An Oracle White Paper
June 2009

Securing Web Services and Service-Oriented Architectures with Oracle Web Services Manager 11*g*

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.

Oracle addresses web-services-based SOA security and management with Oracle Web Services Manager (WSM), a standards-compliant solution delivered as part of Oracle SOA Suite and Oracle Weblogic Server.

Oracle WSM allows companies to (1) centrally define and store declarative policies applied to the multiple web services making up a SOA infrastructure, (2) locally enforce security and management policies through configurable agents, and (3) monitor runtime security events such as failed authentication or authorization.

Oracle WSM provides business agility to respond to security threats and security breaches by allowing policy changes to be enforced in real time without the need to interrupt the running business processes.

Oracle WSM is a best-in-class web services security and management system that can be used by both developers at design time, and systems or security administrators in production environments.

"[Oracle] provides Oracle Web Services Manager with deep features for integrating with identity management software and propagating identity to underlying service platforms."

Randy Heffner, Forrester Research, Inc.

Web Services and SOA Infrastructures

The purpose of a SOA infrastructure is to allow *consumers* to invoke *services* exposed by *providers*.

In a business-to-business environment, a consumer may be a local procurement application located at Company XYZ invoking a remote purchase order processing service located at Company ABC. In an intranet environment, consumers and providers are part of the same company, thus facilitating the integration of heterogeneous applications into a more homogeneous corporate framework.

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 more about a web service than its location and the WSDL file that describes what it can do.

As shown in Figure 1, 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.

In the example presented in Figure 1, the web service consumer sends a request in the form of a SOAP message (SOAP is defined in the *Support for Industry Standards* section later in this document). The web service provider (www.xmethods.com) processes the request and returns the response, in this case the stock quote for Oracle.

Web services use XML documents and (mainly) the pervasive Hyper Text Transport Protocol (HTTP) to carry out transactions. This means that traditional network firewalls alone won't be enough to secure access to web services.

In the example shown in Figure 1, the web service provider may have asked for credentials to access the service, for example a username and a password. Also, the web service provider may have encrypted the response (the value of the stock).

In summary, web services are loosely coupled, distributed environments that allow companies to integrate heterogeneous applications within the enterprise or expose business functions to their customers and partners over the Internet.

Web services are characterized by three factors:

- What they do (the business functionality they expose).
- Where they are (the web site which exposes that functionality).
- How they can be accessed (the set of published interfaces necessary to use the exposed functionality).

Figure 1: Web service request and response

Web services rely on XML-based industry standards:

- A data format that allows uniform communication between web services consumers and web services providers (the Extensible Markup Language (XML) specification).
- A framework that describes XML vocabularies used in business transactions (XML Schema).
- An envelope used to send structured requests to, and receive structured responses from the web service provider (SOAP).
- A language that defines what a web service does (WSDL).
- A framework to publish and look up web services on the Internet (Universal Description, Discovery, and Integration -- UDDI).

Web Services Security

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 (more on this later in this document). 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).

In many cases, web services security solutions such as Oracle WSM rely on Public Key Infrastructure (PKI) environments. A PKI uses cryptographic keys (mathematical functions used

to encrypt or decrypt data). Keys can be private or public. In an asymmetric cipher model, the receiving party's public key is used to encrypt plaintext, and the receiving party's matching private key is used to decrypt the ciphertext. Also, a private key is used for encryption with digital signatures and the public key is used for verifying digital signatures. Public-key certificates (or certificates, for short) are used to guarantee the integrity of public keys.

Web services security requirements are supported by industry standards both at the transport level (Secure Socket Layer) 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 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. Please refer to the *Support for Industry Standards* section later in this document for more information about the specifications supported by Oracle WSM.

Web Services Security Requirements

What is required is (1) the use of transport security to protect the communication channel between the web service consumer and web service provider, and (2) message-level security to secure a transaction end-to-end through intermediaries (for example, several web services may be used to complete a single business transaction; in this case, security and identity information must be seamlessly passed across all the web services involved in the transaction).

