

Intel® Cloud Builders Guide: Cloud Design and Deployment on Intel® Platforms

Cloud Gateway Security with Intel® SOA Expressway

AUDIENCE AND PURPOSE

Cloud computing offers a path to greater scalability and lower costs for service providers, infrastructure hosting companies, and large enterprises. Establishing an infrastructure that can provide such capabilities requires experience. Intel has teamed up with leading cloud vendors through the Intel® Cloud Builders program to help any customer design, deploy, and manage a cloud infrastructure.

Data center operators, solution architects, application users and architects, and security architects are usually responsible for implementing and maintaining the appropriate security model for a particular enterprise, regardless of how the enterprise exposes itself outside the DMZ. For enterprise IT, cloud services pose unique security challenges compared to traditional access security models.

The traditional security model, also known as the single domain security model, focuses on privileged data user access, trusted and anonymous user access, and application access control for data. In the private, public, and hybrid cloud models, however, security requirements evolve significantly. Early solutions for establishing IaaS connectivity have

centered on extending the enterprise network perimeter to encompass the cloud services. This model, normally based on virtual LAN technology, allows for easy bi-directional network access between the established enterprise domain and IaaS-type domains. The basic advantage of this model lies in its simplicity; it's built using well-understood technology from multiple sources. Plus, it is transparent to higher layers of the open systems interconnection (OSI) network stack, making application integration over the network boundary relatively easy—latency and reliability concerns aside.

However, the single domain security model has significant security vulnerabilities. The extended network pattern is essentially one virtual security domain that covers both on- and off-premise resources. The enterprise has substantially increased the attack surface of its perimeter and has created a weak link by giving remote and third-party managed resources unlimited access into the primary network.

The authorization domains model, or secure access model, is an efficient alternative that enforces information security in the cloud environment while addressing the shortcomings in the single domain

security model. Authorization domains support independent security domains that cooperate to achieve integration while enforcing a consistent security policy. Here, on- and off-premise applications are deliberately isolated from each other, so that distribution is explicit, even if the exact locations are not. The connecting components are service gateways rather than the switches of the VLAN approaches. These gateways are usually based on the technologies of SOA.

The Intel Cloud Builders program provides a starting point by supplying a basic hardware blueprint and available cloud software management solutions, such as Intel® SOA Expressway. The use cases described in this reference architecture can be used as a baseline to build more complex usage and deployment models to suit specific customer needs.

The audience for this reference architecture is cloud service providers, cloud hosters, and enterprise IT that want to realize the revenue potential of their existing data center infrastructure by offering cloud computing services to their customers or internal users.

Table of Contents

Executive Summary	3
Introduction	3
Intel® SOA Expressway Product Implementation Overview.	4
Service Gateway	4
Security	4
Governance	5
Performance	5
Testbed Blueprint Overview	6
Hardware Description	6
Technical Review	6
Installation Overview	6
Use Case: Authenticating Credentials for a Cloud Operator	6
Intel SOA Expressway Solution	6
Explaining Intel SOA Expressway Workflow Logic	7
Use Case: Policy Enforcement of Single Sign-On Access to a Cloud	11
Intel SOA Expressway Solution	11
Explaining Intel SOA Expressway Workflow Logic	12
Use Case: Secure Credential Federation for a Hybrid Cloud Environment	17
Intel SOA Expressway Solution	17
Configuring Cloud Federation	17
A Detailed Look at the Federation Process	18
Use Case: Two-factor Authentication for Ensuring Client Credentials Are Valid.	19
Intel SOA Expressway Solution	19
Configuring Two-Factor Authentication	19
A Detailed Look at the Authentication Process	20
Execution and Results.	22
Next Steps	23
Things to Consider	23
Glossary	23
References	24

Executive Summary

The emerging use of public, private, and hybrid cloud paradigms has driven a focus on the security of both infrastructure and virtual machines as well as on applications and identities on the cloud.

Traditional access security models have focused on privileged data user access; trusted and anonymous user access; and application access control for data, applications, and networks. In the private, public, and hybrid cloud models, however, security requirements evolve significantly. The authorization domains model is an efficient way of enforcing information security in the cloud environment. Authorization domains support independent security domains that cooperate to achieve integration while enforcing consistent security policy. Here, on- and off-premise applications are deliberately isolated from each other, so that distribution is explicit, even if the exact locations are not. The connecting component is Intel® SOA Expressway—a service, security, and cloud gateway that provides a solution for identity federation, security policy enforcement for identities and resources that span multiple security domains, and threat protection and trust functions for application-level network traffic between a client and cloud services.

Intel's cloud implementation consists of a six-node cluster. Each machine is a Dell PowerEdge® 2950 powered by Quad-Core Intel® Xeon® processors. Each machine has 100 GB of memory and 8 GB of RAM. Intel SOA Expressway is exposed as a gateway on the cloud perimeter and acts as a transparent proxy between the client and the cloud service. Intel SOA Expressway removes concerns over fixed-capacity fragile virtual networks by carrying secured messages across multiple security domains. Intel SOA Expressway passes messages directly, service to service, removing integration obstacles by doing inter-domain communications for key shared concerns, and providing a simple, central location for application development and deployment.

The most important considerations for this model are service authorization, policy delegation, and security reporting. By residing on the cloud perimeter, Intel SOA Expressway provides remote domain authorization, since existing network security models can no longer be relied on. In addition, the Intel SOA Expressway runtime provides strong authorization control and auditing for any message regardless of the security domain, thereby addressing the inherent weakening perimeter controls in a cloud and protecting against internal and external abuse. Intel SOA Expressway supports a robust reporting infrastructure that centralizes the collection and analysis of policy violations via auditing and reporting mechanisms.

Introduction

Traditional access security models have focused on privileged data user access; trusted and anonymous user access; and application access control for data, applications, and networks. In the private, public, and hybrid cloud models, however, security requirements evolve significantly.

Early solutions for establishing IaaS connectivity have centered on extending the enterprise network perimeter to encompass the cloud services. This model, normally based on virtual LAN technology, allows for easy bi-directional network access between the established enterprise domain and IaaS-type domains. The basic advantage of this model lies in its simplicity; it's built using well-understood technology available from multiple sources. Plus, it is transparent to higher layers of the OSI network stack, making application integration over the network boundary relatively easy—latency and reliability concerns aside.

This extended network pattern is essentially one virtual security domain that covers both the on- and off-premise resources of an enterprise. This domain is different in three key respects from

an on-premise only domain:

- It encompasses third-party network traffic in the off-premise partition
- It critically depends on the security of a network tunnel
- It is substantially larger than the original domain

The enterprise has substantially increased the attack surface of its perimeter and has created a weak link by giving remote and third-party managed resources unlimited access into the primary network. It is little wonder that security continues to be the main concern of enterprises that are deploying cloud services.

The virtual LAN model also creates fixed-size chokepoints between cloud instances, be they private to public, private to partner, or any combination of these. Each of these pairings requires capacity planning and monitoring to ensure they present sufficient bandwidth, support low-latency transactions, and meet uptime requirements. Beyond the security concerns, this exponential operational overhead leads to a fragile base infrastructure on which to host service-to-service interactions.

The following are typical requirements in cloud computing. Intel SOA Expressway is a solution for all of them:

- Consistent, secure credential validation for cloud management access
- Consistent, secure policy enforcement for a combination public and private cloud environment
- Security policy enforcement for application access that can be hosted on a private or public cloud
- Prevent rogue access to the cloud data and applications

Intel® SOA Expressway Product Implementation Overview

Intel SOA Expressway is a software appliance designed to simplify SOA architecture on-premise or on the cloud. It expedites deployments by addressing common SOA bottlenecks—it accelerates, secures, integrates, and routes XML, web services, and legacy data in a single, easy-to-manage software-appliance form factor.

Intel SOA Expressway offers a complete “Cloud-in-a-Box” solution, specifically designed to meet the requirements of carriers, service providers, and hosting providers who want to offer revenue-generating, infrastructure-on-demand cloud computing services to their customers quickly and easily, with a compelling and differentiated feature set.

Intel SOA Expressway features include:

- A fully multi-tenant, carrier-class cloud service platform
- Detailed resource metering and accounting
- Using clustering and load balancing, the Intel SOA Expressway runtime can scale up as well as scale out
- Software service router provides security, governance, and mediation
- Standard operating system support: Linux* OS, Windows* OS, and Solaris*
- Optimized for Intel® Multi-Core processors, Intel SOA Expressway scales directly on standard Intel-based servers
- Performance: Scalable service mediation engine and wire-speed XML acceleration
- Service mediation: Supports both simple proxy and complex mediation use cases
- Service governance: Message throttling, QoS, and standardized logs and policies

- Security features include security proxy, services firewall, AAA, TLS, and trust mediation. Provides software-based core XML IP and best-in-class threat protection and extensible trust features
- Software appliance: Appliance manageability with software extensibility
- Extensibility: custom business rules, service hosting, and data and messaging adapters
- Build dynamic applications: Innovate faster, gain a competitive edge, and securely expose mass customizable applications on-premise or on the cloud, regardless of the abstraction pattern (SOA, WOA), delivery method (app store, cloud), or protocol (REST, SOAP)
- A centralized policy enforcement point to authenticate, authorize, and govern service interactions with customers, partners, and employees as they consume or deploy applications

Service Gateway

Real-world SOA tends to grow organically. Businesses can have divisions with local SOA implementations that include registry, repository, service inventory, ESB, portals, and security infrastructure all based on the needs and trends of that local domain. This situation arises because different domains have made significant investments in proprietary, integrated SOA technologies from major software vendors. While the integrated stack provides SOA for the local domain, it creates middleware heterogeneity and simply repeats the silo problem at the next level.

The transition to cloud environments presents a catalyst for the enterprise to put in place a solution that works across internal and external domains. Intel SOA Expressway is a true cross-domain service gateway that bridges these islands of integration, no matter the protocol or deployment pattern.

Intel SOA Expressway is a highly scalable software product that provides this service gateway functionality—combining common functions of a service bus, security gateway, and XML acceleration engine into a single product that scales on next-generation Intel® Multi-Core processor-based servers for the modern virtualized data center (see Figure 1 on the next page).

Security

One of the biggest challenges to widespread cloud-based services is runtime governance and web services security. Intel SOA Expressway provides a common interface for REST- and SOAP-based communication, service virtualization, message-level security, delegated AAA functions, and threat prevention. Intel SOA Expressway also provides perimeter defense against content attacks. Moreover, Intel SOA Expressway is a software appliance form-factor that avoids custom, hard-to-manage, proprietary XML hardware. It runs on secure, open operating systems, avoiding the “security by obscurity” problem of “hardened” hardware appliances.

