VA Enterprise Design Patterns:

Privacy and Security

Secure Messaging Enterprise Design Pattern

Office of Technology Strategies (TS)
Architecture, Strategy, and Design (ASD)
Office of Information and Technology (OI&T)

Version 2.0

Date Issued: September 2016

THIS PAGE INTENTIONALLY LEFT BLANK FOR PRINTING PURPOSES

APPROVAL COORDINATION

Gary E. Marshall 137891

Digitally signed by Gary E. Marshall 137891
DN: dc=gov, dc=va, o=internal, ou=people, 0.9.2342.19200300.100.1.1=gary.marshall@va.gov, cn=Gary E. Marshall 137891
Date: 2016.09.15 07:32:56 -04'00'

Gary Marshall
Director, Technology Strategies, ASD

PAUL A. TIBBITS 116858

Digitally signed by PAUL A. TIBBITS 116858
DN: dc=gov, dc=va, o=internal, ou=people, 0.9.2342.19200300.100.1.1=paul.tibbits@va.gov, cn=PAUL A. TIBBITS 116858
Reason: I am approving this document.
Date: 2016.09.28 10:36:46 -04'00'

Paul A. Tibbits, M.D.
DCIO Architecture, Strategy, and Design

REVISION HISTORY

Version	Date	Organization	Notes
0.1	1/21/2015	ASD TS	Initial Draft/Outline
0.5	2/18/2015	ASD TS	Updated draft incorporating stakeholder feedback
1.0	4/29/2015	ASD TS	Final version with ASD TS contractor team internal revisions and Government Lead validation
1.1	7/26/2016	ASD TS	Updated document to new to Enterprise Design Pattern template
1.5	8/5/2016	ASD TS	Incorporated use cases and additional information on API gateways
1.7	8/18/2016	ASD TS	Incorporated stakeholder comments

REVISION HISTORY APPROVALS

Version	Date	Approver	Role
1.0	4/29/2015	Joseph Brooks	Enterprise Design Pattern Lead

TABLE OF CONTENTS

1 INTRODUCTION 3

1.1 BUSINESS PROBLEM 3
1.2 BUSINESS NEED 3
1.3 BUSINESS CASE 4
1.4 APPROACH 4

2 CURRENT CAPABILITIES 4

2.1 TRANSPORT-LEVEL SECURITY 4
2.2 ENTERPRISE MESSAGING INFRASTRUCTURE (EMI) 5

3 FUTURE CAPABILITIES 6

3.1 MESSAGE LEVEL SECURITY 6
3.2 API GATEWAYS 8
3.2.1 Service Discovery 9
3.3 ALIGNMENT TO THE TECHNICAL REFERENCE MODEL (TRM) 10
3.4 ALIGNMENT TO VETERAN-CENTRIC INTEGRATION PROCESS (VIP) 11

4 USE CASES 11

4.1 USE CASE #1 – EXTERNAL VA SERVICE CONSUMER 11
4.2 USE CASE #2 – INTERNAL VA SERVICE CONSUMER 13

APPENDIX A. SCOPE 15

APPENDIX B. TYPES OF WEB SERVICES 17

APPENDIX C. DEFINITIONS 18

APPENDIX D. ACRONYMS 20

APPENDIX E. REFERENCES, STANDARDS, AND POLICIES 22

FIGURES

Figure 1 – Current VA Enterprise Infrastructure (ESS Security Group) 6
Figure 2 – Service-Oriented Architecture (SOA) Security Pattern 7
Figure 3 – Use Case #1 12
Figure 4 – Use Case #2 13

TABLES

Table 1 – Advantages/Disadvantages of Transport Level Security 5
Table 2 – Advantages/Disadvantages of Message Level Security 8
Table 3 – Analysis of Server-Side vs. Client-Side Discovery 10
Table 4 - List of Approved Tools and Standards for Secure Messaging 11

1 INTRODUCTION

1.1 BUSINESS PROBLEM

Current systems at the Department of Veterans Affairs (VA) cannot ensure full end-to-end security, and therefore compromise the security and privacy of sensitive Veteran data. Hypertext Transfer Protocol Secure (HTTPS), now required for all VA websites, uses transport-level security. Transport-level security only ensures security for point-to-point connections; messages are only protected in transit. Therefore, security ends at *intermediary points* in the Information Technology (IT) infrastructure; messages become unencrypted at these locations. Attackers can intercept unencrypted messages or alter a user's identity credentials as it traverses the network.

Security risks identified with the current infrastructure include:

- Only transport-level security is achieved within the VA network. Transport-level security does not adequately secure systems with multiple intermediary hops. Intermediary hops were introduced by VA's transition to a Service-Oriented Architecture (SOA) enterprise framework.
- The absence of an approved protocol list, to set limits to the number of transport protocols, results in increased security vulnerabilities across the VA Enterprise.
- Some external entities are communicating directly with VA resources, and not through the centralized enterprise SOA infrastructure (e.g., eMI). This limits VA's ability to monitor and filter malicious requests, exposing VA systems to such threats as denial-of-service (DoS) attacks.

