An Oracle White Paper
February 2011

Securing SOA and Web Services with Oracle Enterprise Gateway

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Introduction

Companies worldwide are actively deploying service-oriented architecture (SOA) infrastructures using web services, both in intranet and extranet environments. While web services offer many advantages over traditional alternatives (e.g., distributed objects or custom software), deploying networks of interconnected web services still presents key challenges, especially in terms of security and management.

Web services can be implemented using different approaches and technologies which need to be secured at the different stages of the request / response cycle between clients (relying parties such as users or applications) and service providers (companies or divisions within a company exposing web services).

Several security layers are defined between clients and web services providers. The first security layer, also known as “perimeter security” or “first line of defense,” is referred to as the demilitarized zone or DMZ. The second security layer, or “green zone” to continue with the military analogy, is located behind the inner firewall of the DMZ. In some cases, the green zone may include several security sub-layers designed to further filter access to web services. Finally, the last security layer, or “last-mile security,” is provided by agents co-located with the web services or applications to be protected.

This document focuses on web services security in the DMZ, provided by Oracle Enterprise Gateway. Oracle Enterprise Gateway operates with and complements other web services security systems, in particular Oracle Web Services Manager, which focuses on the security of web services deployed in the green zone, and Oracle Service Bus.

Oracle Enterprise Gateway is a software solution that provides application-level routing (based on source, target, sender identity, and XML message type); XML conversion, validation and threat scanning; XML acceleration; security (selective encryption and signature of XML messages, decryption and signature validation); monitoring (response time, logging, and alerting); and governance (service access and usage).

Oracle Enterprise Gateway is tightly integrated with Oracle Access Manager, Oracle Entitlements Server, Oracle Web Services Manager, and Oracle SOA Suite to provide transport- and application-level security across all layers involved in web services requests.

Web Services Security

A web service is a program that can be written in any language. What this program can do (i.e., the functionality it implements) is described in a standard XML vocabulary called Web Services Description Language (WSDL). For example, a banking web service may implement functions to check an account, print a statement, deposit and withdraw funds. These functions are described in a WSDL file that any consumer can invoke to access the banking web service. As a result, a consumer does not have to know anything about a web service other than its location and the WSDL file that describes what it can do.

A web service consumer (e.g., a desktop application or a Java Platform, Enterprise Edition (Java EE) client such as a portlet) invokes a web service by submitting a request in the form of an XML document to a web service provider. The web service provider processes the request and returns the result to the web service consumer in an XML document.

Through more recently designed protocols, web services can also be directly invoked from a web browser, as shown in Figure 1.

However they are accessed, web services mainly use the pervasive HyperText Transport Protocol (HTTP) to carry out transactions. This means that traditional network firewalls alone won't be enough to secure access to web services.

Web Services Security Types

Because of their nature (loosely coupled connections) and their use of open access (mainly HTTP), SOA infrastructures implemented by web services add a new set of requirements to the security landscape.

Web services security includes several aspects:

- *Authentication*: Verifying that the user is who they claim to be. A user's identity is verified based on the credentials presented by that user, such as username/password, digital certificate, standard Security Assertion Markup Language (SAML) token, or Kerberos token. In the case of web services, credentials are presented by a client application on behalf of the end user.
- *Authorization (or Access Control)*: Granting access to specific resources based on an authenticated user's entitlements or specific role (e.g., corporate buyer).
- *Confidentiality, privacy*: Keeping information secret. Personally Identifiable Information (PII) or confidential business data could be present in web service request or response messages. Confidentiality of such data can be achieved by encrypting the content of request or response messages using the XML Encryption standard.
- *Integrity, non repudiation*: Making sure that a message remains unaltered during transit by having an authority digitally sign that message; a digital signature also validates the sender and provides a time stamp ensuring that a transaction can't be later repudiated by either the sender or the receiver. XML messages are signed using the XML Signature standard.

Web services security requirements also involve credential mediation (exchanging security tokens in a trusted environment), and service capabilities and constraints (defining what a web service can do, under what circumstances).

Web services security requirements are supported by industry standards both at the transport level and at the application level relying on XML frameworks.

Transport-Level Security

Secure Socket Layer (SSL), otherwise known as Transport Layer Security (TLS), the Internet Engineering Task Force (IETF) officially standardized version of SSL, is the most widely used transport-level data-communication protocol providing:

- Authentication (the communication is established between two trusted parties).
- Confidentiality (the data exchanged is encrypted).
- Message integrity (the data is checked for possible corruption).
- Secure cryptographic key exchange between client and server.

SSL provides a secure communication channel. However, when the data is not "in transit," the data is not protected, which makes the environment vulnerable to attacks in multi-step transactions (SSL provides point-to-point security, as opposed to end-to-end security).

Application-Level Security

Application-level security complements transport-level security. Application-level security is based on XML frameworks defining confidentiality, integrity, authenticity; message structure; trust management; identity propagation.

Web Services Security Layers

Before accessing a web service, a web service request should go through several security layers.

Demilitarized Zone

The demilitarized zone or DMZ is the first line of defense. The DMZ typically includes a firewalling capability that filters incoming requests to web services to make sure these requests are valid.

Web services requests are inspected in the DMZ to prevent various attacks:

- First, the structure of incoming web services requests is validated ("schema validation") to make sure messages can be correctly parsed.
- Denial-of-service through "XML bombs." XML bombs are small messages designed to overpower the XML parser by growing out of control the XML message to be parsed, which would ultimately bring the web server or application server down.