Governance

Intel SOA Expressway reduces the management, development, and capital costs of large distributed applications and can secure, transform, route, and mediate between services offered by any cloud vendor, whether they use a silo-, legacy-, or standards-based communication mechanism. Furthermore, it uses a code-less Eclipse*-based designer that supports simple or complex mediation applications.

Intel SOA Expressway also supports integrated governance for SOA by executing and enforcing governance policies of arbitrary complexity at runtime. This makes it easy to integrate Intel SOA Expressway with existing SOA repository or policy design environments as well as report monitoring information to web services management and monitoring solutions.

Performance

Intel SOA Expressway provides an optimized service mediation engine to orchestrate workflow processing. Intel SOA Expressway provides a single runtime instance that offers XML and service mediation acceleration that scales with any Multi-Core Intel Xeon processor-based server, regularly beating custom hardware appliances by a factor of 8 to 2 or greater.¹ Intel SOA Expressway instantly brings the power of Intel Multi-Core processors and Moore's Law to XML-rich business applications without requiring any special programming or custom hardware.

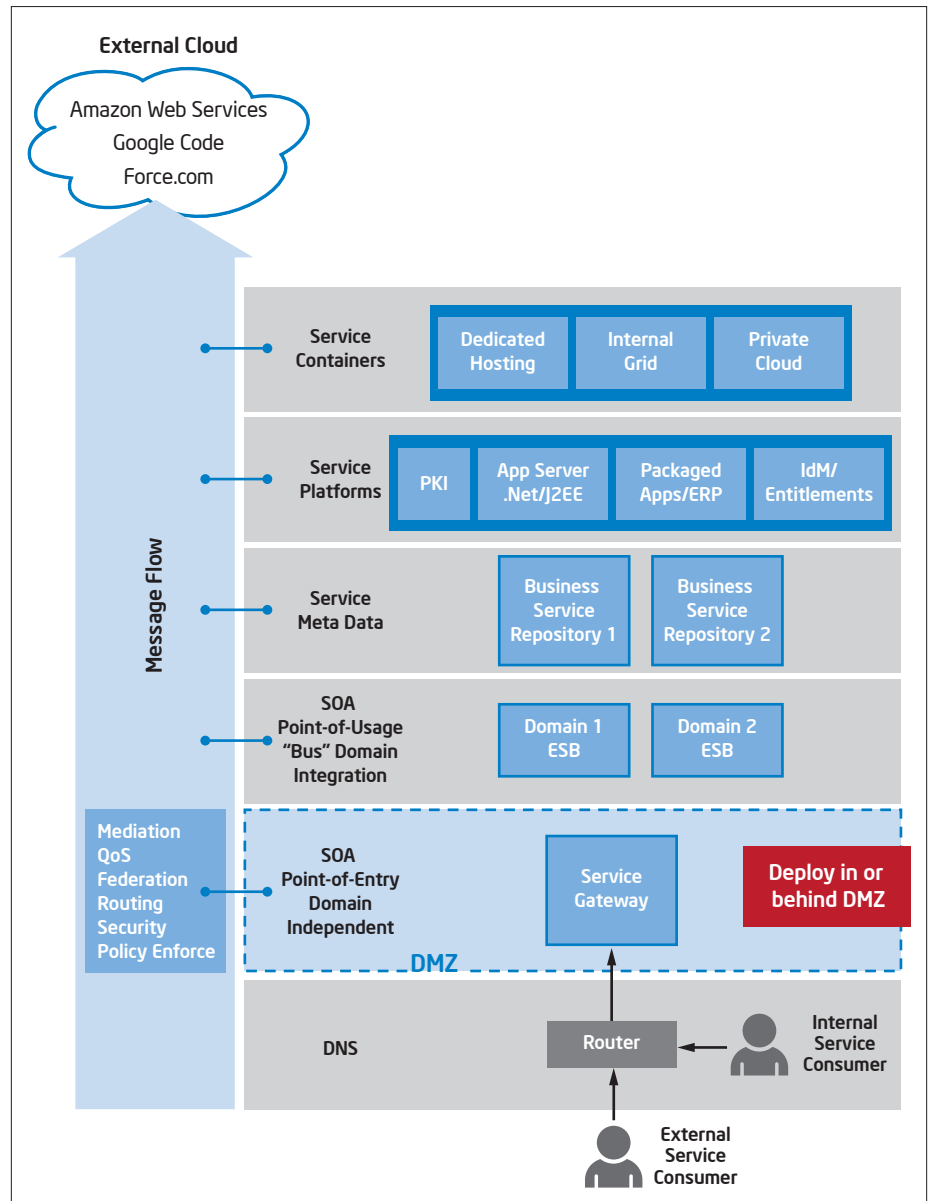


Figure 1. Service Gateway Protecting Cloud Services

Testbed Blueprint Overview

Hardware Description

In this reference architecture, the cloud implementation consists of a six-node cluster. Each machine is a Dell PowerEdge 2950 (powered by Quad-Core Intel Xeon processors). Each machine has 100 GB of memory and 8 GB of RAM.

Technical Review

Installation Overview

To install Intel SOA Expressway, perform the following steps:

1. Obtain the Intel SOA Expressway RPM by contacting intelsoaeinfo@intel.com. The RPM Package Manager is a Linux package management system. The Intel SOA Expressway RPM is software packaged in the .rpm file format.
2. Copy the Intel SOA Expressway RPM into a directory on the target system. Use SCP (secure copy) or FTP to do this.
3. Ensure that you have root privileges to do the RPM install. For security reasons, it is recommended that you install Intel SOA Expressway running as non-root.
4. Execute the following command to install Intel SOA Expressway: **rpm -i [soae rpm]**, where [soae rpm] is the absolute file path to the RPM.
5. Start the post-installation process by executing the command: **cli postinstall**.
6. Start Intel SOA Expressway by executing the command: **cli serviceStart**.

Use Case: Authenticating Credentials for a Cloud Operator

The emerging cloud paradigm poses significant security problems for establishing client identity, trust, and access control to particular resources on the cloud. Typically, a server is placed in a cloud which can be accessed by a client via an Internet connection. Generally, the server will not know the identity of this client nor be able to authenticate the client's identity and authorize access to server resources. In addition, because communication occurs over the Internet, the risk of attack by malicious users is greatly increased. This makes establishing trust, identity, and user access even more important than the type of security found in a corporate intranet.

These cloud-based access and identity issues are demonstrated in the following use case: A service that creates and manages virtual machines is deployed to the cloud. Any client can remotely access and manage the virtualization service by sending and receiving REST calls to and from the back end. For authentication, the client provides a username and password in an XML document to the back end. However, the server has neither the capability to authenticate the client nor a mechanism to determine appropriate access rights, such as whether the client can create a virtualization instance.

Intel SOA Expressway Solution

Intel SOA Expressway is a service gateway that can do the following:

- Enforce security policies across security domains
- Delegate authentication and authorization to a local credential directory
- Log authentication failures for auditing
- Proxy messages between two endpoints. The message could be binary, XML, REST, or SOAP

In the service virtualization use case, Intel SOA Expressway can be placed on the cloud perimeter between the server and the client. From the perimeter, Intel SOA Expressway can transparently proxy messages between the endpoints and delegate authentication and access control to a third-party entity, such as LDAP or Oracle Access Manager.* If the delegated authenticator validates the client's credentials, then Intel SOA Expressway generates valid login credentials that are hidden from the client and submits those to the back end. The back end in turn generates a HTTP session cookie, which Intel SOA Expressway forwards to the client. After this exchange, the client is permitted to submit commands to the server as long as each command has the session cookie in it. However, if authentication fails, then Intel SOA Expressway logs the failure and blocks requests from the client until authentication succeeds or the client violates DoS controls. In addition, if the session cookie is missing then authentication and authorization must be repeated.

Explaining Intel SOA Expressway Workflow Logic

The following procedure explains how to generate the workflow logic that Intel SOA Expressway uses to mediate messages between the client and the virtualization service. This procedure is performed in Intel® Services Designer, the Eclipse IDE for creating Intel SOA Expressway applications.

1. Create a project. In the project, create an empty workflow and an AAA policy.
2. The AAA policy will extract the username and password from the client's request and forward it to a third party for authentication and authorization. In the AAA policy, perform the following steps:
 - a. Select the Identity Management tab.
 - b. In the **Identity Source** drop-down menu, select **Username** from the workflow.
 - c. Select the **Authenticate Identity** check box.
 - d. In the **Authenticate using** drop-down menu, select one of the following entities to send the login credentials to: Tivoli® Access Manager, Oracle Access Manager, LDAP, CA SiteMinder®, or Oracle Entitlements Server.
 - e. Select the Resource Authorization tab.
 - f. In the Resource Authorization tab, select the **Authorize Resource** check box. The configuration options for setting up resource authorization will display.
 - g. Configure the resource authorization so that the user is validated against access policies such as duration, request type, and time of day.
3. In the workflow, insert a **Receive** action from the **Palette** menu. This action receives messages from the client.
4. In the Receive action's **Properties** view, perform the following steps:

- a. In the **Request data** and **Response data** structure areas, select the **XML** radio buttons.
- b. In the **Service URL** field, enter **https://localhost/sdk/vimService**. This means that the service located at the /sdk/vimService URL is protected, and clients can only access that service through Intel SOA Expressway.

As a result, the Receive action should look like this:

5. In the workflow, insert an **If** action from the **Palette** menu and rename the action to **IfMessageBody**. This action determines whether the message request has an XML document in it.

6. In the IfMessageBody action's **Properties** view, enter the following XPath expression in the field: **soae-xf:get-message-size(\$Receive.body) > 0**. Then, select the **Add Else** button.

The get-message-size extension function retrieves the message payload's size. Then, the XPath expression determines if the message size is greater than zero. If it is, then the expression evaluates to true; if not, then it evaluates to false. When the If action evaluates to true, the message will be sent to a branch with actions that authenticate the document's credentials. When the If action evaluates to false, then the message will be sent to a branch with actions that forward the REST call to the virtualization service.

Figure 2. Receive Action

Activity	Expression
	soae-xf:get-message-size(\$Receive.body) > 0
	<otherwise>

Figure 3. IfMessageBody Action

- 7. In the **IfMessageBody** branch, add a sequence from the **Palette** menu and rename it **LoginVM**. In the **Else** branch, add a sequence from the **Palette** menu and rename it **VMCommand**.
- 8. In the **LoginVM** sequence, insert an **Expression** action from the **Palette** menu. Rename the action **MessageBody**. In the **MessageBody** action's **Properties** view, enter `soae-xf:get-primary-xml-document($Receive.body)` in the **Expression** field. As a result, the extension function retrieves the XML document from the **Receive** action and places the document in the **MessageBody** variable.
- 9. In the **LoginVM** sequence, insert an **AAA** action from the **Palette** menu. Rename the action to **AuthenticateUser**.