1.2 BUSINESS NEED

At present, VA guidance for secure messaging requires the use of transport-level security, which provides only point-to-point security, using methods that include transport layer security (TLS). With significant limitations to managing point-to-point security for systems that require multiple system hops, communication is only secure at the transport level and not the message level. Once the message reaches an intermediary hop, it is no longer secure. VA's common web services security framework does not account for multi-hop messaging. The move towards a SOA-based enterprise infrastructure requires the addition of message-level security. VA will develop guidance on establishing proof of origin of messages, and building a SOA web services trust framework.

1.3 BUSINESS CASE

The VA information technology (IT) infrastructure will meet the increased demand to VA services from both Veterans and VA employees/contractors. To support this increased demand, VA is transitioning to SOA and microservice architectures. While this new architecture brings greater agility and scalability than the existing framework, it also introduces more message routing than what exists at present. VA is responsible for ensuring that messaging can be done securely, while also supporting the increased message load. This Enterprise Design Pattern (EDP) provides guidance on how messaging should be secured once it reaches the VA network. Additional information on microservices can be found in the Microservices EDP. Internal attacks to the VA network will be addressed with the same vigilance as those external to the network. Securing messages to the final destination within the VA network helps to address internal security risks.

1.4 APPROACH

The following steps define a near-term path toward strengthening the security of messages traveling within the VA network. This is particularly relevant for existing production systems that cannot guarantee end-to-end transport security. In the long run, VA will adopt cloud services that enable end-to-end transport security (both for data at rest and data in transit). Cloud services will include built-in functionality to support message-level security. More information on cloud services and cloud security can be found in the Cloud Computing Architecture and Cloud Security Enterprise Design Patterns.

- Gain agreement on standards for incorporating message-level security to VA
- Review the existing capabilities of the API gateway incorporated into eMI
    - Determine whether this API gateway can be extended to an enterprise wide API gateway or if we need to introduce one to the VA network
- Incorporate the enterprise-wide API gateway that includes all required capabilities

2 CURRENT CAPABILITIES

2.1 TRANSPORT-LEVEL SECURITY

Transport-level security via TLS is, at present, used within VA to protect confidentiality. Prior to VA moving towards a SOA web services framework, systems and applications were developed in a monolithic fashion, which accommodated dedicated point-to-point connections that supported end-to-end TLS. However, even with VA's transition to a web services framework, services are integrated using intermediary components, such as an Enterprise Service Bus (ESB). Transport-level security can be interrupted when messages are routed through the ESB to their final destination. TLS is now mandated by the Office of Management and Budget (OMB) M-15-

13 for all interactions with Government resources, and modern hosting platforms, including cloud services, which support end-to-end TLS for all of their services. The following table highlights the advantages and disadvantages of transport-level security.

Table 1 – Advantages/Disadvantages of Transport Level Security

Advantages	Disadvantages
• Does not need any extra coding as protocol inherent security is used • Performance is better as hardware accelerators can be used • Protocol-agnostic; clients do not need to understand standards such as WS-Security as it is built in the protocol itself	• Protocol implemented security that only work from point to point • Security is dependent on the protocol which limits security support and is bounded to the protocol's security limitations

2.2 ENTERPRISE MESSAGING INFRASTRUCTURE (EMI)

The role of the eMI in VA is to minimize point-to-point connections and support a SOA infrastructure in support of VA distributed applications. Figure 1 depicts the current VA enterprise infrastructure and the security mechanisms used for message security. External VA service consumers that conduct two-way communication with VA utilize TLS security standards (FIPS 140-2). These service consumers communicate with VA services that expose either SOAP or Representational State Transfer (REST) Application Programming Interfaces (API). The eMI routes the messages to their proper service providers.