- Message throttling. Message throttling is applied to prevent the maximum number of concurrent web services requests from being exceeded. The maximum number of concurrent web services requests is usually defined in a service-level agreement (SLA) between the client (requester) and the web service provider.

DMZ security also includes confidentiality, message integrity, authentication, and authorization as described above.

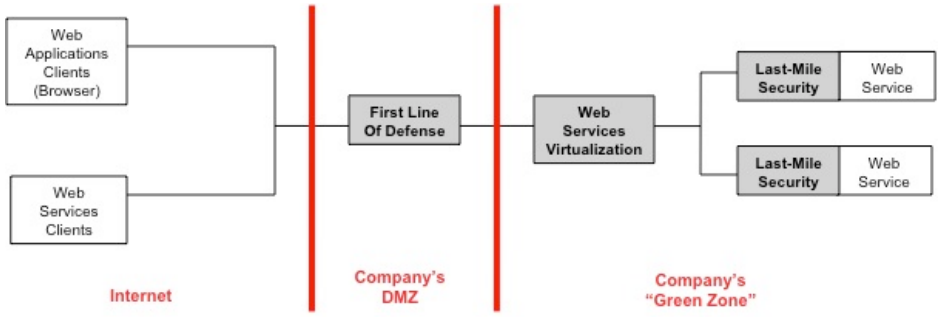

Figure 1: Web Services Security Layers

Green Zone

The "green zone" is located behind the inner firewall of the DMZ. Once a web service request is validated in the DMZ, it is forwarded to the requested web service. The green zone may include several sub-layers to redirect requests to the appropriate web service ("web service virtualization"), and to provide last-mile security. Web service virtualization is generally provided by a service bus, and last-mile security is ensured by agents directly injected in the web service application so that there's no unprotected space between the first line of defense and the requested web service (unprotected space could lead to "man-in-the-middle" attacks where an attacker inserts himself in the communication channel between client and web service provider, masquerading as the actual web service requester).

Introducing Oracle Enterprise Gateway

Oracle Enterprise Gateway primarily provides first line of defense in the DMZ. Oracle Enterprise Gateway is designed to secure, accelerate, and integrate all types of traffic between web services requesters and web services providers.

Oracle Enterprise Gateway provides policy-driven, efficient processing of multiple data formats (described later in this document); protocol and content transformation on the wire; XML filtering and access control to services; threat management; and SOA infrastructure governance.

Oracle Enterprise Gateway can be deployed standalone or as an integral component of a strategic enterprise SOA infrastructure, interfacing with enterprise service bus, enterprise management, and identity management platforms.

In addition to being deployed in the DMZ, Oracle Enterprise Gateway can be deployed at the data center for acceleration and application off-loading to relieve processing bottlenecks, and in front of key enterprise applications to identify service usage and behavior.

Oracle Enterprise Gateway Architecture and Components

Oracle Enterprise Gateway is available as a single executable for the Microsoft Windows, Linux, and Oracle Solaris platforms.

As mentioned previously, in a typical deployment scenario Oracle Enterprise Gateway components are deployed in the DMZ. The connection between clients and Oracle Enterprise Gateway is protected by a perimeter firewall, and the connection between Oracle Enterprise Gateway and target web services is protected by a Network Address Translation (NAT) firewall.

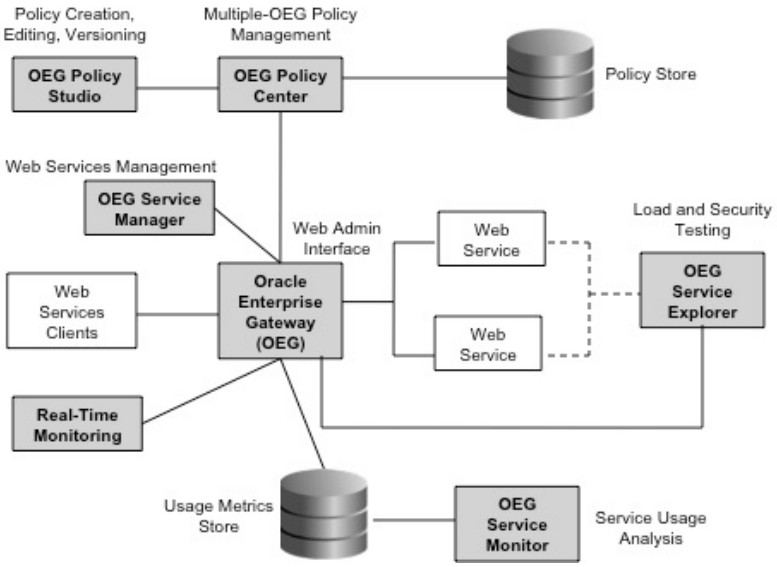

Figure 2: Oracle Enterprise Gateway Components

Oracle Enterprise Gateway Service Explorer

OEG Service Explorer is a web service test client used to generate test messages that are sent to Oracle Enterprise Gateway and back to OEG Service Explorer. OEG Service Explorer supports both SOAP-based and RESTful invocations.