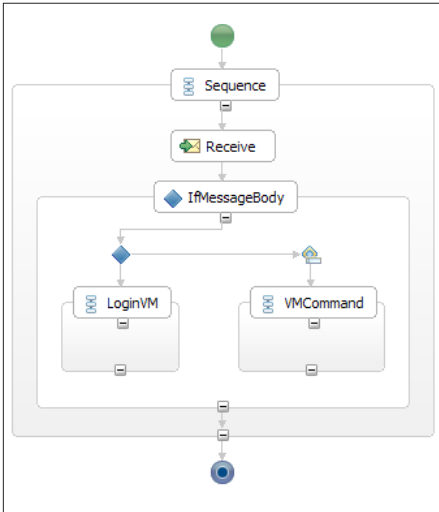


Figure 4. Workflow skeleton

- 10. In the **AuthenticateUser** action's **Properties** view, perform the following steps:
 - a. In the **Message** drop-down menu, select **Receive**.
 - b. Select the **Browse** button. In the **Browse for AAA policy** dialog box, select the AAA policy that authenticates the login credentials and then click the **OK** button.
 - c. In the **AAA Parameters** table, locate the field next to **Source Identity: Username**. In that field, enter the following XPath expression: `$MessageBody//viminternal:userName/text()`.
 - d. In the **AAA Parameters** table, locate the field next to **Source Identity: Password**. In that field, enter the following XPath expression: `$MessageBody//viminternal:password/text()`.

- As a result, the AAA action selects the username and password from the **MessageBody** variable and then sends them to the AAA policy. The AAA policy sends the credentials to a third party for authentication and authorization.
- 11. In the **LoginVM** sequence, insert a scope from the **Palette** menu. Then, place the **AuthenticateUser** action in the scope.
 - 12. Right-click the scope that contains the **AuthenticateUser** action and then select **Add Fault Handler** from the content menu. As a result, if authentication failure is generated from the AAA action, then it is captured by the fault handler.
 - 13. In the fault handler, delete the **Rethrow** action. Then, from the **Palette** menu, add the **Transaction Log** and **Exit** actions to the fault handler.

Name:

☐ Override destination:

Message:

Receive

Policy:

Policy.aaa

Browse...

Update policy

Create new policy

Policy Description

Identity processing:

Extract identity from **data within workflow** (user name and password)

Authenticate identity with **keystore query**

AAA Parameters:

Name	Value
Source Identity: Username	<Value not set. Click here to edit the value...>
Source Identity: Password	<Value not set. Click here to edit the value...>

Figure 5. AAA Action

14. The Transaction Log action records data in the Intel SOA Expressway transaction logs. To log authentication failures in the transaction log, perform the following steps in the Transaction Log action's **Properties** view.

- In the **Comment** field, enter the following string Authentication or authorization of the client failed.
- In the **Optional Data** drop-down menu, select **\$AuthenticateUser**.

As a result, the authentication failure from the AuthenticateUser action and the string in the Comment field are recorded to the Intel SOA Expressway transaction logs.

15. If authentication is successful, then the workflow must generate a valid username that is accepted by the virtualization service. To do this, perform the following steps:

- In the LoginVM sequence and after the scope that contains the AuthenticateUser action, insert an **Expression** action. Rename the action to **Username**.
- In the Username action's **Properties** view, enter string ('username') in the **Expression** field.
- Select the **Override destination** check box.
- In the **Override destination** field, select the **X**.
- In the Build XML data query dialog box, enter **\$MessageBody//viminternal:userName/text()** and then select the **OK** button.

As a result, this action extracts the username from the incoming request and inserts it as a string to the "userName" node in MessageBody element.

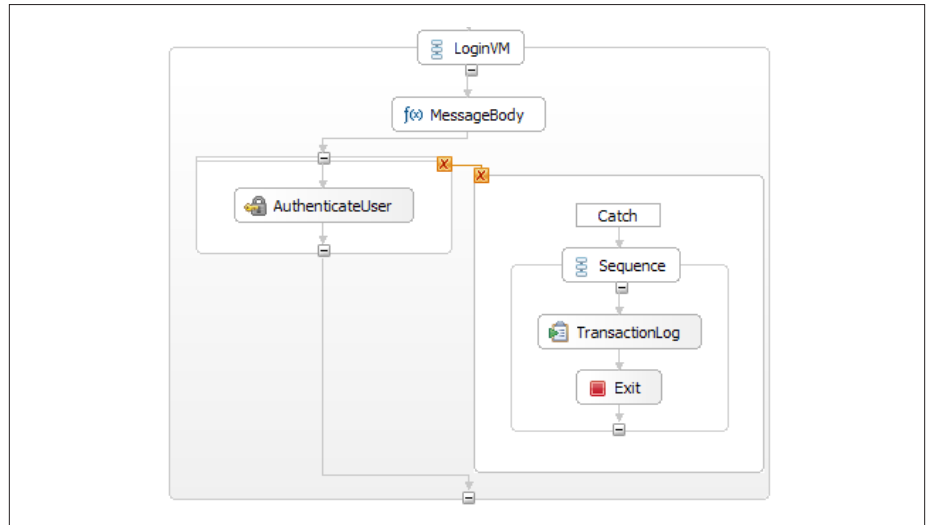


Figure 6. Workflow

Name: ☐ Override destination:

Comment: Log Level:

Optional data:

Figure 7. TransactionLog Action

Name: ☒ Override destination:

Expression: Output type:

☐ Keep source element name

Figure 8. Expression Action

16. If authentication is successful, then the workflow must generate a valid password that is associated with the username. To do this, perform the following steps:
 - a. In the LoginVM sequence and after the scope that contains the AuthenticateUser action, insert an Expression action. Rename the action to **Password**.
 - b. In the Password action's **Properties** view, enter string ('password') in the **Expression** field.
 - c. Select the **Override destination** check box.
 - d. In the **Override destination** field, select the **X**.
 - e. In the Build XML data query dialog box, enter **\$MessageBody//viminternal:password/text()** and then select the **OK** button.

As a result, the Password action generates a password and then places that string into the password node of the MessageBody variable's XML document. This is the XML document sent by the client. The password is never seen by the client and is only known to Intel SOA Expressway and the back end.

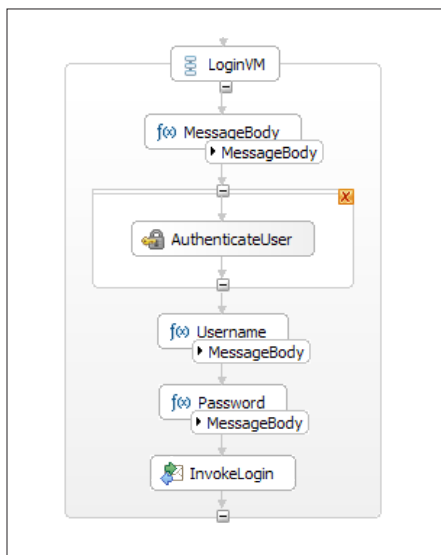


Figure 9. Login Workflow

17. In the LoginVM sequence, insert an **Invoke** action after the Password action. Rename the action **InvokeLogin**. Then, in the InvokeLogin action's **Properties** view, perform the following steps:
 - a. In the **Message data from** drop-down menu, select **\$MessageBody**.
 - b. In the **Service URL** field, enter the URL of the service virtualization server.
18. In LoginVM sequence, insert a **Reply** action after the InvokeLogin action.

As a result, the XML document with valid login credentials is sent to the back-end server.

19. In the **VMCommand** sequence, insert **Invoke** and **Reply** actions.
20. In the Invoke action, perform the following steps:
 - a. In the Message data from drop-down menu, select **\$Receive.body**.
 - b. In the Service URL field, enter the URL of the back-end server.
21. In the Reply action's Properties view, select **\$Invoke.body**. As a result, the message responses sent by the back-end server are forwarded to the client.

As a result, commands are sent by the client to the virtualization service.

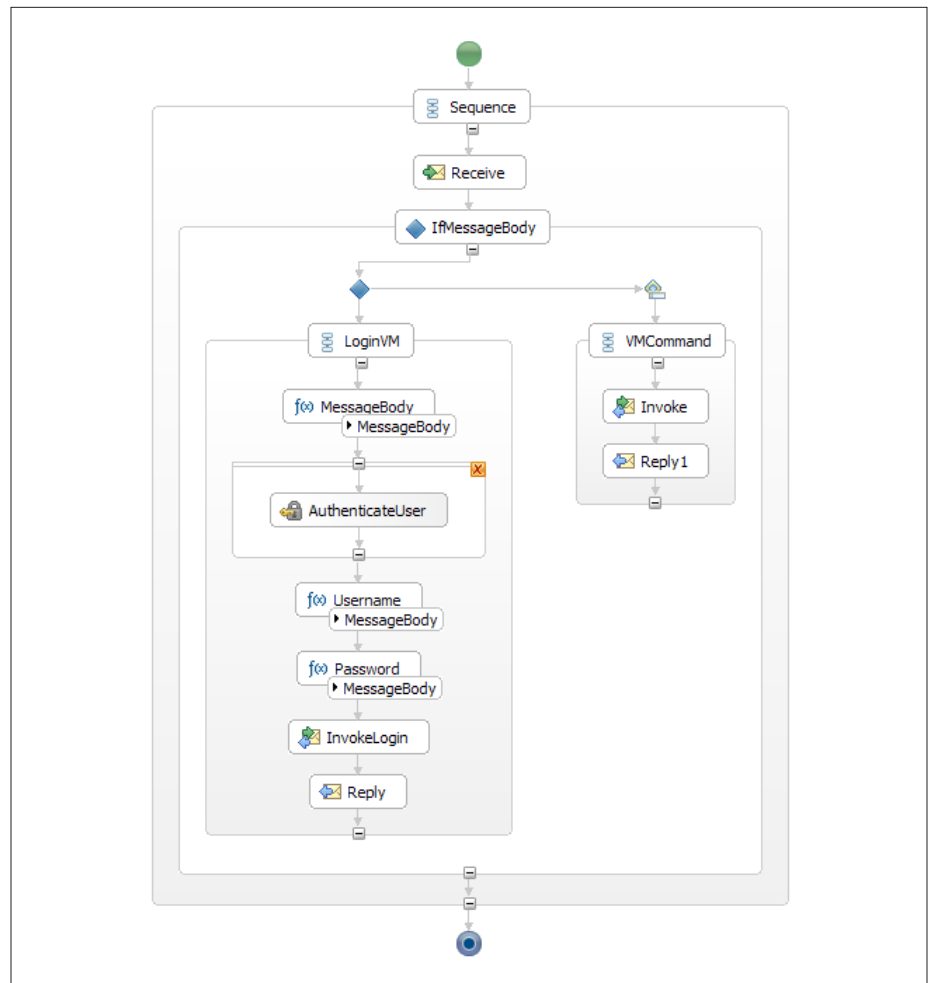


Figure 10. Workflow Snapshot

Use Case: Policy Enforcement of Single Sign-On Access to a Cloud

The emerging cloud paradigm poses significant problems for single sign-on access. Enterprises usually place a server on a cloud, which can be accessed by a client via an Internet connection. In Internet communications, a client usually resides in a unique security domain and uses a unique mechanism for authentication and authorization. Frequently, the client's security token or sign-on credential cannot be understood or consumed by the service on the cloud.

