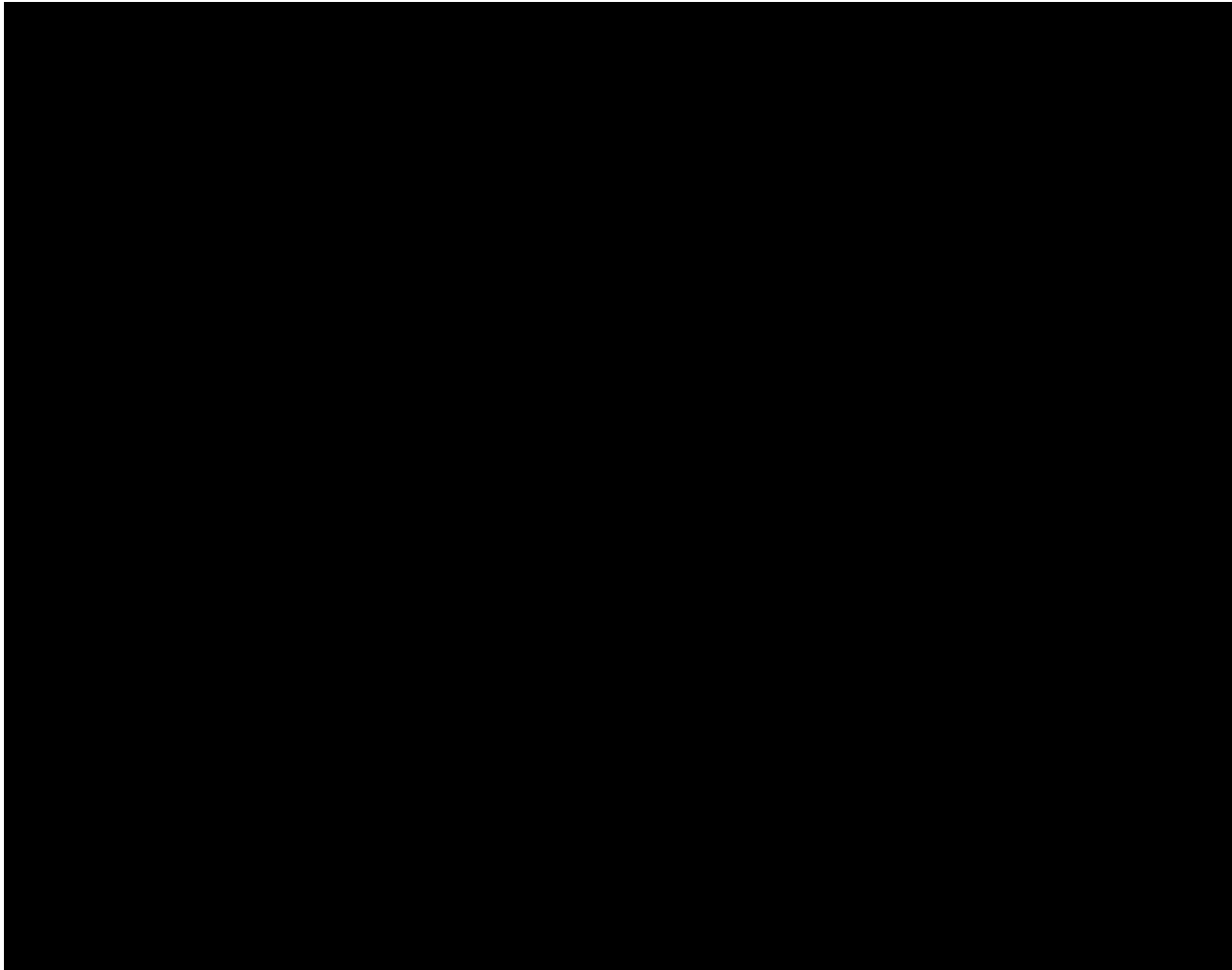


Figure 116: Device Registration

8.7.6. Token Consent Management Page

User can manage Authorization code/Access Tokens that have been issued for this particular user. User can view, delete issued Authorization code/Access Tokens.