Oracle Enterprise Gateway Policy Studio

OEG Policy Studio is a policy management tool that enables an administrator to easily configure policies and Oracle Enterprise Gateway settings to control and protect all deployed web services. For example, OEG Policy Studio enables you to create and assign policies, version policies, configure the full range of Oracle Enterprise Gateway settings, and manage your Oracle Enterprise Gateway

deployments. OEG Policy Studio is typically installed on a separate machine from Oracle Enterprise Gateway to enable remote administration.

Oracle Enterprise Gateway Policy Center

OEG Policy Center is intended for managing policy deployments across multiple Oracle Enterprise Gateways (multiple-gateway policy management). OEG Policy Center manages policy migration between development, staging, and production.

Oracle Enterprise Gateway Service Manager

OEG Service Manager is a web-based system administration tool that simplifies Oracle Enterprise Gateway management tasks. OEG Service Manager enables you to manage your services and policies deployed on Oracle Enterprise Gateway. For example, with OEG Service Manager you can register web services and assign policies to them.

Web Administration Interface

You can manage Oracle Enterprise Gateway using the Web Administration Interface. For example, you can configure the hostname and Internet Protocol (IP) address for Oracle Enterprise Gateway, the default Oracle Enterprise Gateway to use, the Domain Name System (DNS) server to use, the system time, the Secure Shell (SSH) server, and system users.

Oracle Enterprise Gateway Service Monitor

OEG Service Monitor is a separately installed component that generates reports and charts based on the usage metrics of all the Oracle Enterprise Gateways in your network. OEG Service Monitor provides integration with databases such as Oracle DB, Oracle MySQL Server, and Microsoft SQL Server. OEG Service Monitor also includes a Real-Time Monitoring console providing remote real-time monitoring of XML traffic processed by Oracle Enterprise Gateway. This web-based console enables administrators to detect malicious activity in real time, and to take precautionary actions if they feel a service is under attack.

Note: OEG Policy Studio, OEG Policy Center, OEG Service Explorer, and Oracle Enterprise Gateway itself run on the operating system directly. Real-Time Monitoring and OEG Service Manager are web-based applications accessed through a browser (e.g., http://localhost:8090/). The Oracle Enterprise Gateway welcome page contains links to all web-based tools and related information.

Oracle Enterprise Gateway Concepts

This section describes the main concepts in the Oracle Enterprise Gateway product architecture.

Policy

A policy is a network of message filters in which each filter is a modular unit that processes a message. A message can traverse different paths through the network, depending on which filters succeed or fail.

For example, this enables you to configure policies that route messages that pass a Schema Validation filter to a back-end system, and route messages that pass a different Schema Validation filter to a different system.

Filter

A filter is an executable rule that performs a specific type of processing on a message. For example, the Message Size filter rejects messages that are greater or less than a specified size. There are many categories of message filters available with Oracle Enterprise Gateway, including authentication, authorization, content filtering, signing, and message conversion. In OEG Policy Studio, a filter is displayed as a block of business logic that forms part of an execution flow known as a circuit.

Circuit

A circuit is a series of filters through which a message passes. A circuit can contain other circuits, which enables you to build modular reusable policies. In OEG Policy Studio, a circuit is displayed as a path through a set of filters. A filter can have only one Success Path and one Failure Path. You can use these success paths and failure paths to create sophisticated rules. For example, if the incoming data matches Schema A, scan for attachments and route to Service A, otherwise route to Service B.

Message Attributes

Each filter requires input data and produces output data. This data is stored in message attributes. You can use specific filters to create your own message attributes and their values. The Trace filter enables you to trace message attribute values at execution time.

Faults

When a SOAP transaction fails, you can use a SOAP fault to return error information to the SOAP client. By default, Oracle Enterprise Gateway returns a basic SOAP fault to the client when a message filter fails. You can add a SOAP Fault filter to a policy to return more detailed error information to the client.

Policy Shortcut

A policy shortcut enables you to create a link from one policy to another policy. For example, you can create a policy that inserts security tokens into a message, and another that adds HTTP headers. You can then create a third policy that calls the other two policies using Policy Shortcut filters.

A policy shortcut chain enables you to run a series of policies in sequence without needing to create a policy containing policy shortcuts. In this way, you can create modular policies to perform certain specific tasks, such as authentication, content filtering, returning faults, or logging, and then link these policies together in a sequence using a policy shortcut chain.

You can also use OEG Service Manager to automate the creation of a policy shortcut chain simply by dragging and dropping existing policies into a composite policy.

Alerts

Oracle Enterprise Gateway can send alert messages for specified events to various destinations. System alerts are usually sent when a filter fails, but they can also be used for notification purposes. Oracle Enterprise Gateway can send system alerts to Windows Event Log, UNIX/Linux `sylsog`, Simple Network Management Protocol (SNMP) servers, and email recipients.

Policy Container

A policy container is used to group similar policies together (for example, all authentication or logging policies), or policies that relate to a particular service.

A number of useful policies are provided in the Policy Library container (for example, policies that return faults, and policies that block threatening content). You can add your own policies to this container, and add your own policy containers to suit your specific requirements.

Policy Context

Policies can execute in a specified context. You can set a context by associating a relative execution path or listener with a policy. When a policy is called from another policy, the context is set to the calling policy name (for example, Authenticate).

Listeners

You can define different types of listeners and associate them with specific policies. Listeners include HTTP/S, Directory Scanner, Post Office Protocol (POP) mail server connection, Java Message Service (JMS) connection, and TIBCO Rendezvous (RV) and Enterprise Message Service (EMS) connection.