Oracle WSM is designed to define and implement web services security in heterogeneous environments, including authentication, authorization, message encryption and decryption, signature generation and validation, and identity propagation across multiple web services used to complete a single transaction.

Oracle WSM allows the user to monitor web services consumers' activities at runtime. For example, the user is able to detect an abnormal level of failed authentications by consulting graphical charts.

Oracle WSM 11gR1 Functionality

Oracle WSM's purpose is to define and enforce security and reliability policies and provide auditing of runtime events. Following are simple examples of what you can typically do with Oracle WSM:

- Manage policies through a single administration console (Oracle Enterprise Manager).
- Attach policies to clients and service endpoints at both design time using Oracle JDeveloper and at runtime using Oracle Enterprise Manager.
- Advertise security requirements in WSDL and WS-MetadataExchange documents (please refer to *Support for Security Standards* later in this document).
- Perform impact analysis of policy changes before actually making the changes.
- Define authentication and/or authorization policies against an LDAP directory or identity infrastructures such as Oracle Access Manager.
- Generate standard security tokens to propagate identities across multiple web services used in a single transaction.
- Encrypt an element of the payload of a web service request, for example a credit card number.
- View access control events in graphical charts.
- Log request and response messages.

Principles

Oracle WSM allows you to externalize web services security and management from the applications you build. Instead of coding security logic in the application, you use Oracle WSM to implement declarative security and management through predefined policies.

Oracle WSM is based on three main operations: *Define*, *Enforce*, and *Monitor*.

- *Define* consists in attaching security and management policies to the web services to be protected. Examples of policies are Authenticate Request messages using username/password, Decrypt Messages using WS-Security, Sign Response messages.
- *Enforce* is the ability provided by Oracle WSM to distribute policies from a central Policy Manager to several policy enforcement points (PEP) or Agents that locally execute security and management policies at runtime.

- *Monitor* is the tracking (in graphical charts) of runtime security and management events captured by the Oracle WSM enforcement points (Oracle WSM Agents send information to Oracle Enterprise Manager which displays the information in actionable dashboards and charts).

The Pipeline Metaphor

Oracle WSM uses a pipeline metaphor (see Figure 2). Different categories of policies are executed in a predefined order for the request and the response messages. The order is also dependent on whether the policy is being executed at the client side or the service side. Figure 2 describes the order used for policy execution on the service side. On the client side, the policies are executed in the reverse order.