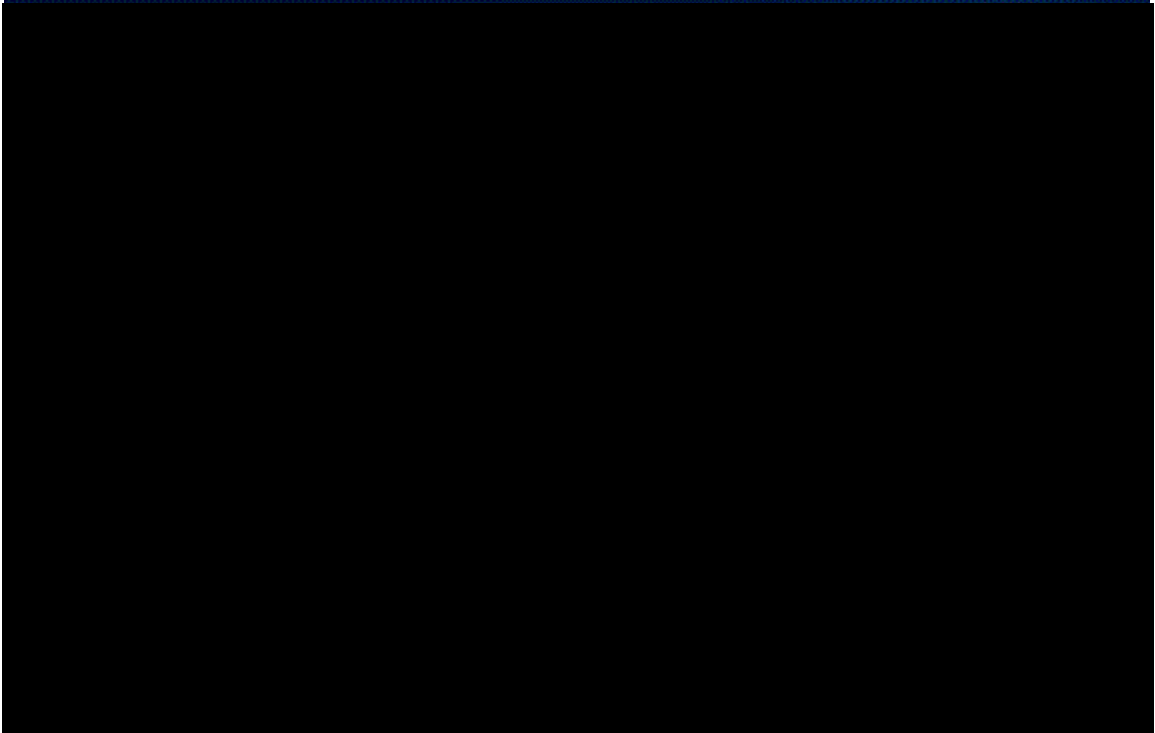


Figure 117: Token Consent Management

9. Security and Privacy

9.1. Security

For information on the system integrity controls, see the VAAFI System Security Plan. AccessVA relies on the system integrity controls inherent in VAAFI.

9.2. Privacy

For information on privacy controls, see the VAAFI System Security Plan and Privacy Impact Assessment... AccessVA relies on the privacy controls inherent in VAAFI.

Attachment A – Approval Signatures

This section is used to document the approval of the System Design Document. The review should be conducted face to face where signatures can be obtained ‘live’ during the review. If unable to conduct a face-to-face meeting then it should be held via LiveMeeting and concurrence captured during the meeting. The Scribe should add/es/name by each position cited. Example provided below.

The Chair of the governing Integrated Project Team (IPT), Business Sponsor, IT Program Manager, and Project Manager are required to sign.

	Date
IAM Integrated Project Team (IPT) Chair and Business Sponsor	

	Date
IAM Program Manager	

Signed: _____

	Date
Access Services Project Manager	

Signed: _____

	Date
Chief Architect	

Signed: _____

	Date
Service Delivery and Engineering	

Appendix A. Additional Information

Attach any addition information that supplements the design specification.

A.1 RTM

The Requirements Traceability Matrix (RTM) is provided as a separate document.

A.2 Packaging and Installation

The packaging and installation information is in the VAAFI installation and configuration guides.

A.3 Design Metrics

VAAFI is a COTS implementation and therefore does not have any design metrics.

A.4 Acronym List and Glossary

The abbreviations and *terms used in this document are defined in the IAM Services Master Glossary.*

Table 46: Glossary

Term	Meaning
See IAM Services Master Glossary	

A.5 Required Technical Documents

This section is not applicable.