This single sign-on access issue is demonstrated in the following use case: Two companies have independently contracted with a third party to handle their payroll services. Each company is in its own security domain. In addition, each company has its own unique payroll system and a specific method for handling authentication and authorization to the payroll system—one company uses Oracle Access Manager while the other uses CA SiteMinder.

In the past, the vendor set up two independent solutions to integrate with each company's payroll system. Recently, however, the vendor decided to deploy a single payroll solution to the cloud that handles both the companies' payroll needs in a centralized location. To ensure security, the vendor must develop a mechanism that can transparently consume authentication and authorization information from CA SiteMinder and Oracle Access Manager and consistently enforce appropriate security policies.

Intel SOA Expressway Solution

Intel SOA Expressway is a service gateway that can do the following:

- Enforce security policies across security domains.
- Delegate authentication and authorization to a local credential directory.
- Log authentication failures for auditing.
- Proxy messages between two endpoints. The message could be binary, XML, REST, or SOAP.

In the single sign-on use case, Intel SOA Expressway can be placed on the cloud perimeter between the payroll service and the clients. From the perimeter, Intel SOA Expressway can transparently delegate authentication to CA SiteMinder and Oracle Access Manager, map authenticated information to a SAML assertion, and then forward the SAML assertion to the payroll system. Since the vendor's payroll system can consume SAML assertions, Intel SOA Expressway acts as a mediator that translates the clients' authentication information into a format that the payroll system can handle. However, if authentication fails, then Intel SOA Expressway logs the failure and blocks requests from the client until authentication succeeds or the client violates DoS controls.

To interact with the payroll system, the vendor imposed three requirements on the companies: First, the client must indicate in an HTTP header what authentication mechanism is being used. The header name must be AUTH, and the header value must either be OAM or CA SiteMinder. Second, the username and password must be accessible from the HTTP headers. Third, the message request must be a SOAP document.

Explaining Intel SOA Expressway Workflow Logic

The following procedure explains how to generate the workflow logic that Intel SOA Expressway uses to mediate messages between the clients and the vendor's payroll solution. This procedure is performed in Intel Services Designer, the Eclipse IDE for creating Intel SOA Expressway applications.

1. Create an Intel SOA Expressway project.
2. In the project, create two AAA policies. Name the first AAA policy **OracleAM**. Name the second AAA policy **CaSM**.
3. Open the OracleAM policy.
4. In the OracleAM policy, perform the following steps:
 - a. Select the Identity Management tab.
 - b. In the Identity Source drop-down menu, select Username from HTTP Basic Authentication header.
 - c. Select the **Authenticate Identity** check box.
 - d. In the Authenticate using drop-down menu, select Oracle Access Manager.
 - e. In the Oracle Access Manager details area, select the **Return Single Sign-On token** check box. Then, configure the authentication options based on your needs.
 - f. Select the **Map Identity** check box.
 - g. In the **Map identity to** drop-down menu, select **SAML Assertion**.
 - h. In the **Issuer** field, enter **SOAE**.

As a result, the AAA policy will extract the username and password from the HTTP headers, send that data to Oracle Access Manager to be authenticated, and then map the authentication response to a SAML assertion. The SAML assertion will contain a single sign-on token.

Figure 11. AAA Policy - Extract and Authenticate Identity

Figure 12. AAA Policy - Map Identity to SAML

5. In the CaSM policy, perform the following steps:
 - a. Select the Identity Management tab.
 - b. In the Identity Source drop-down menu, select Username from HTTP Basic Authentication header.
 - c. Select the **Authenticate Identity** check box.
 - d. In the **Authenticate using** drop-down menu, select **CA SiteMinder**.
 - e. In the CA SiteMinder details area, select the **Return Single Sign-On token** check box. Then, configure the authentication options based on your needs.
 - f. Select the **Map Identity** check box.
 - g. In the **Map identity to** drop-down menu, select **SAML Assertion**.
 - h. In the **Issuer** field, enter **SOAE**.

As a result, the AAA policy will extract the username and password from the HTTP headers, send that data to CA SiteMinder to be authenticated, and then map the authentication response to a SAML assertion. The SAML assertion will contain a single sign-on token.

6. In the project, create an empty workflow.
7. Insert a **Receive** action into the workflow from the **Palette** menu. This action receives message requests sent by a client.
8. In the Receive action's **Properties** view, perform the following steps:
 - a. In the **Endpoint binding type** drop-down menu, select **SOAP**.
 - b. In the **Request data** and **Response data** structure areas, specify global elements for the SOAP message request and response.
 - c. In the **Service URL** field, enter **https://localhost/payrollService**. This means that the service located at the /payrollService URL is protected, and clients can only access that service through Intel SOA Expressway.

9. In the workflow, place a **GetMessageMetadata** action after the Receive action.
10. In the GetMessageMetadata action's **Properties** view, select **Receive** from the **Message** drop-down menu. As a result, the action retrieves HTTP metadata from the message request and allows other actions in the workflow to use that metadata.
11. In the workflow, place an **If** action after the GetMessageMetadata action.
12. In the If action's **Properties** view, perform the following steps:

- a. In the Expression field, enter the following XPath expression: `$GetMessageMetadata/md:transport/md:httpRequest/md:headers/md:header/md:field/md:value/text()='OAM'`
- b. Select the **Add Elseif** button.
- c. In the next Expression field, enter the following XPath expression: `$GetMessageMetadata/md:transport/md:httpRequest/md:headers/md:header/md:field/md:value/text()='CA SiteMinder'`
- d. Select the **Add Else** button.

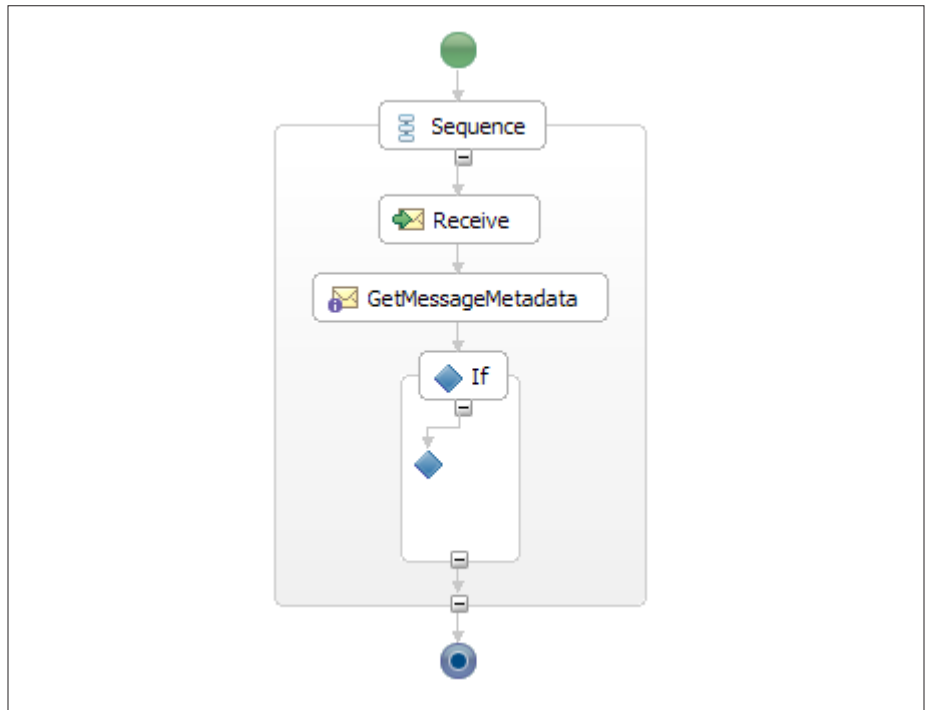


Figure 13. Empty Workflow Skeleton

Conditions:		
Activity	Expression	
<no activity>	<code>\$GetMessageMetadata/md:transport/md:httpRequest/md:headers/md:header/md:field/md:value/text()='OAM'</code>	Add ElseIf
<no activity>	<code>\$GetMessageMetadata/md:transport/md:httpRequest/md:headers/md:header/md:field/md:value/text()='CA SiteMinder'</code>	Add Else
<no activity>	<code><otherwise></code>	Up
		Down
		Delete

Figure 14. If Expressions

As a result, if the message request has a header with the value OAM, then it is routed to one branch; if the message request has a header with the value CA SiteMinder, then the message is routed to another branch. If neither OAM nor CA SiteMinder is found in an HTTP header value, then it is routed to the Else branch.

13. Add a sequence to the first If branch. Name this sequence **OAM**.

14. Add a sequence to the second If branch. Name this sequence **CA SiteMinder**.

15. Add a sequence to the third branch. Name this sequence **PayrollService**.

16. In the OAM sequence, add an **AAA** action. Rename the AAA action to **OAMAuthenticate**.

17. In the OAMAuthenticate action's **Properties** view, perform the following steps:

- In the **Message** drop-down menu, select **Receive**.
- Select the **Browse** button.
- In the **Browse for AAA Policy** dialog box, select **OracleAM** and then select the **OK** button.

As a result, if a message request is sent to the OAM branch, then the OAMAuthenticate action sends the message to the OracleAM AAA policy. The AAA policy, in turn, sends the message to Oracle Access Manager for authentication and then maps the authenticated response to a SAML assertion. The SAML assertion will contain a single sign-on token.

18. In the OAM sequence, insert a scope from the **Palette** menu. Then, place the OAMAuthenticate action in the scope.

19. Right-click the scope that contains the OAMAuthenticate action and then select **Add Fault Handler** from the context menu. As a result, if authentication failure is generated from the AAA action, then it is captured by the fault handler.