Oracle Enterprise Gateway is designed to support protocol mediation (for example, receiving a SOAP request over JMS, and transforming it into a SOAP/HTTP request to a web service end-point).

Remote Hosts

You can define a remote host when you need more control over the connection settings to a particular web service server.

The available connection settings include the HTTP version, IP addresses, timeouts, buffers, and caches. For example, by default Oracle Enterprise Gateway uses HTTP 1.1; you can force it to use HTTP 1.0 using Remote Host settings.

Servlet Applications

Oracle Enterprise Gateway provides a web server and servlet application server that can be used to host static content (for example, documentation for your project), or servlets providing internal services. This feature is not meant to replace a bona fide enterprise Java EE server such as Oracle WebLogic server, but rather to enable you to write functionality using technology such as servlets.

Configuration

A configuration is a store of information required to run Oracle Enterprise Gateway. For example, a specific configuration instance can store certificates, users, core policies and web services, external connections, or listeners. A configuration can have a number of versions, which are created by users. You can use OEG Policy Studio to deploy configuration versions to Oracle Enterprise Gateway processes, and copy existing versions to create new configurations.

Process

A process is a running instance of Oracle Enterprise Gateway. You can use OEG Policy Studio to configure and deploy Oracle Enterprise Gateway processes.

Service Virtualization

When you register a web service and deploy it to Oracle Enterprise Gateway, Oracle Enterprise Gateway virtualizes the web service. Instead of connecting to the web service directly, clients connect through Oracle Enterprise Gateway. Oracle Enterprise Gateway can then apply policies to messages sent to the destination web service (for example, enable security, monitoring, and acceleration).

XML Firewalling and Acceleration

Oracle Enterprise Gateway provides the following XML firewalling capabilities:

- *XML contents attacks*: checking for XML well-formedness; XML document size; XPath and XQuery injection; SQL injection; XML encapsulation; XML viruses; scanning outgoing messages for sensitive content based on metadata or regular expression patterns; detecting XML bombs and XML clogging; scanning WSDL files.
- *XML schema and Document Type Definition (DTD) attacks*: schema validation; checking for XML entity expansion and recursion; schema poisoning; recursive elements, jumbo tag-names; malicious includes (also called XML external entity (XXE) attacks).
- *Cryptographic attacks*: checking for Public Key denial of service; replay attacks.
- *SOAP attacks*: SOAP operation filtering; checking for rogue SOAP attachments (viruses and other).
- *Communication attacks*: HTTP header and query string analysis; IP address filtering; traffic throttling.

Oracle Enterprise Gateway accelerates performance as follows:

- *Processing Offload*: Oracle Enterprise Gateway can be used to offload the heavy lifting of XML from application servers on to the network. This frees up resources on application servers and enables applications to run faster.
- *VXA Platform*: The VXA engine is integrated into Oracle Enterprise Gateway to accelerate the essential XML security primitives. The VXA engine provides XML processing at faster levels than those performed by common Java API for XML Processing (JAXP) implementations in application servers and other applications that sit downstream from Oracle Enterprise Gateway. The VXA

engine performs Document Object Model (DOM) processing, XPath, Extensible Stylesheet Language Transformation (XSLT) conversion, and XML validation.

In addition, Oracle Enterprise Gateway uses OpenSSL to perform cryptographic operations, such as encryption and decryption, signature generation and validation, and SSL tunneling. OpenSSL exposes an *Engine API*, which makes it possible to plug in alternative implementations of some or all of the cryptographic operations implemented by OpenSSL. When configured appropriately, OpenSSL calls the engine's implementation of these operations instead of its own.

For example, a particular engine may provide improved implementations of the asymmetric operations RSA and DSA. This engine can then be plugged into OpenSSL so that whenever OpenSSL needs to perform either an RSA or DSA operation, it calls out the engine's implementation of these cryptographic algorithms rather than its own.

Typically, OpenSSL engines provide a hardware implementation of specific cryptographic operations. The hardware implementation usually offers improved performance over its software-based counterpart, referred to as *cryptographic acceleration*.

Cryptographic acceleration can be configured at the process level in Oracle Enterprise Gateway. The OEG Policy Studio is used to configure the Oracle Enterprise Gateway process required to use an OpenSSL engine instead of the default OpenSSL implementation.

Security Standards Supported

Much like Oracle Web Services Manager, Oracle Enterprise Gateway supports all key XML security policy standards.

WS-Security

WS-Security specifies SOAP security extensions that provide confidentiality using XML Encryption and data integrity using XML Signature. WS-Security also includes profiles that specify how to insert different types of binary (e.g., Kerberos) and XML (e.g., SAML) security tokens in WS-Security headers for authentication and authorization purposes. Oracle Enterprise Gateway supports WS-Security 1.0 and 1.1.

WS-Trust

In a message exchange using WS-Security only, it is assumed that both parties involved in the exchange have a prior agreement on which type of security tokens they must use for sharing security information. However, there are cases where these parties don't have such an agreement, as a result trust must be established before exchanging messages. Trust between two parties exchanging SOAP / WS-Security-based messages is established by implementing the WS-Trust specification.