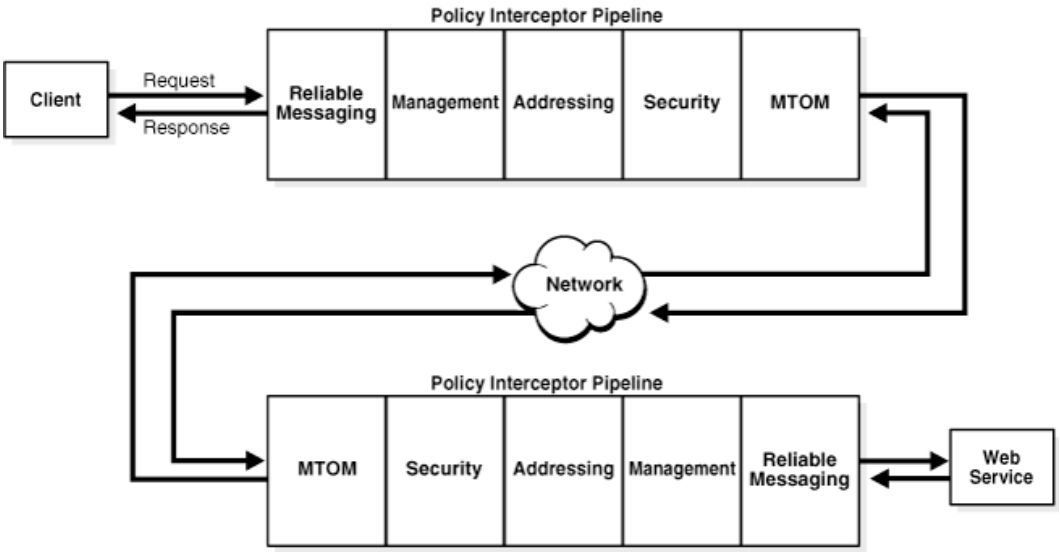

Figure 2: Oracle WSM's Policy Interceptor Pipeline

Typically, a web service client makes a request to a web service provider. The Oracle WSM Agent intercepts the request and executes the Request pipeline policies. If successful, the Agent forwards the request to the web service. The web service processes the request and sends a response to the web service client. The Agent intercepts the response and executes the Response pipeline policies. If successful, the Agent forwards the response back to the application server which then forwards it to the web service client.

Policy Assertions

Oracle WSM policies are made up of one or multiple policy assertions. For example, a security policy can be made up of two assertions: (1) a *Log* assertion and (2) a *WS-Security* assertion.

Policy assertions are executed in the order they are listed within a policy. In this case, the *Log* assertion is executed first (logging the request message to a log file), followed by the *WS-Security* assertion (authenticating the requester based on the token sent in the message and decrypting the message if the request is encrypted).

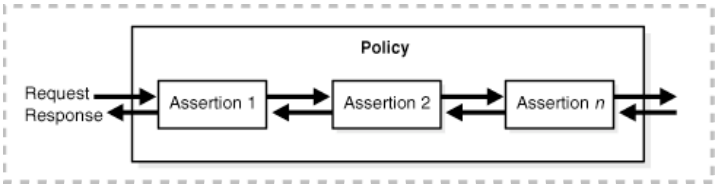

Figure 3: Policy Assertions

Policy Assertion Templates

Oracle WSM policy assertions are instances of policy assertion templates that are added to a policy at policy creation time. Oracle WSM ships with a predefined set of policy assertion templates. Additional templates can be created through Oracle Enterprise Manager to meet an organization's specific policy needs using built-in functionality or adding custom implementations.

Custom Policy Assertions

Oracle WSM allows users to define custom policy assertions that can be executed in a policy along with predefined policy assertions. Custom policy assertions are used when specific functionality is not provided out of the box such as support for non-standard security tokens. As previously mentioned, a policy assertion is an instance of a policy assertion template. To add a custom policy assertion to a policy, a custom policy assertion template needs to be added to Oracle WSM.

A custom policy assertion template is made up of two parts:

- Java archive (JAR) files that contain the compiled custom code and dependent libraries.
- An XML file that represents the custom policy assertion template and the configurable parameters.

Oracle WSM's documentation manual includes the application programming interfaces (API) and examples for developing custom policy assertion templates.

Once a custom policy assertion template is deployed, it becomes available to the Oracle WSM environment in the same way standard policy assertion templates are available.

Authenticating Requests

Oracle WSM security policies extract the security token from the request message based on the policy being applied to a service. The security token can be a username/password (extracted from the HTTP header, WS-Security header, or from the message body using XPath, a standard designed to navigate XML documents for queries); an X.509 certificate used for signing the request; a Kerberos ticket; a SAML token, or an Oracle Access Manager cookie token (extracted from a proprietary SOAP header).

After extracting the security token, Oracle WSM sends it to the Oracle Platform Security Services (OPSS) login module which validates the token, and then passes the credential information stored within the token to Weblogic Server's Authenticator (OPSS is Oracle Fusion Middleware's security layer; Oracle WSM consumes OPSS's services for authentication and authorization; please refer to the *Oracle Platform Security Services* technical white paper for more detail on OPSS).