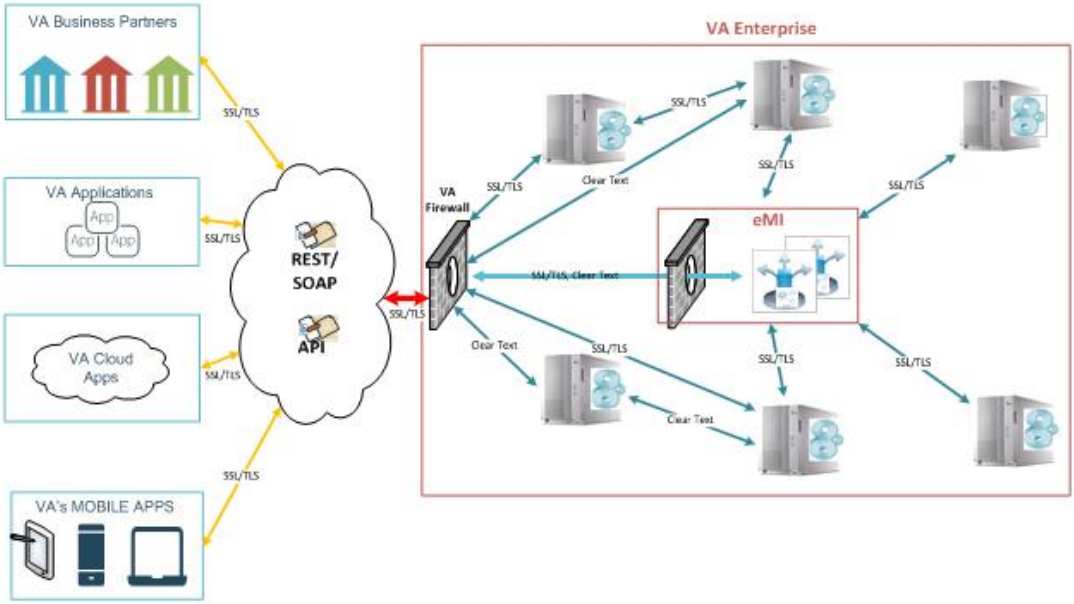

Figure 1 – Current VA Enterprise Infrastructure (ESS Security Group)

3 FUTURE CAPABILITIES

All new VA applications will adhere to the following constraints to ensure message integrity using both message-level and transport-level security mechanisms:

- Use message-level security for service-to-service communication when possible, while utilizing transport-level security otherwise.
- Adhere to WS-Security and associated specifications (e.g. WS-SecureConversation, WS-Trust, WS-Policy) for SOAP-based messages.
- Integrate with VA enterprise middleware and Identity and Access Management (IAM).

The above constraints generally apply to all solutions that integrate VA services, including Enterprise Shared Services (ESS). Furthermore, the following are existing limitations that need to be considered when implementing secure messaging.

- Currently, web services standards do not include security information within the core interface definition language, Web Services Description Language (WSDL). Security information will be provided "out of band."
- X.509 certificates may be required to support digital signatures, authentication, and message encryption.
- Granular authorization will still be handled within the context of the service being invoked.
- The service implementation may need to have knowledge of the user identity and manage permissions internally.
- Presently, only SOAP provides standardized message-level security.

3.1 MESSAGE LEVEL SECURITY

Message-level security provides end-to-end security, transport independence, and security of stored messages. The following figure depicts the message layer involving a message traversing between a service consumer and a service provider. Security information is applied at the message layer and travels along with the message. The message header contains the security header information which includes the security token, digital signature, and encryption information. Message layer security differs from transport-layer security in that it can be used to decouple protection from transport. Message-level security directly encrypts and signs the

message to ensure that messages remain protected after transmission, regardless of how many hops they travel.

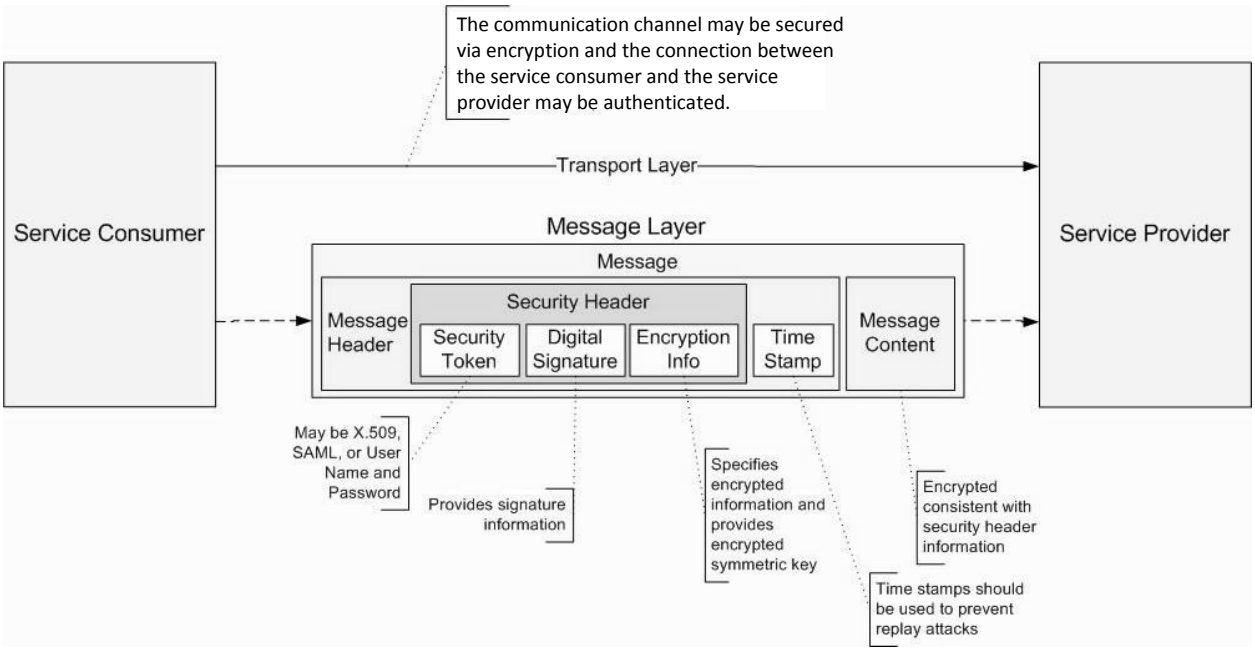

Figure 2 – Service-Oriented Architecture (SOA) Security Pattern[1]

Message-level security is required for:

- End-to-End Security: Secure transport protocols can only assure the security of messages during transmission. When intermediaries receive and process messages, secure end-to-end communication is not possible unless the intermediaries are completely trusted.
- Transport Independence: Even if all the communication links are secure and the intermediaries can be trusted, security information, such as the authenticity of the originator message, must be translated to the next secure transport protocol along the message path. This adds complexity, which in turn increases the risk of security breaches. A best practice is to handle security concerns at the message layer, independently of the transport layers.
- Security of Stored Messages: Once a transmission is received and decrypted, transport-layer security no longer protects data from unauthorized access and modifications. When messages are stored and then forwarded, message layer security is required.