In this context, establishing trust between partners means that the service provider must trust the security information submitted by the requesting party (client). This trust must be brokered when both parties don't use the same security tokens (i.e., the incompatibility of the security token formats must be resolved). WS-Trust addresses these issues by:

- defining a request / response protocol where a client (or Oracle Enterprise Gateway on its behalf) sends a `RequestSecurityToken` (RST) met with a `RequestSecurityTokenResponse` (RSTR),
- providing a Security Token Service (STS) that enables security token exchange, token issuance, and token validation.

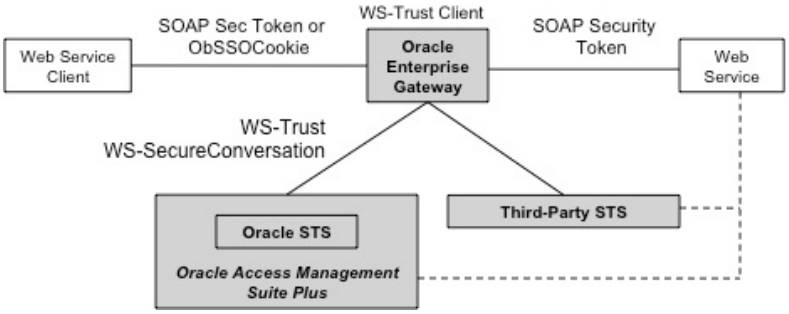

Figure 3: Oracle Enterprise Gateway as a WS-Trust Client

A typical scenario involves a client using X.509 certificates accessing a service provider using SAML, with no prior trust established between client and service provider.

- The client sends (possibly through Oracle Enterprise Gateway) an initial request to the service provider; The SOAP / WS-Security message includes an X.509 certificate and uses XML Signature to establish the identity of the client.
- Oracle Enterprise Gateway recognizes that the client is using an X.509 certificate and that the service provider is expecting a SAML assertion, so Oracle Enterprise Gateway sends a WS-Trust request to the STS including an RST element (*Note*: the STS could be Oracle STS or any other WS-Trust compliant STS). Included in the RST element is the token type requested by the service provider, which in this case is SAML. In essence, the client is asking the STS to map a SAML assertion to its own X.509 certificate.
- The STS sends back a message including an RSTR element with an embedded `RequestedSecurityToken` including the SAML assertion (in this case, the `saml:Subject` element is the converted client identifier, i.e., the X.509 certificate).
- Oracle Enterprise Gateway can now forward to the service provider a request including a SAML assertion inserted in a WS-Security header (the SAML assertion is signed by the STS, and the `saml:ConfirmationMethod` used is *Sender-Vouches*).
- The service provider can now accept the request.

As mentioned above, Oracle Enterprise Gateway can act as a WS-Trust client to Oracle STS or any other third-party STS.

WS-Policy and WS-SecurityPolicy

WS-Policy enables one to specify policy information that can be used to access web services. A policy is expressed as one or more policy assertions. A policy assertion represents a capability or a requirement. For example, a policy assertion may stipulate that a request to a web service be encrypted with a specific encryption algorithm. WS-Policy is the standard security model for Oracle Fusion Middleware.

WS-SecurityPolicy defines a set of security policy assertions used in the context of the WS-Policy framework. WS-SecurityPolicy assertions describe how messages are secured on a communication path.

Oracle Enterprise Gateway acts as a Recipient and Initiator in a WS-Policy security scenario.

As a Recipient, Oracle Enterprise Gateway advertises to clients the security mechanism that it is enforcing for a service with support for the following security mechanisms: Username authentication with symmetric keys; Mutual certificate security; Transport security (SSL); Message authentication over SSL; SAML authorization over SSL; Endorsing certificate; SAML *Sender-Vouches* with certificates; SAML *Holder-of-Key*; STS-issued token; STS-issued token with service certificate; STS-issued endorsing token.

As an Initiator, Oracle Enterprise Gateway consumes WSDL containing WS-Policy from the target web service. Oracle Enterprise Gateway circuits enable Oracle Enterprise Gateway to securely communicate with the target web service. As an Initiator, Oracle Enterprise Gateway supports WS-Security 1.0 and 1.1 tokens including Username, X.509, SAML, Kerberos, and WS-Trust / WS-SecureConversation scenarios (WS-SecureConversation leverages WS-Security and WS-Trust to define the creation and sharing of security contexts between communicating parties).

All tokens generated in Oracle Enterprise Gateway are WS-I BSP compliant (Web Services Interoperability (WS-I) is an industry consortium promoting interoperability of web services across heterogeneous platforms. WS-I provides implementation guidelines in "profiles." The Basic Security Profile (BSP) provides guidance on the use of WS-Security and the Username, SAML, X.509, and Kerberos security tokens).

Oracle Enterprise Gateway also supports user authentication and authorization of requests to access resources protected by Oracle Access Manager (OAM):

- Authentication of username / password and X.509 certificate.
- Validation of OAM's ObSSOCookie via Oracle Enterprise Gateway's OAM-SSO Token Validation filter.
- SAML support to OAM is supplied via Oracle Enterprise Gateway's standards SAML PDP filters.

Finally, Oracle Enterprise Gateway can validate username/password tokens (HTTP authentication (basic / digest), and WS-Security username token) against a database. Oracle Enterprise Gateway can also retrieve attributes/roles of a user from a database. Custom tokens may also be authenticated using a database, provided that the token can be retrieved from the message using a standard mechanism

such as XPath. For example, a custom token retrieved via an XPath operation can be passed into a SQL stored procedure in order to validate it against a database store of valid tokens.