20. In the fault handler, delete the **Rethrow** action. Then, from the **Palette** menu, add the **Transaction Log** and **Exit** actions to the fault handler.

21. The Transaction Log action records data in the Intel SOA Expressway transaction logs. To log authentication failures, perform the following steps in the Transaction Log action's **Properties** view.

- In the **Comment** field, enter the following string: Authentication or authorization of the client failed.
- In the **Optional Data** drop-down menu, select **\$OAMAuthenticate**.

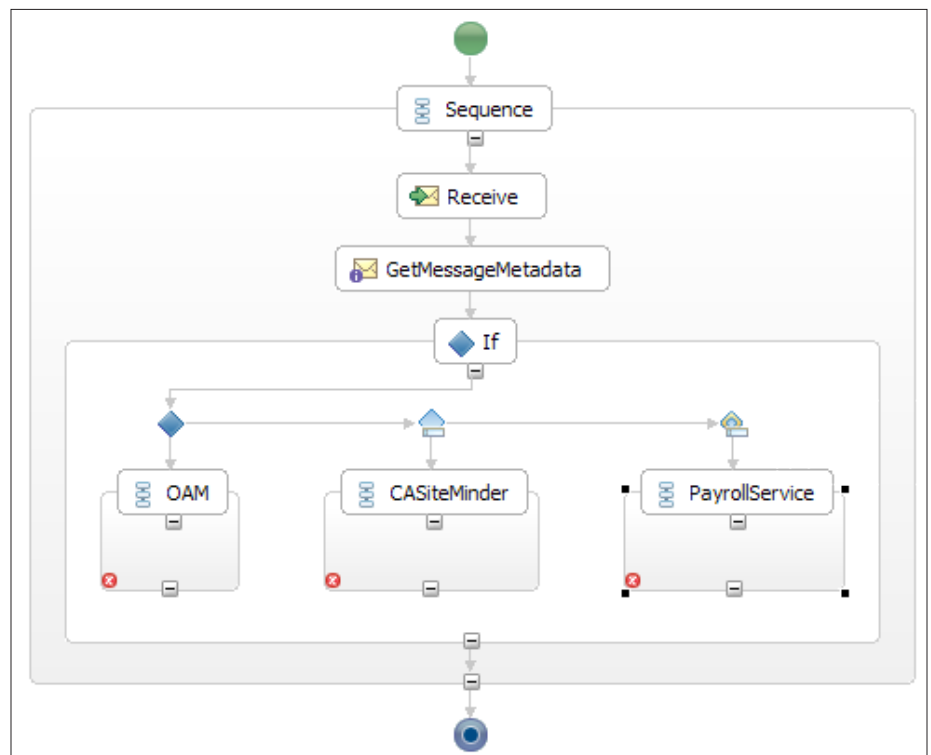


Figure 15. Workflow Skeleton

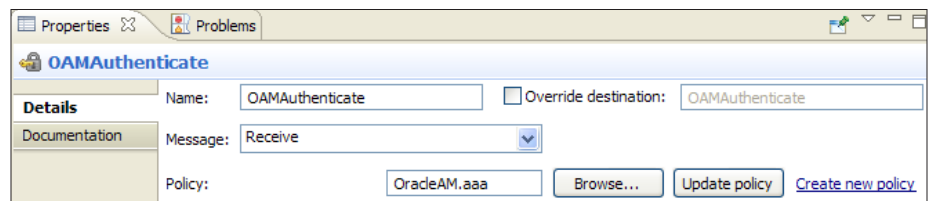


Figure 16. AAA Action for OAM authentication

As a result, the authentication failure from the OAMAuthenticate action and the string in the Comment field are recorded in the Intel SOA Expressway transaction logs.

22. In the OAM sequence, insert an **Invoke** action. Rename the action to **SendSAML_OAM**.
23. In SendSAML_OAM action's **Properties** view, perform the following steps:
 - a. In the Message data from drop-down menu, select \$OAMAuthenticate.
 - b. In the **Service URL** field, enter the hostname and port number of the payroll service.

As a result, the SAML assertion with the OAM single sign-on token is sent to the payroll service. The payroll service will consume the SAML assertion and thus authenticate the user. If the authentication is successful, then the back-end server sends a randomly generated identity token back to Intel SOA Expressway.

24. In the OAM sequence, insert a **Reply** action. In the Reply action, select **\$SendSAML_OAM.body** from the **Message Data from** drop-down menu. As a result, the token generated by the payroll service is sent back to the client. From this point onward, the client must send messages with this token in it. This allows the back-end server to determine whether the client should be trusted or not. The token is valid for the length of the HTTP session, after which client authentication must be performed again.
25. In the CA SiteMinder sequence, add an **AAA** action. Rename the AAA action to **CaSMAuthenticate**.
26. In the CaSMAuthenticate action's **Properties** view, perform the following steps:
 - a. In the **Message** drop-down menu, select **Receive**.
 - b. Select the **Browse** button.

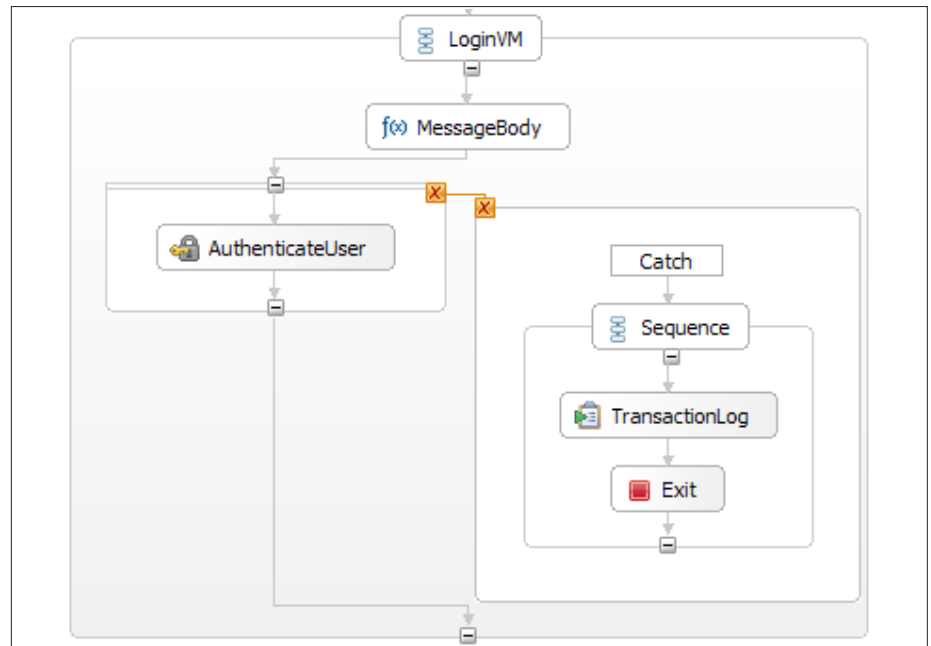


Figure 17. Authentication Error Handling

- c. In the **Browse for AAA Policy** dialog box, select **CaSM** and then select the **OK** button.

As a result, if a message request is sent to the CaSM branch, then the CaSMAuthenticate action sends the message to the CaSM AAA policy. The AAA policy, in turn, sends the message to CA SiteMinder for authentication and then maps the authenticated response to a SAML assertion. The SAML assertion will contain a single sign-on token.

27. In the CA SiteMinder sequence, insert a scope from the **Palette** menu. Then, place the CaSMAuthenticate action in the scope.
28. Right-click the scope that contains the CaSMAuthenticate action and then select **Add Fault Handler** from the context menu. As a result, if authentication failure is generated from the AAA action, then it is captured by the fault handler.

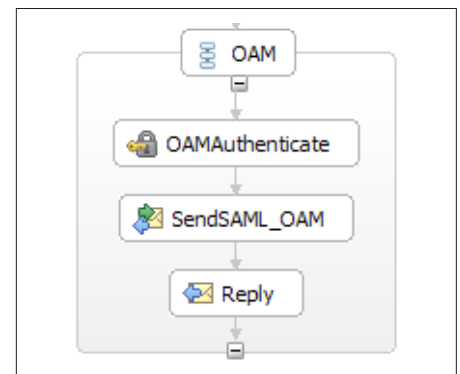


Figure 18. Add remaining steps to the Workflow

29. In the fault handler, delete the **Rethrow** action. Then, from the **Palette** menu, add the **Transaction Log** and **Exit** actions to the fault handler.

30. The Transaction Log action records data in the Intel SOA Expressway transaction logs. To log authentication failures, perform the following steps in the Transaction Log action's **Properties** view.

- In the **Comment** field, enter the following string: Authentication or authorization of the client failed.
- In the **Optional Data** drop-down menu, select **\$CaSMAuthenticate**.

As a result, the authentication failure from the CaSMAuthenticate action and the string in the Comment field are recorded in the Intel SOA Expressway transaction logs.

31. In the CA SiteMinder sequence, insert an **Invoke** action. Rename the action to **SendSAML_CaSM**.

32. In SendSAML_CaSM action's **Properties** view, perform the following steps:

- In the Message data from drop-down menu, select **\$CaSMAuthenticate**.
- In the **Service URL** field, enter the hostname and port number of the payroll service.

As a result, the SAML assertion with the CA SiteMinder single sign-on token is sent to the payroll service. The payroll service will consume the SAML assertion and thus authenticate the user. If the authentication is successful, then the back-end server sends a randomly generated identity token back to Intel SOA Expressway.

33. In the CA SiteMinder sequence, insert a **Reply** action. In the Reply action, select **\$SendSAML_CaSM.body** from the **Message Data from** drop-down menu. As a result, the token generated by the payroll service is sent back to the client. From this point onward, the client must send messages with this token in it. This allows the back-end server to determine whether the client should be trusted or not. The token is valid for the length of the HTTP session, after which client authentication must be performed again.

34. In the payrollService sequence, insert **Invoke** and **Reply** actions.

35. In the Invoke action, perform the following steps:

- In the **Message data from** drop-down menu, select **\$Receive.body**.
- In the **Service URL** field, enter the URL of the back-end server.

As a result, messages are sent by the client to the payroll service. These messages will only be processed by the back-end server if the message contains the randomly generated identity token.

36. In the Reply action's **Properties** view, select **\$Invoke.body**. As a result, the message responses sent by the back-end server are forwarded to the client.