[1]The NIH Enterprise Architecture. (n.d.). *Service-Oriented Architecture (SOA) Security Pattern*.

The following table highlights the advantages and disadvantages of message-level security.

Table 2 – Advantages/Disadvantages of Message Level Security

Advantages	Disadvantages
• As the message is secured (signed and encrypted) while transmitting through the network, any intermediate hop in the network has no impact on security • Being transport-independent, it can support multiple transport options • Supports a wide range of security options, including implementation of custom security • Logged messages will still have sensitive data encrypted	• For legacy systems that cannot support WS-Security, application refactoring is needed to implement security • As every message is encrypted and signed there are performance issues • To support interoperability with legacy systems a façade needs to be developed which would reduce performance

3.2 API GATEWAYS

API Gateways acts as the guardian to the internal VA web services. VA leverages enterprise-grade API Gateways (often referred to as SOA Gateways) to act as an intermediary in guarding the VA's internal web services from untrusted services. The API Gateway acts as the internal web service to the untrusted service and forwards all communication to the internal web service. API Gateways enforce all messages to pass through a hardened gateway first. Additionally, API Gateways restrict access based on source, destination, and WS-Security encryption.

API Gateways also support schema validation, and a subset offers support for SOAP intrusion prevention against attacks that target vulnerabilities native to XML and XML-based services including:

Attack	Description
WSDL scanning	Aims at discovering non-public web services, once their WSDL file is retrieved, by using various common method names
Parameter tampering	A form of Web-based attack in which certain parameters in the URL or Web page form field data entered by a user are changed without that user's authorization

Replay attacks	A form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed
Recursive/oversized payload attacks	An attack which aims at limiting the availability of the targeted web service
External reference attacks	This attack occurs when XML input containing a reference to an external entity is processed by a weakly configured XML parser, leading to the disclosure of confidential data, denial of service, and other system impacts.
Schema poisoning	When an attacker is able to maliciously alter metadata, such as web service address, message format, required or security parameters, and spread them across web service clients
SQL injection	A code injection technique, used to attack data-driven applications, in which nefarious SQL statements are inserted into an entry field for execution

To provide in-depth security, API Gateways are implemented at the perimeter, and support in-depth logging facilities for auditing. The use of WS-Security or HTTPS for all internal web services is implemented as well. Message-level security is applied to the message at the API Gateway where encryption and a digital signature are added. The API Gateway will also handle messages already digitally signed from external sources. Support for federated identities and integration with IAM services (e.g. Single Sign On External (SSOe) Secure Token Service (STS)) will be supported to ensure credentials are not altered as they enter the VA network. More information can be found in the User Identity Authentication Enterprise Design Pattern.

3.2.1 Service Discovery

The API gateway needs to know the location (i.e. IP address and port) of each web service, including microservices, registered to it. While certain infrastructure services will have a static location, application services have dynamically assigned locations and will run in containerized environments. Furthermore, the set of instances of a service changes dynamically due to auto-scaling, failures, and upgrades. To address this issue, the API Gateway will use either server-side or client-side discovery.

Client-side discovery requires the client to be responsible for determining the network locations of available service instances and load balancing requests across them. A service registry for the client to query is established. The service registry is further discussed in the Microservices EDP. With the use of load-balancers, the client is able to select one of the available service instances and make a request.

Server-side discovery requires the client to make a request to a service through a load balancer. The load balancer queries the services registry and routes each request to an available service instance. The details of the service discovery are abstracted away from the client, as all requests are directed to the load balancer.

Table 3 – Analysis of Server-Side vs. Client-Side Discovery

	Server-Side Discovery	Client-Side Discovery
Advantages	• No need to implement discovery logic for each programming language and framework used by service clients • Some cloud providers include this functionality	• Relatively straightforward with no moving parts aside from the service registry • The client can make intelligent, application-specific load-balancing decisions on which service instances to utilize
Disadvantages	• Unless the load balancer is part of the cloud environment, it is setup and managed while providing high availability • More network hops are required than client-side discovery	• The client is coupled with the service registry • Client-side service discovery logic is implemented for each programming language and framework used by the service clients

For the near-term, the advantages of client-side discovery, namely straightforward design and less network hops compared to server-side discovery, make it the preferred method for service discovery. The advantages will provide a more secure message environment. However, as VA looks to move more of its services and infrastructure to the cloud, server-side discovery will need to be re-examined as a preferable option; the load balancer may be integrated into the services offered and the increased efficiencies of the cloud can minimize the extra network hops of server-side discovery.