Transport Protocols Supported

Oracle Enterprise Gateway supports the following transport protocols:

HTTP 1.0 and 1.1; Java Messaging Service (JMS); IBM WebSphere MQ; File Transfer Protocol (FTP); Secure FTP (SFTP); TIBCO Rendezvous; TIBCO Enterprise Message Service (EMS); Simple Mail Transfer Protocol (SMTP); Post Office Protocol (POP); and Transmission Control Protocol (TCP).

Web Services Protocols Supported

Web services clients can access web services using different web services protocols.

Oracle Enterprise Gateway supports the following:

SOAP 1.1 and 1.2; Plain XML (POX); Representational State Transfer (REST); Asynchronous JavaScript and XML (Ajax); and JavaScript Object Notation (JSON, an alternative to XML to represent (serialized) data structures that can be transmitted over a network).

Oracle Enterprise Gateway Deployment

As shown in Figure 4, Oracle provides security for each security layer encountered by a web service request.

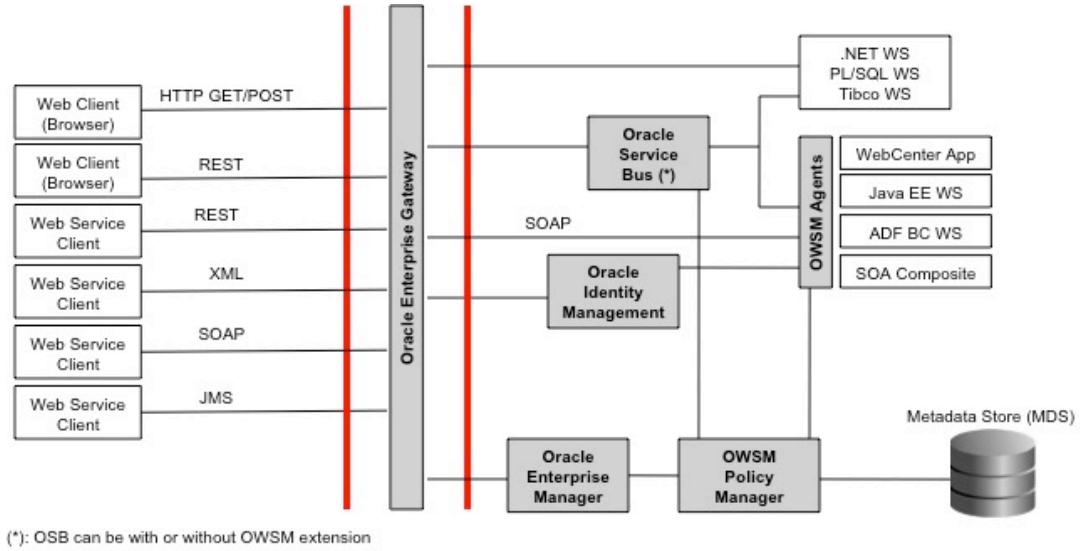

Figure 4: Oracle Enterprise Gateway Deployment

A web service requester can use two types of clients: a web browser for actual users, or an application sending the web service request on behalf of an actual user. In the former case, actual users are most likely authenticated by an identity management infrastructure such as Oracle Access Manager.

As mentioned earlier, clients can use multiple protocols, for example XML over SSL, SOAP, JMS, and REST to invoke RESTful web services.

As a first line of defense, Oracle Enterprise Gateway intercepts all the requests for web services in the DMZ and ensures that requests are valid. Oracle Enterprise Gateway authenticates web services requests and then passes those requests to the next security layer in the infrastructure, Oracle Service Bus (OSB).

OSB may or may not include Oracle Web Services Manager (OWSM) security extension. OWSM security extension allows you to use OWSM security policies in OSB, thus leveraging OWSM's centralized policy management and support for the WS-Policy and WS-SecurityPolicy standards. Essentially, the OWSM extension allows OSB to host an OWSM agent's functionality that fully integrates with Oracle Enterprise Gateway, thus combining OSB's web services virtualization capabilities with OWSM's security features.

Once OSB has received the web service request forwarded by Oracle Enterprise Gateway, OSB may re-authenticate that request and redirect it to the appropriate web service end-point, which can be a pure Java EE web service, an Oracle WebCenter application, an Oracle Application Development Framework (ADF) Business Component (BC) object (ADF BC is commonly used in Fusion Applications), or a SOA composite application.

Finally, OWSM agents ensure last-mile security. Each end-point, for example components of a SOA composite such as BPEL processes, is protected by an OWSM agent co-located with the application itself. In this way, OSB (with OWSM extension) and OWSM agents can be centrally managed in a single point of control (Oracle Enterprise Manager's Fusion Middleware Control).

The next section describes how Oracle Enterprise Gateway leverages Oracle Identity Management.

Oracle Enterprise Gateway and Identity Management

Web services security is intimately related to identity management. Oracle Enterprise Gateway enables unified security policies and identity management to be applied to XML applications across the enterprise.

Customers don't have to perform the integration between identity products since Oracle Enterprise Gateway offers a wide variety of connector options into an existing identity management infrastructure, whether it is Oracle Identity Management or another vendor's offering.