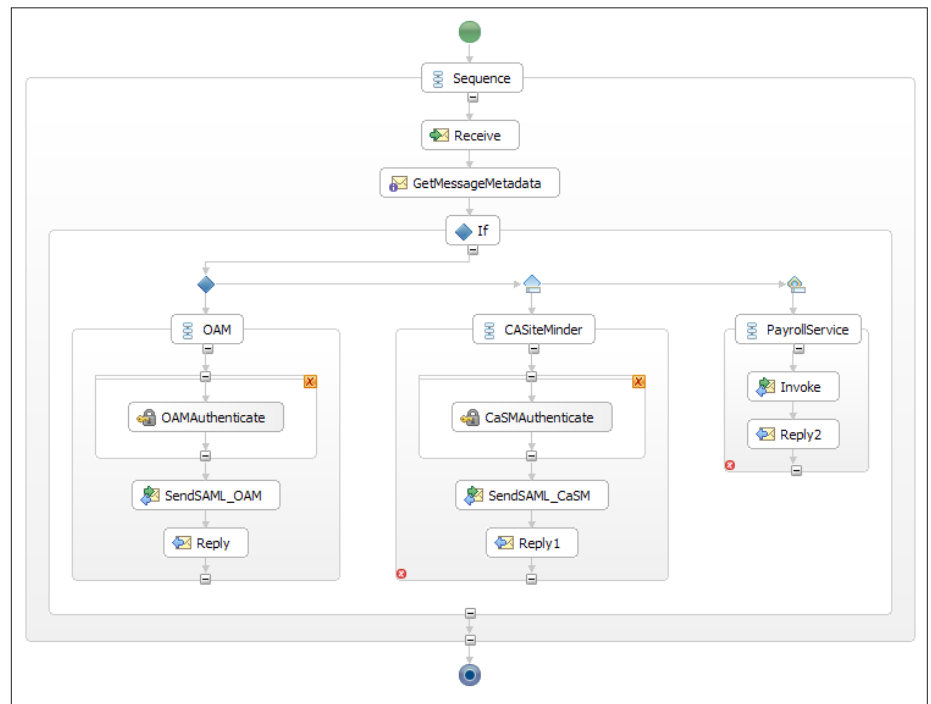


Figure 19. Complete Workflow

Use Case: Secure Credential Federation for a Hybrid Cloud Environment

Deploying enterprise applications into a private cloud is an increasingly popular way to maximize the efficiency of enterprise data centers. However, when the private cloud reaches capacity, additional instances deployed in a public cloud can be federated to bear the heightened load. Enterprises are expanding their use of public clouds to scale up capacity for peak usage periods. This trend is gaining momentum because application deployment to public clouds reduces costs by allowing enterprises to pay for only the capacity they need when they need it.

However, by scaling application instances out to public clouds, enterprises risk exposing their user identities and credentials, creating the potential for an attacker to gain unauthorized entry into the enterprise domain. To avoid the theft of the identities, enterprise users must be authenticated within the enterprise perimeter and their identities mapped to different identities provisioned to the public cloud instances.

In this use case, an enterprise application is deployed in a private enterprise cloud scaled for average application usage. During usage peaks, when the private cloud reaches capacity, application instances must be started in a public cloud. The enterprise needs a solution to authenticate and authorize the user to access the public cloud instance while protecting the enterprise identity.

Intel SOA Expressway Solution

Intel SOA Expressway is a service gateway that can do the following:

- Enforce security policies across security domains.
- Delegate authentication and authorization to a local credential directory.
- Log authentication failures for auditing.
- Proxy messages between two endpoints. The message could be binary, XML, REST, or SOAP.

In this use case, Intel SOA Expressway is placed on the enterprise perimeter to proxy public cloud instances hosted in the Amazon Elastic Compute Cloud* (EC2) service. When a user attempts to connect to a public cloud instance, the Intel SOA Expressway service gateway authenticates the user's credentials against the enterprise identity store, and on successful authentication, maps the enterprise identity to a public cloud identity, never exposing the protected identity or credentials outside the enterprise perimeter. Then, Intel SOA Expressway starts the cloud instance and signs the user in with his mapped public cloud identity.

Configuring Cloud Federation

To create the hybrid cloud environment, enterprise data center operators and application architects must select an Amazon EC2 platform and create an application image to deploy on the platform. This should be straightforward given the choices Amazon offers.

The data center operators then deploy Intel SOA Expressway in the enterprise DMZ and set it up as a proxy for the public cloud instances. The proxy service determines whether to start and connect to application instances in the private cloud

or in the public cloud. For the public cloud instances, the proxy service also orchestrates the authentication and authorization process before starting an instance in the public cloud. This eliminates the cost of an unnecessary instance on the CPU that an authentication or authorization failure would cause.

Administrators would follow these steps to configure Intel SOA Expressway for the authentication and authorization process:

1. Configure Intel SOA Expressway to connect to the enterprise identity directory.

The settings depend on whether the directory is an LDAP or Active Directory server. Essentially, the settings needed are the server address and port, a trusted user account (like an administrator), and the part of the directory where the user entries are based.

2. Configure an identity service in Intel SOA Expressway to connect to the application instance on the public cloud. This consists of configuring the information needed to build the SAML assertion sent to the public cloud instance to sign in the user with his public cloud identity.
3. Configure any authorization policies that apply to the cloud application:
 - a. Configure the built-in policy decision point with the applicable XACML policies based on user ID, group ID, or role within the enterprise.
 - b. In the identity service, enable an additional policy enforcement step that extracts the necessary subject and resource information needed by the decision point.
4. Activate the identity service.

A Detailed Look at the Federation Process

During application deployment, the application architects and data center operators must size the enterprise private cloud. Then the data center operators deploy Intel SOA Expressway at the enterprise perimeter and configure it to act as a proxy to the public cloud running on the Amazon EC2 service. With Intel SOA Expressway linked to the enterprise directory and the identity service configured and running, the application is ready for users to sign on.

When a user is unable to sign on to the private cloud application because the cloud has reached capacity, Intel SOA Expressway begins the authentication and authorization process for the public cloud instance (see Figure 21).

If the user has already signed on to the enterprise through an enterprise single sign-on solution like Integrated Windows Authentication, the Intel SOA Expressway identity service knows the user has been authenticated and skips asking the user for his credentials. In other cases, the identity service presents the user a sign-on page, where the user's credentials must be presented. After receiving the credentials, Intel SOA Expressways verifies these credentials against the enterprise directory.

If an authentication failure occurs, Intel SOA Expressways logs an authentication failure event to the audit log for later investigation by the security administrators. Because the failures occur within the enterprise authentication domain, the log contains authentication failure events for both the private and public cloud instances. Therefore, authentication failures for the private and public clouds are integrated.

When a user is both authenticated and authorized, Intel SOA Expressway finishes the public cloud access by starting the public cloud application instance and logging in the user (see Figure 22). The user's enterprise identity is mapped to the user's public cloud identity, and the user begins his work with the application.

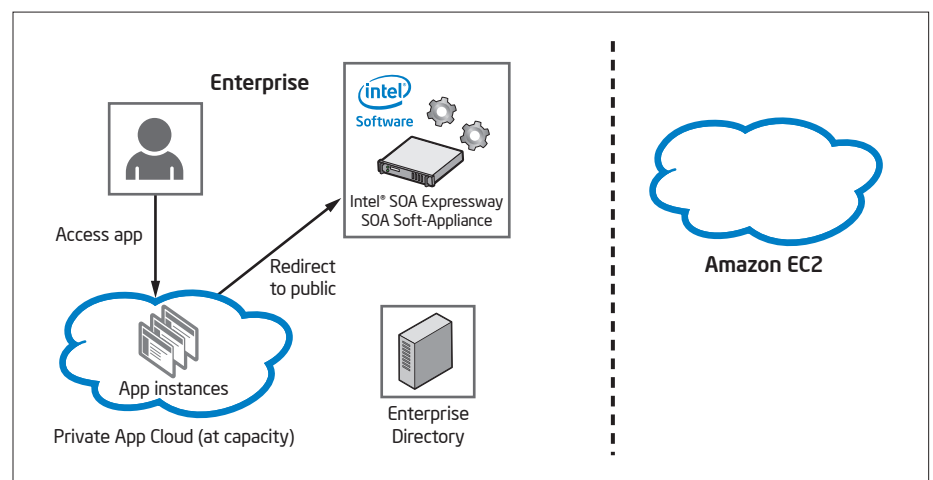


Figure 21. Initiating Redirection to the Public Cloud in a Hybrid Cloud Environment

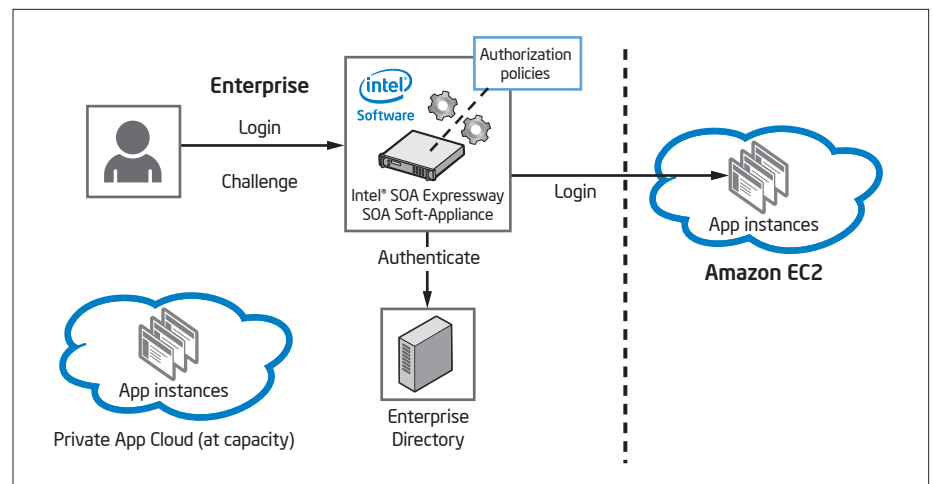


Figure 22. Secure Credential Federation to the Public Cloud in a Hybrid Cloud Environment

Use Case: Two-factor Authentication for Ensuring Client Credentials Are Valid

Enterprises are expanding their use of public clouds to host applications. In some cases, the public cloud is used to scale up capacity for peak usage periods. Increasingly, however, enterprise applications are completely hosted in the public cloud. This trend is gaining momentum because application deployment to public clouds creates significant cost savings by allowing enterprises to pay for only the capacity they need when they need it.

However, moving to public clouds creates new security issues, namely protecting enterprise data from unauthorized access. Enterprise users must be authenticated and authorized to use the cloud-based application. In order to protect enterprise identities and enable a strong authentication method, an enterprise-based gateway integrated with a two-factor authentication solution is needed.