3.3 ALIGNMENT TO THE TECHNICAL REFERENCE MODEL (TRM)

All projects will leverage the approved tools and standards located in the VA Technical Reference Model (TRM)[2] to comply with the architectural guidance provided in this document. **Error! Reference source not found.** lists the approved tools for this Enterprise Design Pattern.

[2] http://trm.oit.va.gov/

Table 4 - List of Approved Tools and Standards for Secure Messaging

Category	Example Approved Tools and Standards
Message Encryption	XML Encryption, XML Signature, WS-Security
API Gateway	IBM WebSphere DataPower Integration Appliance, CA Gateway (for eMI)
Message Oriented Middleware	IBM Integration Bus, IBM MQ, CA SecureSpan Gateway
Authentication	SAML, OAuth 2.0
Transport-level Encryption	HTTPS, TLS

3.4 ALIGNMENT TO VETERAN-CENTRIC INTEGRATION PROCESS (VIP)

VIP is a Lean-Agile framework that services the interest of Veterans through the efficient streamlining of activities that occur within the enterprise. The VIP framework unifies and streamlines IT delivery oversight, and will deliver IT products more efficiently, securely, and predictably. VIP is the follow-on framework from Project Management Accountability System (PMAS) for the development and management of IT projects. VIP will propel the Department with even more rigor toward Veteran-focused delivery of IT capabilities.

More information can be found here (https://vaww.oit.va.gov/veteran-focused-integration-process-vip-guide/).

4 USE CASES

The following use cases demonstrate application of the capabilities and recommendations described in this document.

4.1 USE CASE #1 – EXTERNAL VA SERVICE CONSUMER

This use case shows the high-level architecture of how messages are secured when an external user accesses VA services.

Assumptions

- The external user has proper authorization to access VA services
- The external user utilizes an application that makes web service calls to access VA services

- Backend services are implemented using a microservices framework, along with certain services still provided via legacy systems

Use Case Description

The use case for an external VA service consumer is displayed in Figure 3.

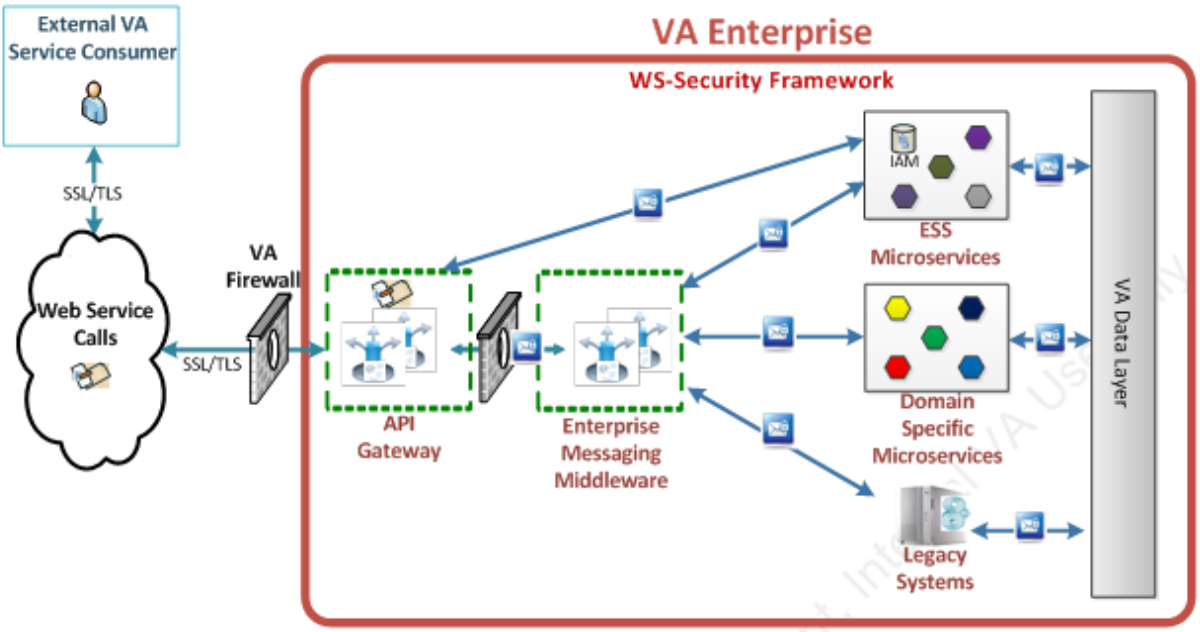

Figure 3 – Use Case #1

The steps for the External VA Service Consumer are as follows:

1. The external VA service consumer accesses VA services through an application via web service calls that are secured through TLS (transport level security).
2. The service call reaches the VA external facing firewall.
3. Authorization is performed by utilizing the IAM service. Once authorized, the message is routed to the API gateway.
4. The API gateway lies between the external and internal facing firewalls, which together form a demilitarized zone (DMZ).
5. From there, the data is fed through the API Gateway, where data mediation, protocol mediation, security checks, and identity federation take place.
6. Message-level security is performed on the message, where encryption and a digital signature are added.
7. The secured message passes through the internal facing firewall and is routed to the enterprise messaging middleware (presently the eMI), where routing, traffic management (SLA), and orchestration takes place.