Oracle Identity Management includes directory services, identity provisioning, user management and role governance, as well as access control (web and desktop applications single sign on (SSO), strong authentication, fine-grained authorization, and identity federation).

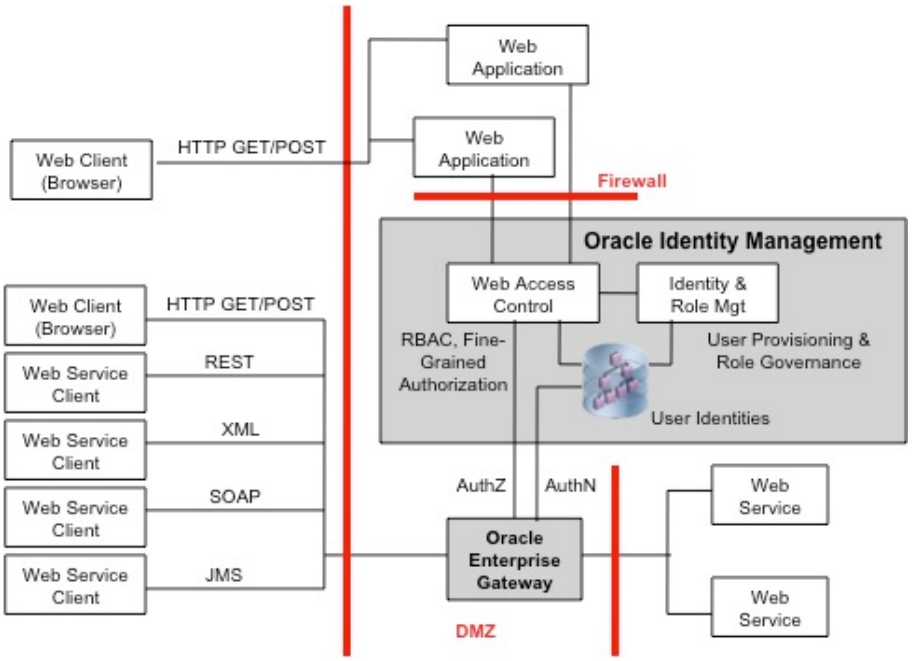

Figure 5: Web Services Security and Oracle Identity Management

Oracle Enterprise Gateway can use Oracle Identity Management (and third-party identity management infrastructures) to perform authentication and authorization of XML traffic.

Integration with Oracle Directory Services

Oracle Enterprise Gateway can directly leverage Oracle's directory services, i.e., Oracle Internet Directory (OID) and Oracle Directory Server, Enterprise Edition (ODSEE) for access control purposes. Oracle Enterprise Gateway can query OID or ODSEE for user profile data. Oracle Enterprise Gateway's Retrieve from Directory Server filter can lookup a user, retrieve that user's attributes, and set them to the `attribute.lookup.list` message attribute, which stores a map of name-value pairs. Oracle Enterprise Gateway also has direct support for Microsoft Active Directory (AD).

Integration with Oracle Access Manager

Oracle Enterprise Gateway's Oracle Access Manager (OAM) SSO Token Validation filter is used to check an Oracle Access Manager SSO token (ObSSOCookie) to make sure that it is still valid. The SSO token is issued by OAM after Oracle Enterprise Gateway authenticates to it on behalf of an end-user via the HTTP Basic or HTTP Digest filter. After successfully authenticating to OAM, the SSO token is stored in the `oracle.sso.token` message attribute.

Likewise, Oracle Enterprise Gateway's OAM Authorization filter is used to authorize an authenticated user for a particular resource against OAM (the user must first have been authenticated to OAM via

the HTTP Basic or HTTP Digest filter). Oracle Enterprise Gateway only leverages OAM for coarse-grained authorization.

Integration with Oracle Entitlements Server

Oracle Enterprise Gateway leverages Oracle Entitlements Server (OES) for fine-grained authorization.

Oracle Enterprise Gateway natively integrates with OES. Oracle Enterprise Gateway ships with an out-of-the-box adapter that calls OES Java Security Service Module (SSM) APIs. Oracle Enterprise Gateway automatically becomes a managed component in OES's centralized administration console. This means that all aspects of configuration, provisioning, and management of Oracle Enterprise Gateway's authorization policies can be seamlessly integrated with the rest of your enterprise using OES's attribute-based access control (ABAC) and resource management.

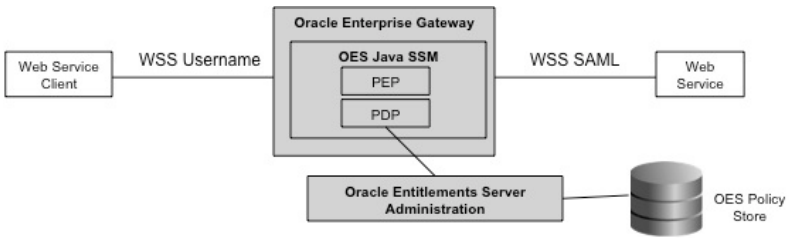

Figure 6: Oracle Enterprise Gateway and Oracle Entitlements Server

In a typical use case, OES can impose restrictions on who can invoke various SOAP web services and REST APIs. For example, OES can have a rule such as "a junior trader cannot participate in after-hours trading." When a trader tries to invoke a Buy or Sell operation on a stock after hours, Oracle Enterprise Gateway automatically calls into OES and performs an authorization policy check. Depending on OES's response, Oracle Enterprise Gateway either rejects or grants the request to the client. This model gives you the flexibility to integrate Oracle Enterprise Gateway authorization policies with existing SOA, Java EE, Spring, Microsoft .NET, and Microsoft SharePoint applications.