The Weblogic Server Authenticator can be configured to validate credentials against a variety of identity stores such as LDAP (Oracle Internet Directory, Microsoft Active Directory, etc.), Oracle Access Manager, Oracle Database, CA SiteMinder. If successful, the Weblogic Server Authenticator creates a Java *Subject* and populates it with principals containing the username and roles associated with the authenticated user. The *Subject* is then made available to subsequent policy assertions and the web service itself.

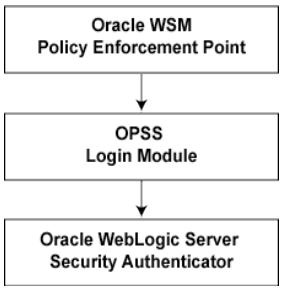

Figure 4: Request Authentication

Authorizing Requests

Oracle WSM provides two types of authorization policies: role-based, and permission-based. An authorization policy must be preceded by an authentication policy that sets the Java *Subject* containing the user roles used for authorization.

- A role-based authorization policy is configured to check if the user belongs to one of the configured roles in the policy.
- A permission-based authorization policy ensures that the *Subject* has the required Java permission to invoke the service. It is implemented by leveraging Oracle Platform Security Services to check if the authenticated *Subject* has been granted `oracle.wsm.security.WSFunctionPermission`. `WSFunctionPermission` can be granted to a user, a group, or an application role. If you grant `WSFunctionPermission` to a user or group it will apply to all the web services that are deployed in the same WebLogic Server domain.

Policy Management

The following sections describe how policies can be attached, enforced, and managed through the centralized Oracle WSM Policy Manager, and monitored and audited through Oracle Enterprise Manager (EM).

Policy Attachment

Oracle WSM provides two solutions for attaching policies to clients and services: Oracle JDeveloper and Oracle EM.

Application developers can attach Oracle WSM policies at design time within Oracle JDeveloper. In this case, only policy references (policy names) are attached to the web service. When the web service is deployed to an application server, the Oracle WSM Agent looks up the policy definition details from the Oracle WSM Policy Manager by providing the policy name as the key for lookup.

Administrators can use Oracle EM to attach (and detach) Oracle WSM policies in clients and services or make changes to policies already attached.

Policy Enforcement

An Oracle WSM Agent intercepts the requests to and responses from an application (client or service), and enforces the policies that are attached to requests and responses. The Agent looks up the policy definitions from the Oracle WSM Policy Manager and caches the policies to increase performance and prevent downtime should the Policy Manager become unavailable.

Oracle WSM supports dynamic policy updates. When the administrator changes policies to respond to higher level of security threats, the Policy Manager propagates the changes to the Agent which refreshes the policy cache and applies the changed policy immediately to the next request received.

Policy Governance

Oracle WSM provides policy governance through the centralized Policy Manager application that distributes policies and policy changes to all the Oracle WSM Agents within a Weblogic Server

domain. The Policy Manager leverages the Metadata Store (MDS) for reading and storing policies and policy attachment data used for policy impact analysis (see Figure 5).

In terms of policy governance Oracle WSM's architecture provides the following benefits:

- *Centralized visibility*: Customers can go to one console (Oracle EM) to view all the policies available, and determine the number of services a policy is attached to.
- *Policy reuse*: Policies can be reused by allowing the same policy to be applied to multiple clients or services.