8. The messages are routed to the requested services, which are logically grouped into legacy system services, ESS, and domain specific microservices.
9. All services utilize the data layer containing the Enterprise Create Read Update Delete (eCRUD), which provides access to the data lake, authoritative data sources (ADS), non-ADS, VA data warehouse, and archival data storage. Further information on the VA data layer is addressed in the Hybrid Data Access EDP.

4.2 USE CASE #2 – INTERNAL VA SERVICE CONSUMER

This use case shows the high-level architecture of how messages are secured when an internal user access VA services.

Assumptions

- The internal user has proper authorization to access the VA services.
- Backend services are implemented using a microservices framework, along with certain services still being provided via legacy systems.

Use Case Description

The use case for an internal VA service consumer is displayed in Figure 4.

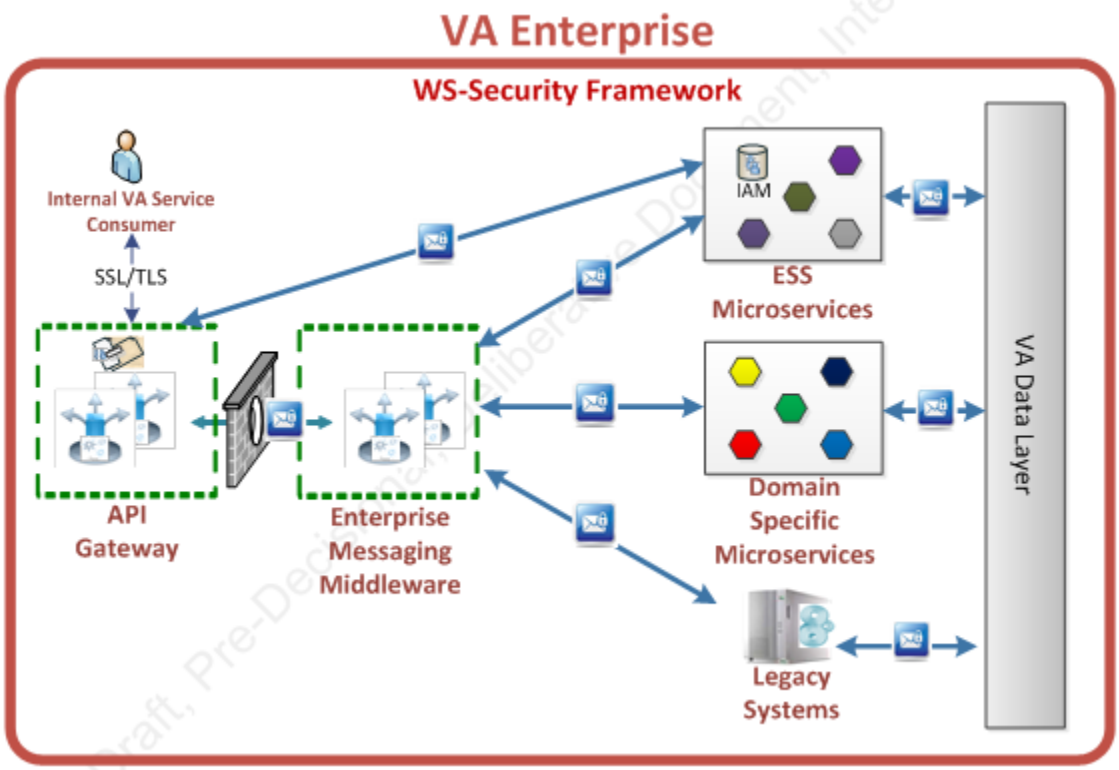

Figure 4 – Use Case #2

The steps for the Internal VA Service Consumer are as follows:

1. The internal VA service consumer accesses VA services through an application, via web service calls, that are secured through TLS (transport level security).
2. The service call reaches the API gateway where data mediation, protocol mediation, security checks, and identity federation take place.
3. Message-level security is performed on the message, where encryption and a digital signature are added.
4. The secured message passes through the internal facing firewall and is routed to the enterprise messaging middleware (currently the eMI), where routing, traffic management (SLA), and orchestration takes place.
5. The messages are routed to the requested services which are logically grouped into legacy system services, ESS, and domain specific microservices.
6. All services utilize the data layer containing the Enterprise CRUD (eCRUD), which provides access to the data lake, authoritative data sources (ADS), non-ADS, VA data warehouse, and archival data storage. Further information on the VA data layer is addressed in the Hybrid Data Access EDP.

APPENDIX A. SCOPE

Background

Enterprise Design Patterns, developed by the Office of Information and Technology (OI&T) and Office of Technology Strategies (TS), provide a generalized architectural framework and guidance to drive solution architecture development to align to best practices, standards, and guidance that support the VA Enterprise Technical Architecture (ETA). This document guides developing and implementing message-level security, using a service-oriented construct.