A.6 STSUniversalUser Document Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:ibm:names:ITFIM:1.0:stsuuser"
xmlns:stsuuser="urn:ibm:names:ITFIM:1.0:stsuuser"
elementFormDefault="qualified">

    <xsd:element name="STSUniversalUser">
        <xsd:complexType>
            <xsd:sequence>

                <xsd:element name="Principal"
type="stsuuser:PrincipalType"
minOccurs="1" maxOccurs="1"/>

                <xsd:element name="GroupList"
type="stsuuser:GroupListType"
minOccurs="0" maxOccurs="1"/>

                <xsd:element name="AttributeList"
type="stsuuser:AttributeListType"
minOccurs="0" maxOccurs="1"/>

            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

```

                                <xsd:element
name="RequestSecurityToken" type="stsuuser:RequestSecurityTokenType"
    minOccurs="0" maxOccurs="1"/>
                                </xsd:sequence>
                                <xsd:attribute name="version" type="xsd:string"
use="required"/>
                                </xsd:complexType>
                                </xsd:element>

                                <xsd:complexType name="PrincipalType">
                                    <xsd:sequence>
                                        <xsd:element name="Attribute"
type="stsuuser:AttributeType"
    minOccurs="0" maxOccurs="unbounded"/>
                                    </xsd:sequence>
                                </xsd:complexType>

                                <xsd:complexType name="RequestSecurityTokenType">
                                    <xsd:sequence>
                                        <xsd:element name="Attribute"
type="stsuuser:AttributeType"
    minOccurs="0" maxOccurs="unbounded"/>
                                    </xsd:sequence>
                                </xsd:complexType>

                                <xsd:complexType name="AttributeType">
                                    <xsd:sequence>
                                        <xsd:element name="Value" type="xsd:string"
minOccurs="0" maxOccurs="unbounded"/>
                                    </xsd:sequence>
                                    <xsd:attribute name="name" type="xsd:string"
use="required"/>
                                    <xsd:attribute name="type" type="xsd:string"
use="optional"/>
                                    <xsd:attribute name="nickname" type="xsd:string"
use="optional"/>
                                    <xsd:attribute name="preferEncryption"
type="xsd:boolean" use="optional"/>
                                </xsd:complexType>

                                <xsd:complexType name="AttributeListType">
                                    <xsd:sequence>
                                        <xsd:element name="Attribute"
type="stsuuser:AttributeType"
    minOccurs="0" maxOccurs="unbounded"/>
                                    </xsd:sequence>
                                </xsd:complexType>

                                <xsd:complexType name="GroupListType">
                                    <xsd:sequence>
                                        <xsd:element name="Group" type="stsuuser:GroupType"
minOccurs="0" maxOccurs="unbounded"/>
                                    </xsd:sequence>
                                </xsd:complexType>

                                <xsd:complexType name="GroupType">
                                    <xsd:sequence>

```

```
        <xsd:element name="Attribute"
type="stsuuser:AttributeType"
    minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
    <xsd:attribute name="type" type="xsd:string" use="optional"/>
</xsd:complexType>

</xsd:schema>
```

A.7 Attach Documents

Once the SDD is approved, submit the AERB Design Compliance Decision Certificate as an attachment to the completed and approved SDD.

Template Revision History

Date	Version	Description	Author
January 2015	2.8	Updated to latest Section 508 guidelines and remediated with Common Look Office Tool	Process Management
September 2014	2.7	Adds Enterprise Shared Services terms and requires AERB Compliance Certificate attachment.	Process Management
August 2014	2.6	Signature block update authorized by AERB CR_018934	Process Management
March 2014	2.5	Section 508 repairs to new version approved by AERB Chair approved	Process Management
August 2013	2.3	Replaced the Service Architecture sub-section with new sub-sections for consumed and provided services. Also applied miscellaneous feedback from VA team.	ASD Enterprise Shared Services (ESS) Work Group
June 2013	1.3	Upgraded to MS Office 2007-2010 format	Process Management
June 2013	1.2	Address inconsistencies in Section 3, Conceptual Design, Correct headings	Process Management
March 2013	1.1	Formatted to documentation standards and edited for Section 508 conformance	Process Management
January 2013	1.0	Initial Document	PMAS Business Office

See TOGAF® 9.1, Part III: ADM Guidelines & Techniques, Gap Analysis on TOGAF website at <http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap27.html>