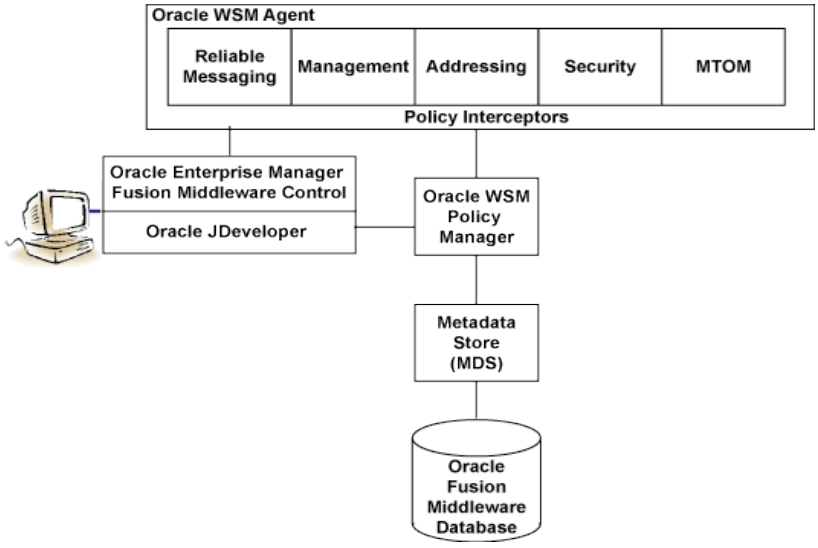

Figure 5: Oracle WSM Architecture

- *Impact analysis*: Before making a change to a policy, the administrator uses Oracle EM to view all the web services endpoints attached to that policy and evaluate the effect of the change on attached policies.
- *Policy versioning*: Oracle WSM maintains a history of all changes made to a given policy as separate versions viewable through Oracle EM. This is useful for audit purposes (who modified the policy and when, and what the change was). In addition, Oracle WSM allows rollback of a policy to an earlier version through Oracle EM in cases where the policy execution fails after changes are made to a policy.

Policy Monitoring

Oracle WSM collects monitoring statistics around policy execution such as number of authentication successes or failures. These monitoring statistics are available in Oracle EM. Alerts

and service-level agreement (SLA) rules can be applied on these statistics using Oracle EM SOA Management Pack for Oracle Fusion Middleware 11g when it becomes available.

Auditing

Oracle WSM is integrated with Oracle Fusion Middleware's Common Audit Framework (CAF) to audit policy enforcement results and policy creation, modification, and deletion.

This integration allows an administrator to provide predefined Oracle WSM audit reports through Oracle Business Intelligence Publisher.

Oracle WSM Ease of Use

Oracle WSM provides developers and administrators with enhanced ease of use.

- *Predefined policies*: Oracle WSM ships with a set of predefined policies based on standards and best practices. Customers can reuse these policies by attaching them directly to clients and service endpoints.
- *Policy advertisement in WSDL*: Clients (web service requesters) need to be aware of the type of policies used to secure the web service endpoint they invoke in order to make the appropriate request. Oracle WSM advertises a web service endpoint's security requirements in the web service's WSDL file using the WS-PolicyAttachment standard, thus eliminating the need for a web service provider to communicate security requirements to clients out of band (see the *Support for Industry Standards* section for more information on WS-PolicyAttachment).
- *Bulk policy attachment*: As a service provider, if you have standardized on a few policies to be applied to all your services, you can use Oracle WSM's bulk policy attachment feature to attach policies to multiple services at the same time using Oracle EM.
- *Client policy generation*: Oracle WSM allows the generation of a compatible client policy by providing the web service endpoint's WSDL as input. Oracle WSM parses the WS-Policy information in the WSDL file to generate the client policy, thus eliminating the need for the client's developers and administrators to understand the semantics of WS-Policy and its associated standards.
- *Policy compatibility checking*: This feature allows an administrator to check if the policy attached to a client is compatible with the web service endpoint being invoked before sending the first request.

Oracle WSM and Oracle Access Manager Integration

Oracle WSM can be used to authenticate and authorize access to web services against Oracle Access Manager (OAM). The benefit of such an integration allows customers to use OAM for access control to both web applications and web services and leverage value-added OAM features such as dynamic group functionality based on LDAP rules.

In this case, you need to first configure the OAM Authenticator in Oracle Weblogic Server. This allows authentication for all types of tokens to go against OAM. After successful authentication, the OAM Authenticator populates the Java *Subject* roles with all the groups that the authenticated user belongs to, including dynamic groups. Next, you need to define an Oracle WSM role-based authorization policy containing the allowed roles and attach it to the service to be protected. The authorization policy checks if any of the allowed roles are present in the authenticated user's Java Subject's role principals.