Enterprise Design Patterns are developed in alignment with the VA Enterprise Technology Strategic Plan (ETSP), which informs the development of design patterns required for VA's enterprise "to-be" strategic vision. Enterprise Design Patterns align to the "Technology Vision" segments of the ETSP, providing further guidance for implementing solution architectures. They help guide programs in the development of IT systems to support compliance with the VA Enterprise Technical Architecture (ETA).

Purpose

Current guidelines for encrypting data transmission within the VA network rely solely on the use of TLS. While TLS is essential for foundational data transmission security, additional security measures are needed to protect data in a SOA environment. This document addresses the SOA security challenges regarding web service communications.

This document expounds on the message-level security standards needed to integrate the enterprise IT infrastructure and Enterprise Shared Services (ESS). It outlines the capabilities and standards achievable through the use of enterprise middleware solutions such as Enterprise Messaging Infrastructure (eMI) and API Gateways. This guidance applies to both SOAP and non-SOAP message exchanges with systems internal and external to VA.

Scope

This document provides a platform-independent framework of secure messaging functionality and refers to design guidelines and reference implementation to guide implementation. This document is applicable to all VA data domains.

The following content is beyond the scope of this document and is referenced in the appropriate locations to guide further technical planning and coordination:

- Overview of enterprise messaging capabilities and message exchange patterns (reference the VA Enterprise SOA Design Pattern document)

- Details for specific messaging standards and transport protocols (reference the ESS Message Exchange Guide)

Intended Audience

This document is meant to be used by all project teams that are developing new applications that are deployed into production within the VA's IT infrastructure. These applications are device-independent, and encompass the acquisition of Commercial Off-the-Shelf (COTS) software (including open-source solutions), intended to meet data sharing requirements. They will make calls to ESS utilizing message-level security standards provided by enterprise messaging middleware.

Document Development and Maintenance

This Enterprise Design Pattern was developed collaboratively with internal stakeholders from across the Department and included participation from VA's Office of Information and Technology (OI&T), Enterprise Program Management Office (ePMO), Office of Information Security (OIS), Architecture, Strategy and Design (ASD), and Service Delivery and Engineering (SDE). The VHA, VBA, and NCA contributed extensive input and participation. In addition, the development effort included engagements with industry experts to review, provide input, and comment on the proposed pattern. This document contains a revision history and revision approval logs in order to track all changes. Updates will be coordinated with the Government lead for this document, which will also facilitate stakeholder coordination and subsequent re-approval, depending on the significance of the change.

APPENDIX B. TYPES OF WEB SERVICES

The following web services (based on Java implementation) apply to the message-level security standards described in this document:

JAX-RPC based web services, in which a service provider publishes the service definition using a WSDL and the service consumer sends a serialized XML message wrapped in a SOAP envelope.

JAX-RS based web services, in which a service provider publishes the resource name that can be used to consume the service, and the service consumer uses stateless operations from the HTTP protocol and sends requests and receives response messages. Because of the stateless nature of the operations, web services are called Representational State Transfer (REST) services. The message payload can either be in XML or JSON format.

RESTful web services differentiate themselves from SOAP-based web services mainly in the simplicity of their design and implementation. However, they can become vulnerable when they are unsecured, especially when serving controlled data. This is quite true of any service, not just RESTful web services. As a result RESTful messages need to be secured. A SOA can be implemented using a number of other technologies, such as Representational State Transfer (REST). This guidance is limited to SOAP-based Web Services, but much of the guidance in this document may be applicable to other SOA technologies.

APPENDIX C. DEFINITIONS

Key Term	Definition
Application Programming Interface	API is a set of routines, protocols, and tools for building software applications. An API expresses a software component in terms of its operations, inputs, outputs, and underlying types. An API defines functionalities that are independent of their respective implementations, which allows definitions and implementations to vary without compromising each other.
Enterprise Shared Service	A SOA service that is visible across the enterprise and can be accessed by users across the enterprise, subject to appropriate security and privacy restrictions.
Secure Socket Layer (SSL)	A standard security technology for establishing an encrypted link between a server and a client—typically a web server (website) and a browser; or a mail server and a mail client (e.g., Outlook).
Service	A mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description
Service Oriented Architecture (SOA)	A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains; it provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations
SOAP	A messaging protocol that allows programs that run on disparate operating systems (such as Windows and Linux) to communicate using Hypertext Transfer Protocol (HTTP) and its Extensible Markup Language (XML)
Transport Layer Security (TLS)	A protocol that ensures privacy between communicating applications and their users on the Internet

Key Term	Definition
XML	Extensible Markup Language is a text-based format that allows for the structuring of electronic documents and is not limited to a set of labels.