Two-factor authentication is gaining popularity as a way to increase the security of user authentication. When authenticating with two factors, the user supplies something he or she knows (secret password) along with information about something the user has. The popular Intel® vPro Technology-based platform includes a Trusted Platform Module (TPM), which securely holds a private and public key pair that can be used to attest the user is signing on from a trusted system. The system itself then acts as the second factor to secure the authentication process.

In this use case, an enterprise application is deployed to a public cloud provider, and during sign-on the application redirects sign-on and authentication to the Intel SOA Expressway service gateway within the enterprise. Intel SOA Expressway authenticates the user's credentials against the enterprise identity store using a second factor held by the TPM technology built into the client system.

Intel SOA Expressway Solution

Intel SOA Expressway is a service gateway that can do the following:

- Enforce security policies across security domains.
- Delegate authentication and authorization to a local credential directory.
- Log authentication failures for auditing.
- Proxy messages between two endpoints. The message could be binary, XML, REST, or SOAP.

In this use case, Intel SOA Expressway is placed at the enterprise perimeter to transparently proxy application client sign-on and service requests to application instances on the public cloud. During sign-on, Intel SOA Expressway acts as an identity provider (IdP) and initiates the sign-on authentication process. As the client presents its credentials, it also presents a nonce that is signed with the TPM's private key as the second factor of the strong authentication process. The credentials are authenticated against the enterprise credential directory (IdM) and the nonce signature verified, and on successful authentication, Intel SOA Expressway presents an SAML token to the cloud application. In this way, the enterprise identities and credentials are not exposed beyond the enterprise perimeter, and the enterprise benefits from the stronger authentication of its users.

To support compliance needs, the audit log feature of Intel SOA Expressway captures the authentication and authorization events into an audit log. Because the TPM gives a second authentication factor, the authentication events can reveal when an access is attempted by untrusted users or trusted users on untrusted systems. Administrators can filter the log for such events to distinguish the kinds of rogue access attempts being made on the public cloud application.

Configuring Two-Factor Authentication

When the enterprise deploys an application to a public cloud platform, the enterprise security architect enables sign-on to the cloud application through Intel SOA Expressway deployed in the enterprise DMZ. The security architect configures the Intel SOA Expressway identity service to collect the user's credentials and connect to the enterprise identity directory to authenticate those credentials. In addition, the security architect configures the identity service to enable signing using the TPM on his client system. This second factor ensures that the user is signing on from a trusted device and thereby strengthens the protection against rogue access to the cloud application.

Administrators would follow these steps to configure the enterprise for the two-factor authentication method described previously:

1. Configure Intel SOA Expressway to connect to the enterprise identity directory.

The settings depend on whether the directory is an LDAP or Active Directory server. Essentially, the settings needed are the server address and port, a trusted user account (like an administrator), and the part of the directory where the user entries are based.

2. Configure an identity service in Intel SOA Expressway to connect to the cloud application:
 - a. Enable signing using the TPM during user authentication. The administrator needs to collect the TPM public keys from user systems to put in the key store.
 - b. Configure the information needed to build the SAML assertion given to the cloud application to sign on a user.
 - c. Configure the user provisioning process. There are two options: express (on-demand) or bulk (transfer all users then periodically synchronize).

3. Configure any authorization policies that apply to the cloud application:
 - a. Configure the built-in policy decision point with the applicable XACML policies based on user ID, group ID, or role within the enterprise.
 - a. In the identity service, enable an additional policy enforcement step that extracts the necessary subject and resource information needed by the decision point.
4. Activate the identity service.

When the identity service is activated, Intel SOA Expressway performs the bulk provisioning process, if configured, and users can sign on to the cloud application.

A Detailed Look at the Authentication Process

Once the security architect completes the two-factor authentication configuration and activates the identity service, Intel SOA Expressway becomes the enterprise proxy to the publicly deployed cloud application. Users work with the cloud application on trusted systems deployed within the enterprise. When the user accesses the cloud application and begins to sign on, Intel SOA Expressway transparently intercepts the user access to the cloud to begin the two-factor authentication process (see Figure 23).

Intel SOA Expressway initiates the sign on by challenging the user to present his credentials. The user is presented with a sign-on page asking for his user ID and password (see Figure 24). When the user supplies his ID and password, a script in the sign-on page triggers the client system to sign a nonce using the TPM's built-in RSA private key. This signing attests that the user is signing on from a trusted system.

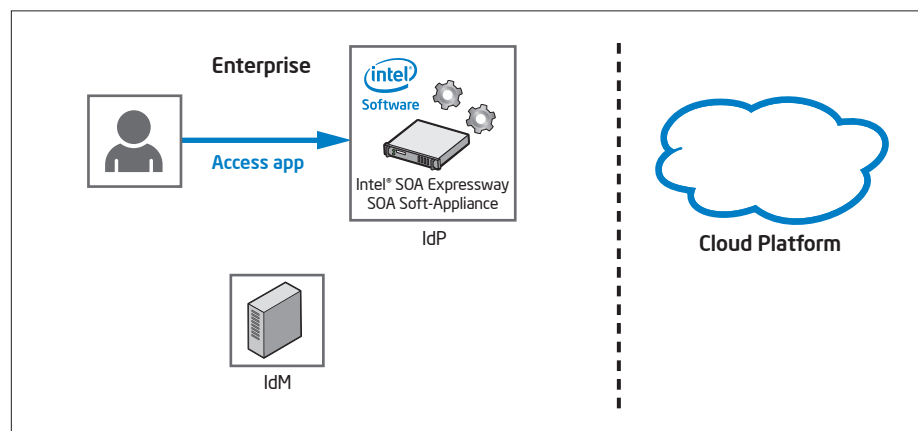


Figure 23. Initiating the Two-factor Authentication Process

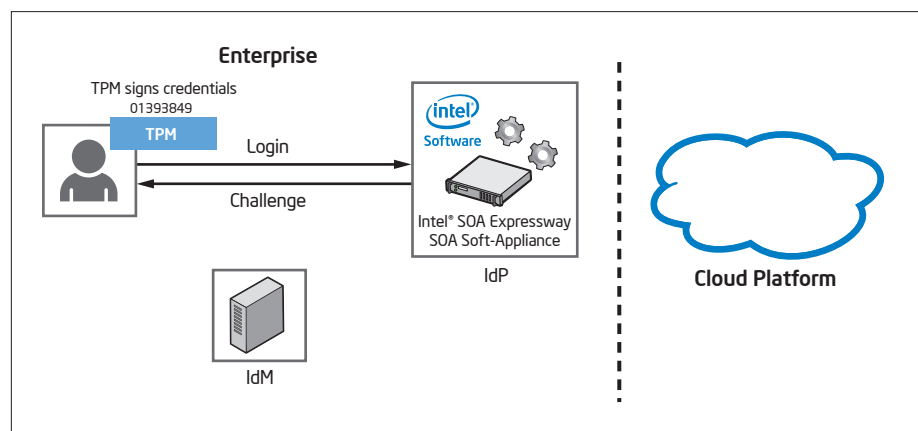


Figure 24. Gathering User Credentials and Trusted System Attestation

After Intel SOA Expressway has gathered the credentials and signed nonce, it authenticates the user and the trusted system in the enterprise directory (see Figure 25). Intel SOA Expressway queries the directory, which can be either LDAP or Active Directory, for the user identity information. On a successful look up, the user's credentials are checked, and the TPM signature is verified with the stored public keys.

At this point, if the user is not authenticated as a trusted user or if he is a trusted user but not on a trusted system, the authentication fails. The user is not signed on to the cloud application, and the enterprise is protected from a rogue access to the application and its data. Intel SOA

Expressway logs the details of the failed authentication in the auditing log, creating an audit trail for security administrators to investigate later.

On a successful user and system authentication, Intel SOA Expressway then determines any and all authorization policies that apply to the user when accessing the cloud application (see Figure 26). The user's ID, group, and role within the enterprise are considered when evaluating the applicable policies. This policy evaluation results in a decision to permit or deny access to the application and its data. In this way, Intel SOA Expressway acts a policy enforcement point (PEP) and policy decision point (PDP).

Now that the user has been authenticated and the authorization decision permits access, the user is signed on to the cloud application. The authentication and authorization process has been conducted securely within the enterprise, out of view of any eavesdroppers between the enterprise perimeter and the public cloud. For further security, Intel SOA Expressway maps the user's identity to a public identity, thereby protecting the security of the user's enterprise identity and password.

After sign on, Intel SOA Expressway continues to act as proxy for the cloud application. This allows for continued auditing, event monitoring and logging, and potentially additional authorization decisions for access to modules and functions of the cloud application.

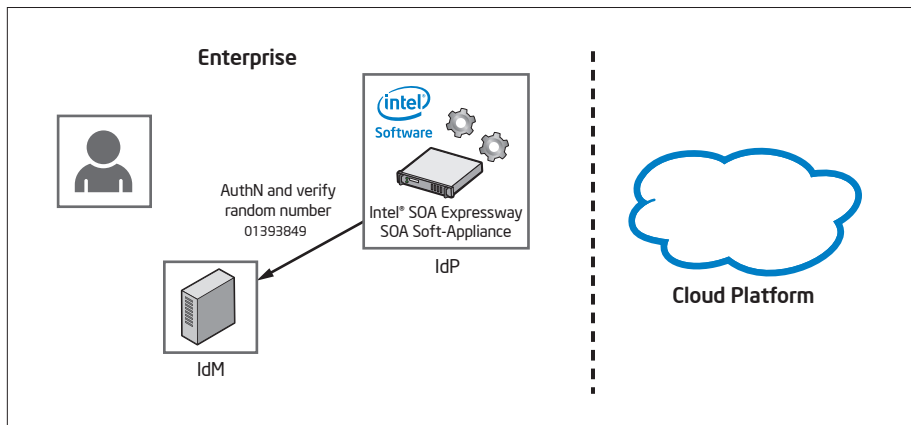


Figure 25. Verifying the User and Trusted System

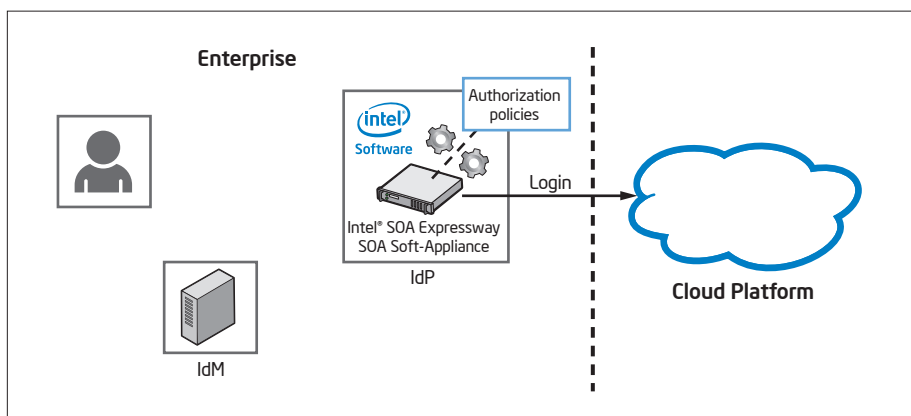


Figure 26. Authorizing the User to Sign On to the Cloud Application

Execution and Results

While each case is unique, the workflow execution and results are similar. The following procedure generally describes how each Intel SOA Expressway application was tested for each use case:

1. A client sends a login request to the cloud, requesting access to a particular resource.
2. Intel SOA Expressway transparently intercepts the login request, authenticates the identity, and then determines appropriate access rights to the requested service.
3. If authentication and authorization are successful, then Intel SOA Expressway becomes a proxy between the client and the cloud, where each request or command the client sends is sent to the runtime and then forwarded to the cloud service. In turn, any responses sent from the service to Intel SOA Expressway are forwarded to the client.
4. To verify whether a message transaction was successful, view the dashboard on the Intel SOA Expressway Management Console.
5. To test the delegated authentication, send invalid login credentials.
6. Verify that the message transaction failed by viewing the Intel SOA Expressway Management Console.
7. To verify that the failed login attempt is recorded for auditing purposes, perform the following steps in the Intel SOA Expressway Management Console.

- Select the **Logs** tab.
- In the Logs tab, select the **Exceptions**.
- Select the **Search** button. As a result, an authentication error message will display (see Figure 29).

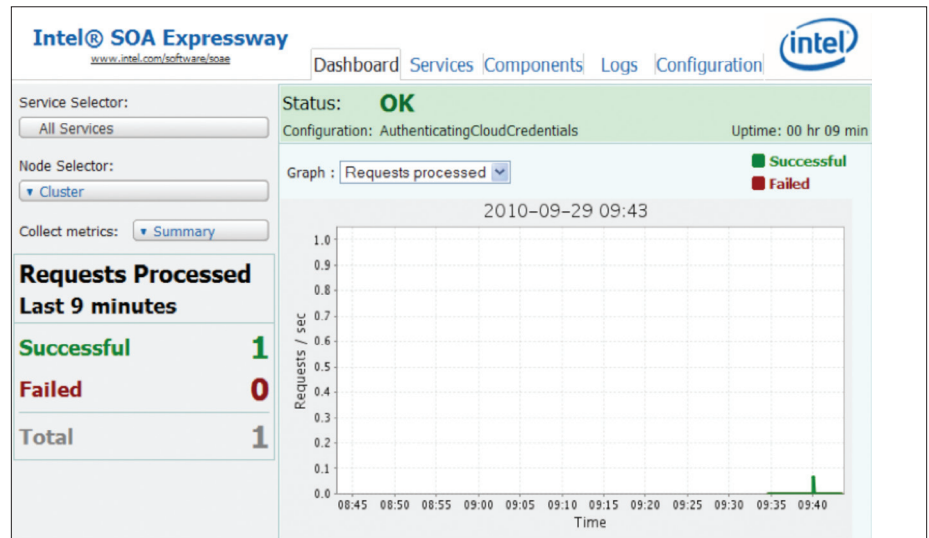


Figure 27. SOA Expressway Dashboard view

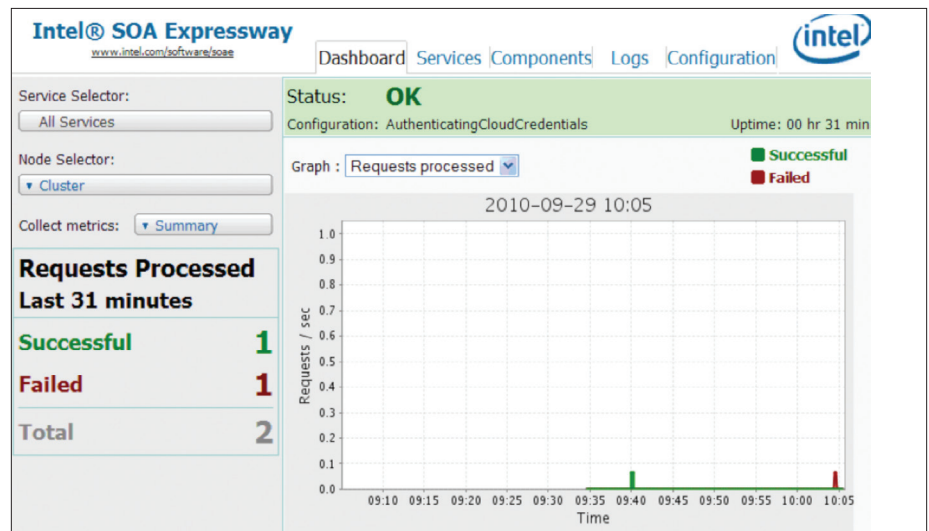


Figure 28. SOA Expressway Dashboard view

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP:Body>
    <SOAP:Fault>
      <faultcode xmlns:X="http://www.intel.com/soae/localService/
aaaDefinition-2007a/">X:oracleAccessManagerAuthenticationFault</fault-
code>
      <faultstring>Uncaught BPEL fault http://www.intel.com/
soae/localService/aaaDefinition-2007a/:oracleAccessManagerAuthent
icationFault executing instruction #230: RETHROW [[file=../app/
Add.bpel, line=580, path=/bpws:process[1]/bpws:sequence[1]/ /
bpws:extensionActivity[3]/soae-ab:AAAAction[1]/bpws:scope[1]/
bpws:scope[1]/bpws:sequence[1]/bpws:scope[1]/bpws:faultHandlers[1]/
bpws:catch[17]/bpws:scope[1]/bpws:sequence[1]/bpws:rethrow[1],
name=AAA]</faultstring>
    </SOAP:Fault>
  </SOAP:Body>
</SOAP:Envelope>
```

Figure 29. This example is of an authentication failure with Oracle Access Manager.

Next Steps

The following process describes how to develop a portable security and integration architecture for cloud deployments using Intel SOA Expressway:

1. Virtualize services.
2. Create security policies.
3. Set up an SAML-based token exchange.
4. Create and locally manage IDs by:
 - a. Integrating to the identity management
 - b. To access a web service, use a single sign solution by using an STS bridge
 - c. Implementing XACML and WS policies
5. Evolve to a private cloud using Intel SOA Expressway. This involves:
 - a. Testing
 - b. Abstraction from implementing policies on the cloud
 - c. Call back to a policy decision point (this is Intel SOA Expressway)
6. Set up a repeatable security and integration model within Intel SOA Expressway that:
 - a. Is based on AAA and WS-Security standards
 - b. Is easy to shift among cloud platforms as relationships change
 - c. Is easier to scale
 - d. Has cloud bursting
 - e. Overlays data security

Things to Consider

The scalability of the cloud solution could be impacted by:

- Network technology (e.g., 10GigE) and architecture
- Selected storage architecture
- Choice of server hardware for the software version of Intel SOA Expressway or choose the Intel SOA Expressway hardware-appliance form factor instead
- The ability to create templates of workflows and policies so that you can author an application once but use it for many services and operations
- Enabling and disabling access to services depending on the user scenario
- How many nodes you need in an Intel SOA Expressway cluster—you can have up to 24. A cluster is used for load balancing and failover. The more nodes you have, the greater the message throughput and system stability.
- The level of security you need. You can explicitly guard against DoS, XML threats, and various content attacks. However, increasing security eventually decreases message throughput due to heavy security processing.

Glossary

AAA – Authentication, Authorization and Auditing policy. In Intel SOA Expressway, a single AAA policy can extract an identity from a workflow, delegate authentication and resource authorization of that identity to a third party, and then map the authenticated identity to another format, such as SAML.

Intel SOA Expressway – Intel® Service-Oriented Architecture Expressway is a software appliance designed to simplify, accelerate, and secure an enterprise's SOA. It expedites SOA deployments by addressing common SOA bottlenecks; and it accelerates, secures, integrates, and routes XML, web services, and legacy data in a single, easy-to-manage form factor.

RPM – Software stored in a rpm format. This is a Linux package management system.

SOA – Service-Oriented Architecture is a type of software architecture that treats software as a modular service that can be consumed by a client.

TLS – Transport Layer Security is a cryptographic protocol that encrypts communication over a network between two endpoints.

Trust Mediation – The process by which a security gateway establishes trust relationships between a client and a server.

WS-Policies – Web Service Policies is a W3C recommendation that specifies how web services use XML to advertise their policies and how web-service consumers specify their policy requirements.

WS-Security – Web Services Security is an extension to SOAP that applies security to web services. WS-Security is part of the WS-* family of web-service specifications and was published by OASIS.

XACML – eXtensible Access Control Markup Language is a declarative access control policy language implemented in XML and a processing model that describes how to interpret access control policies.

About Intel® SOA Expressway SOA Expressway is a soft-appliance deployed to address common XML and SOA problem areas such as acceleration, security, service mediation and service governance. SOA Expressway is available for any organization deploying services (SOA) or looking to reduce security risks for access to cloud services. It is “ecosystem neutral” and integrates with existing identity management, middleware and security monitoring investments. SOA Expressway is available for standard operating systems such as Windows and Linux and requires no special custom hardware other than standard OEM servers.

References

To learn more about deploying cloud solutions, visit our Intel® Cloud Builders Web site at www.intel.com/cloudbuilders

For more on Intel® SOA Expressway, visit our industry resource site on edge security at www.dynamicperimeter.com

Disclaimers

¹ Based on the statement: “Intel SOA Expressway provides superior performance for high service mediation and governance use cases outpacing IBM DataPower in a direct ‘apples-to-apples’ comparison by 1.5X to 8X at a fraction of the total cost.” See “Intel® SOA Expressway Performance Comparison to IBM DataPower XI50.” Download it at http://www.dynamicperimeter.com/thankyou/Comparison_to_IBM_DataPower_XI50.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined.” Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request. Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order. Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting Intel's Web site at www.intel.com.

Copyright © 2010 Intel Corporation. All rights reserved. Intel, the Intel logo, Xeon, vPro, and Xeon inside are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Printed in USA

1010/RR/OCG/XX/PDF

 Please Recycle

324482-001US