Oracle WSM's Role in SOA Governance

Oracle WSM is the runtime policy governance component for the Oracle SOA Governance solution. It provides production assurance for deployed SOA artifacts through policy-oriented security. It also participates at various stages of the closed-loop life cycle control, as shown in Figure 6.

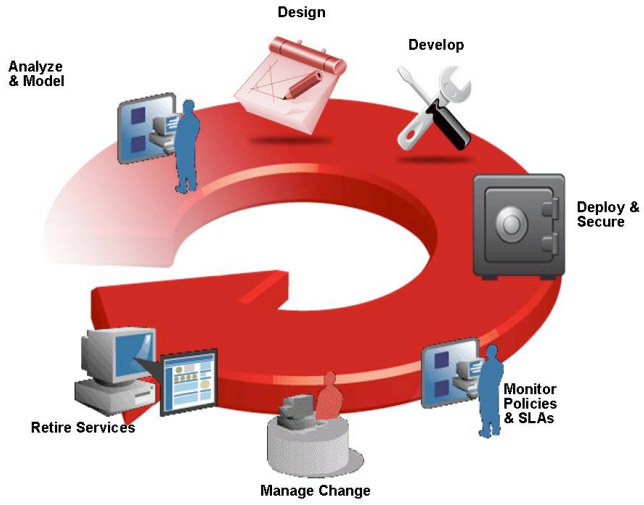

Figure 6: Oracle WSM's Role in SOA Governance

- As part of the service contract workflow, the service provider defines policy intents for the service that are agreed upon by the consumer.
- When the service is deployed, a workflow notification is sent to the security administrator containing policy intents. The administrator looks into the policy intents and applies appropriate Oracle WSM security policies to the deployed service.
- When enforcing policies, Oracle WSM collects metrics around policy execution that are available through graphical charts for runtime monitoring.

Oracle WSM Use Cases Summary

The following sections summarize typical Oracle WSM use cases.

Securing Access to Multiple Types of Web Services

Oracle WSM security and management policies can be applied to protect multiple types of web services: Java Platform, Enterprise Edition (Java EE) Java API for XML Web Services (JAX-WS), Oracle Application Development Framework (ADF) Business Components (BC), or Oracle SOA composites such as a Business Process Execution Language (BPEL) processes exposing a web service interface.

Securing Outbound Calls to a Web Service

An application can invoke a web service to retrieve or process data. If the web service is secured using WS-Security, Oracle WSM client policies can be applied to the service client application to secure the outbound message by inserting an identity token as requested by the service provider, and optionally signing and encrypting the request. If the service provider sends back an encrypted response, the Oracle WSM client policy can decrypt the response before forwarding the message back to the requesting application. The following service clients are supported by Oracle WSM 11gR1: Java EE JAX-WS, Oracle ADF-BC, Oracle SOA composites, and Oracle WebCenter remote portlets.

Propagating Identities from a Web Application to a Web Service

A web application may invoke a web service to retrieve or process data. If the web service is protected, the web application needs to send a request to the web service including a security token that the web service uses for authentication. If the web application itself is protected using an identity and access management solution such as Oracle Access Manager (OAM), the logged-in user identity can be propagated from OAM to the web service. In this case, the Oracle WSM client policy generates a SAML token from OAM's token information and inserts it into the WS-Security header of the outbound request message.

Propagating Identities through a Chain of Web Services

A web service may invoke another web service which in turn may invoke yet another web service to complete a single transaction (this pattern is known as "chained web services"). Each of the services in the chain may be protected. Instead of checking which service is calling which other service, Oracle WSM allows you to check who the original user invoking the chain of web services is. Oracle WSM policies can be used to propagate the original user's identity across the chained web services. Following successful authentication to the first web service in the chain, Oracle WSM sets the user as a Java *Subject* used throughout the transaction. When invoking another service, the Oracle WSM client policy picks up the user identity from the Java *Subject*, generates a SAML token based on the *Subject*'s information, and inserts the SAML token in the WS-Security header of the request message to be sent to the service provider. This allows all the

web services in a chain to track the identity of the actual user calling a web service endpoint instead of having the identity of the prior service in the chain calling the first web service to get that information.