APPENDIX D. ACRONYMS

Acronym	Description
ADS	Authoritative Data Sources
API	Application Programming Interface
ASD	Architecture, Strategy and Design
COTS	Commercial Off-the-Shelf
DMZ	Demilitarized Zone
EA	Enterprise Architecture
eCRUD	Enterprise Create Read Update Delete
EDP	Enterprise Design Pattern
eMI	Enterprise Messaging Infrastructure
ESB	Enterprise Service Bus
ESS	Enterprise Shared Services
ETA	Enterprise Technical Architecture
ETSP	Enterprise Technology Strategic Plan
HTTPS	Hypertext Transfer Protocol Secure
IAM	Identity and Access Management
IT	Information Technology
LOB	Line of Business
OI&T	Office of Information and Technology
OMB	Office of Management and Budget
REST	Representational State Transfer
SDE	Service Delivery and Engineering

Acronym	Description
SOA	Service Oriented Architecture

APPENDIX E. REFERENCES, STANDARDS, AND POLICIES

This Enterprise Design Pattern is aligned to the following VA OI&T references and standards, applicable to all new applications being developed in the VA, and are aligned to the VA ETA:

#	Issuing Agency	Applicable Reference/Standard	Purpose
1	VA	VA Directive 6551	Establishes a mandatory policy for establishing and utilizing Enterprise Design Patterns by all Department of Veterans Affairs (VA) projects that develop information technology (IT) systems in accordance with the VA's Office of Information and Technology (OI&T) integrated development and release management process, the Veteran-focused Integration Process (VIP)
2	VA OIS	VA 6500 Handbook	Directive from the OI&T OIS for establishment of an information security program in the VA, which applies to all applications that leverage ESS http://www1.va.gov/vapubs/
3	VA ASD	VA Enterprise Design Patterns, Office of Technology Strategies	Provides references to the use of enterprise capabilities as part of the integration with SOA support infrastructure services; these documents are intended to standardize and constrain the solution architecture of all healthcare applications in the VA http://www.techstrategies.oit.va.gov/docs_design_patterns.asp
4	VA ASD	ESS Strategy Document and Directive	Provides the overarching strategy for developing, deploying, and managing ESS throughout the VA; ESS guidelines for Message Exchange provide the consensus set of standards for interoperable messaging http://vaww.ea.oit.va.gov/enterprise-shared-services-service-oriented-architecture/

#	Issuing Agency	Applicable Reference/Standard	Purpose
5	NIST SP 800-95	Guide to Secure Web Services	Provides standards and guidelines to deliver adequate information security for all agency operations and assets http://csrc.nist.gov/publications/nistpubs/800-95/SP800-95.pdf
6	NIST SP 800-21-1	Guideline for Implementing Cryptography in the Federal Government	Provides a set of guidelines for selecting, specifying, employing, and evaluating cryptographic protection mechanisms in Federal information systems http://csrc.nist.gov/publications/nistpubs/800-21-1/sp800-21-1_Dec2005.pdf
7	NIST SP 800-32	Introduction to Public Key Technology and the Federal PKI Infrastructure	Developed to assist agency decision-makers in determining if a PKI is appropriate for their agency, and how PKI services can be deployed most effectively within a Federal agency http://csrc.nist.gov/publications/nistpubs/800-32/sp800-32.pdf
8	NIST SP 800-57	Recommendation for Key Management	This Recommendation provides cryptographic key management guidance. It consists of three parts. Part 1 provides general guidance and best practices for the management of cryptographic keying material. Part 2 provides guidance on policy and security planning requirements for U.S. government agencies. Part 3 provides guidance when using the cryptographic features of current systems. http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf
9	FIPS 140-2	Security Requirements for Cryptographic	This publication provides a standard that will be used by Federal organizations when these organizations specify that cryptographic-based security systems are

#	Issuing Agency	Applicable Reference/Standard	Purpose
		Modules	to be used to provide protection for sensitive or valuable data. http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf
10	FIPS 186-2	Digital Signature Standards	This standard specifies a suite of algorithms which can be used to generate a digital signature. Digital signatures are used to detect unauthorized modifications to data and to authenticate the identity of the signatory. http://csrc.nist.gov/publications/fips/archive/fips186-2/fips186-2.pdf
11	VA	WS-Security solutions	http://trm.oit.va.gov/StandardPage.asp?tid=5146&tab=2
12	VA	API Gateways	http://vaww.oed.portal.va.gov/projects/bgs/asa/Wiki%20Pages/Functionalities%20provided%20by%20XML%20Gateway%20v.1.aspx
13	VA	eMI	http://go.va.gov/emi
14	VA	eMI Integration Guidance	The eMI Integration guidance is intended for individuals or organizations seeking to utilize eMI services by requesting, onboarding, and consuming eMI services. The guide also provides set of guidelines and recommendations to develop and use Services and other messaging solutions within SOA framework. Integrating with the eMI aids in the assurance of compliance with the VA Technical Reference Model (TRM) as the eMI leverages only approved technologies outlined in the TRM. http://vaww.oed.portal.va.gov/communities/VAeMI/e

#	Issuing Agency	Applicable Reference/Standard	Purpose
			MI%20Documents/eMI%20Integration%20Guideline%20Overview.pdf
15	VA	Full range of technologies provided by the TRM	http://www.va.gov/TRM/ReportVACategoryMapping.asp
16	VA	Approved Enterprise Design Patterns	http://www.techstrategies.oit.va.gov/docs_design_patterns.asp
17	OMB	OMB M-15-13	Policy to require secure connections across Federal websites and web services https://www.whitehouse.gov/sites/default/files/omb/memoranda/2015/m-15-13.pdf