In addition, Oracle Enterprise Gateway supports the Extended Access Control Markup Language (XACML). This allows Oracle Enterprise Gateway to integrate with any XACML 2.0-compliant authorization system, including OES. (XACML is an industry standard that provides a policy model and an access control decision request / response protocol.)

Oracle Enterprise Gateway and the Cloud

Oracle Enterprise Gateway connects organizations to the Cloud. It manages all connections, including API keys that are vital for authentication to Cloud services, and it provides security, intrusion detection, and enhanced performance.

The PCI-Data Security standard and good corporate governance require that the API keys used to connect to Cloud services such as Amazon Web Services (AWS) should not be sitting on a hard drive or in the hands of a developer. Oracle Enterprise Gateway is designed to protect API keys.

Oracle Enterprise Gateway applies critical governance controls for service access, usage, and availability. It aggregates multi-domain services across their enterprise, partners, and third-party Cloud services such as Force.com Cloud Platform, and Google Apps.

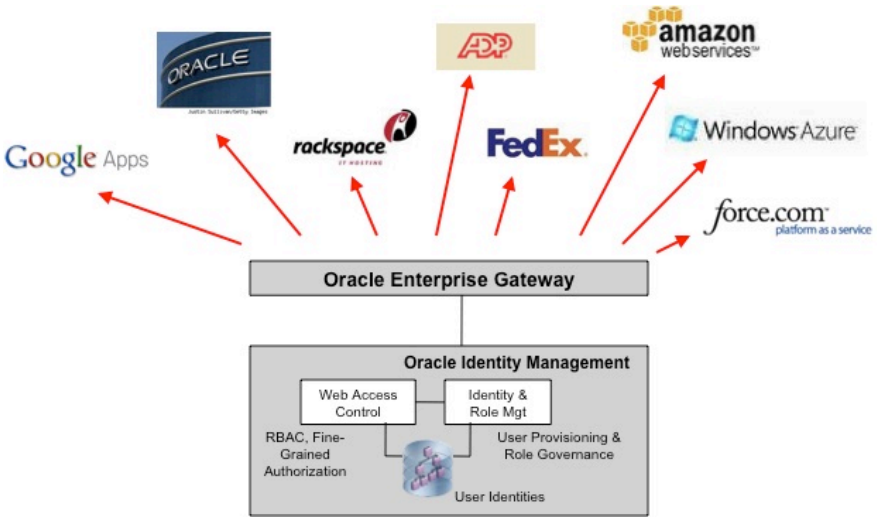

Figure 7: Oracle Enterprise Gateway and the Cloud

By bringing services together, Oracle Enterprise Gateway enables organizations to consistently define and manage the policies across these services and report on them.

Additionally, Cloud services involve connections that may be subject to delays and outages, thus increasing concerns about the reliability of the services and service providers themselves. Oracle Enterprise Gateway allows users to cache frequently accessed and transmitted data to mitigate the threat of Cloud network outages and provide a level of redundancy.

Oracle Enterprise Gateway provides the following benefits in Cloud computing deployments:

- *Mediate between public and private Cloud applications*: Achieve greater time and cost savings by using Oracle Enterprise Gateway to securely integrate local on-site applications with offsite Cloud services (e.g., storage services, sales force automation, and queuing mechanisms) in a hybrid Cloud environment. Oracle Enterprise Gateway also performs transformation and protocol mediation between on-site applications and the Cloud.
- *Out-of-the-box connectors to Cloud Services*: Avoid time-consuming and costly hard coding by enabling developers to drag and drop links to quickly hook on-site applications into cloud services. Oracle Enterprise Gateway offers a selection of pre-programmed Cloud services connections.

- *Safeguard and classify confidential data*: The same regulatory rules apply in the Cloud as outside, e.g., SOX, HIPPA, etc. As a result, companies need to enforce controls for privacy and data integrity policies when communicating through Cloud services. Oracle Enterprise Gateway enables selective encryption or removal of data according to sensitivity classifications.
- *Meter Cloud service usage*: With Oracle Enterprise Gateway, companies can control their data transfer to Cloud computing environments to avoid unwarranted usage levels and unanticipated bills. Oracle Enterprise Gateway provides autonomous metering of the Cloud services usage and hands control and visibility back to internal IT and finance teams.
- *Audit and archive interactions with Cloud services*: Oracle Enterprise Gateway provides full traceability and audit of every transaction conducted.

Conclusion

Oracle Enterprise Gateway is Oracle's software-based solution for securing the DMZ space. Oracle Enterprise Gateway provides security for SOAP and RESTful web services, intrusion detection, communication protocol and message format mediation, and XML acceleration.

Oracle Enterprise Gateway leverages the identity management infrastructure you already have, and it tightly integrates Oracle Fusion Middleware components, in particular Oracle Identity Management, and Oracle SOA Suite.

Oracle Enterprise Gateway
January 2011
Author: Marc Chanliau, Oracle Identity Mgt

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Oracle is committed to developing practices and products that help protect the environment

SOFTWARE. HARDWARE. COMPLETE.