Support for Industry Standards

The following table describes the various industry standards supported by Oracle WSM 11gR1.

STANDARD NAME	DESCRIPTION AND USE
SOAP 1.1 and 1.2	SOAP is an XML messaging standard used to format web service requests and responses (SOAP is not an acronym anymore but if it were, it could very well mean "Service-Oriented Architecture Protocol"). Security information is generally included in the SOAP message header and the message payload (for example a purchase order) is part of the SOAP message body. See Figure 1 in this document for an example of the structure of a SOAP message.
SOAP with Attachments (SWA) 1.1 and 1.2	SWA uses SOAP facilities and standard MIME mechanisms to carry and reference attachments to a SOAP message. MIME (Multipurpose Internet Mail Extensions) is a standard that supports messages with multiple parts (e.g., text, XML documents, pictures, etc.).
Message Transmission Optimization Mechanism (MTOM)	MTOM allows the sending of binary data to and from web services. MTOM is an efficient alternative to the MIME-based attachments used in the SWA mechanism described above.
WS-Security 1.0	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 and XML security tokens in WS-Security headers for authentication and authorization purposes. Oracle WSM 11gR1 supports the following WS-Security 1.0 security tokens: Username Token Profile 1.0; X.509 Token Profile 1.0; SAML Token Profile 1.0 (using SAML 1.1 assertions); SOAP with Attachments Profile 1.1.
WS-Security 1.1	Oracle WSM 11gR1 supports the following WS-Security 1.1 security tokens: Username Token Profile 1.1; X.509 Token Profile 1.1; SAML Token Profile 1.1 (using SAML 1.1 assertions); Kerberos Token Profile 1.1; SOAP with Attachments Profile 1.1.
WS-Policy 1.2	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 WSM 11gR1 and Oracle Fusion Middleware.
WS-SecurityPolicy 1.1	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.
WS-PolicyAttachment 1.1	WS-PolicyAttachment defines how (WS-Policy) policies are attached to web services. Policies can be bound to WSDL.
WS-ReliableMessaging (WS-RM) 1.0 and 1.1	WS-RM defines an XML framework for managing the reliable delivery of messages between web services endpoints. WS-RM is predicated on the SOAP messaging structure (SOAP binding) and relies on WS-Security, WS-Policy, and WS-Addressing to provide reliable messaging.
WS-Addressing 1.0	WS-Addressing provides an XML framework for identifying web services endpoints and for securing end-to-end endpoint identification in messages.
WS-MetadataExchange (WS-MEX) 1.1	WS-MetadataExchange defines how a client can request the metadata it needs to access and communicate with a web service endpoint (metadata can be WSDL or WS-Policy information). WS-MetadataExchange uses WS-Addressing to identify endpoints.
Encryption Algorithms AES-256, AES-192, AES-128, 3-DES	Encryption algorithms used with the WS-Security and XML Encryption standards.
Signature Algorithms RSA, SHA1	Digital signature algorithms used with the WS-Security and XML Signature standards.
Java Key Store (JKS)	JKS is the mechanism used by Oracle WSM 11gR1 to read and store key and certificate entries. JKS is part of Java's core API.

Conclusion

Oracle WSM is a standards-based solution that allows users (developers and system administrators) to implement web services security declaratively (no coding is required, and security is separate from the web services to be protected).

Web services security and management policies are defined centrally in Oracle WSM's Policy Manager, and executed locally at runtime through Oracle WSM's Agents, eliminating security silos.

Oracle WSM leverages standards-based identity management infrastructures to process authentication and access control operations.

White Paper Title
June 2009
Author: Vikas Jain
Contributor: Marc Chanliau

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

0109