OMA Web Services Enabler (OWSER): Core Specifications

Candidate Version 1.1 – 20 Dec 2005

Open Mobile Alliance
OMA-TS-OWSER_Core_Specification-V1_1-20051220-C

Contents

1. SCOPE 4
2. REFERENCES 5
2.1 NORMATIVE REFERENCES 5
2.2 INFORMATIVE REFERENCES 6
3. TERMINOLOGY AND CONVENTIONS 8
3.1 CONVENTIONS 8
3.2 DEFINITIONS 8
3.3 ABBREVIATIONS 9
4. INTRODUCTION 10
5. OMA MWS PROTOCOL ARCHITECTURE (INFORMATIVE) 11
6. SPECIFICATION OF COMMON SUPPORTING TECHNOLOGIES 14
6.1 XML 14
6.2 XML NAMESPACES 14
6.3 SOAP 14
6.4 WSDL 15
6.5 UDDI 15
6.6 HTTP 15
6.7 OTHER TRANSPORT BINDINGS 16
6.8 SOAP WITH ATTACHMENTS 16
6.9 USE OF WS-I PROFILES 16
7. SPECIFICATION OF COMMON FUNCTIONS OF THE OWSER 17
7.1 SECURITY 17
7.1.1 Security Services (informative) 18
7.1.2 Normative security technologies 22
7.2 PRIVACY MANAGEMENT FUNCTIONS (INFORMATIVE) 25
8. SERVICE MANAGEMENT FUNCTIONS 26
8.1 SERVICE REGISTRY FEATURES (INFORMATIVE) 26
8.1.1 Web Service Provider registration 26
8.1.2 Web Service publication 26
8.1.3 Web Service discovery 26
8.2 WEB SERVICE REGISTRY SPECIFICATIONS 27
8.2.1 Web Service registration 27
8.2.2 Web Service publication 27
8.2.3 Web Service discovery 27
APPENDIX A. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE) 28
APPENDIX B. DOCUMENT HISTORY (INFORMATIVE) 30
B.1 APPROVED VERSION HISTORY 30
DRAFT/CANDIDATE VERSION 1.1 HISTORY 30

Figures

Figure 1: The Web Services "stack" described by the OWSER 12

1. Scope

This document is part of a group of documents [OWSEROvw] [OWSERBP] that will be used to specify components of the OMA Web Services Enabler Release (OWSER). This document specifies the Web Service technologies that will be used to publish, discover and use these components in a secure, controlled and auditable manner.

The OMA Web Services Enabler (OWSER) defines the means by which OMA applications can be exposed, discovered and consumed using Web Services technologies. The purpose of the OWSER is to provide designers of Web Services within OMA with solutions to common functions using Web Services technologies. Without such a framework, designers of Web Services within OMA would almost certainly try to solve these problems on their own, each in their own way. Such an effort would inevitably lead to brittle implementations, interoperability problems, and increased time to market. These goals are consistent with those expressed in the OMA Service Enabler Strategy White Paper [OMASES].

The OMA Web Services Enabler (OWSER): Overview document [OWSEROvw] provides the rationale for using Web Services and an overview of the technologies to implement a set of identified common functions that are expected to be available to all OMA applications.

The OMA Web Services Enabler (OWSER) Best Practices: WSDL Style Guide [OWSERBP] provides non-normative information on the use of WSDL that may be used by OMA-defined Web Services.

This document provides the specifications of the components needed to provide the capabilities of the OWSER as identified in [OWSEROvw], in particular the normative use of Web Service technologies to implement such capabilities.

Ths document is intended to provide normative guidance to designers of specific OMA Web Services and implementers thereof. Therefore, material of a tutorial nature is kept to a minimum.

Not all components of the OWSER are specified in this version (1.0) of the document. This version specifies technologies that will enable applications acting in the role of Web Service Requesters (WSR) to access services offered by Web Service providers (WSP). No restriction is placed on the types of computing devices – mobile terminals, personal computers, servers etc. – that can be WSRs or WSPs, so long as they are capable of supporting the technologies specified in this document.

Future versions of this specification will provide additional functionality to complete the full range of capabilities offered by the OWSER.

2. References

2.1 Normative References

[IOPPROC] "OMA Interoperability Policy and Process, Version 1.1". Open Mobile Alliance™, OMA-IOP-Process-V1_1, URL:http://www.openmobilealliance.org/

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels". IETF RFC 2119, S. Bradner. March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt

[HTTP1.0] "Hypertext Transfer Protocol -- HTTP/1.0", IETF RFC 1945, Fielding, R., Frystyk, H. and T. Berners-Lee, May 1996, URL:http://www.ietf.org/rfc/rfc1945.txt

[RFC2616] "Hypertext Transfer Protocol – HTTP/1.1", IETF RFC 2616, R. Fielding, J. Gettys, J. Mogul, H. Frystyk Nielsen, L. Masinter, P. Leach, T. Berners-Lee, June 1999, URL:http://www.ietf.org/rfc/rfc2616.txt

[RFC2145] "Use and Interpretation of HTTP Version Numbers", Mogul, J.C., Fielding, R., Gettys, J., Frystyk Neilsen, H., RFC 2145, May 1997, URL:http://www.ietf.org/rfc/rfc2145.txt

[SOAP1.1] "Simple Object Access Protocol (SOAP) 1.1", Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Nielsen, Satish Thatte, Dave Winer, W3C Note, May 8, 2000, URL:http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[WSDL1.1] "Web Services Description Language (WSDL) 1.1", Erik Christensen, Francisco Cubrera, Greg Meredith, Sanjiva Weeravarana, W3C NOTE, March 15, 2001, URL:http://www.w3.org/TR/wsdl.html

[WSI1.0] "Basic Profile Version 1.0", Web Services Interoperability Organization, Final Material, 2004/04/16, URL: http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html

[XMLNS] "Namespaces in XML", Tim Bray, Dave Hollander, Andrew Layman, W3C Recommendation, 14 January 1999, URL:http://www.w3.org/TR/REC-xml-names/

[XML1.0] "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C Recommendation, 6 October 2000, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, URL:http://www.w3.org/TR/REC-xml

[SSL3.0] "The SSL Protocol Version 3.0", URL:http://wp.netscape.com/eng/ssl3/draft302.txt

[RFC2818] "HTTP Over TLS", E. Rescorla, URL:http://www.ietf.org/rfc/rfc2818

[RFC2246] "The TLS Protocol Version, 1.0." T. Dierks, C. Allen, IETF RFC 2246 January 1999, URL:http://www.ietf.org/rfc/rfc2246.txt

[XMLCanon] "Canonical XML, Version 1.0", W3C Recommendation, 15 March 2001, URL:http://www.w3.org/TR/xml-c14n/

[XML-XCanon] "Exclusive XML Canonicalization, Version 1.0", John Boyer, Donald E. Eastlake 3rd, Joseph Reagle, W3C Recommendation, 18 July 2002, URL:http://www.w3.org/TR/xml-exc-c14n/

[XML-ENC] "XML Encryption Syntax and Processing", W3C Recommendation, 10 December 2002, URL:http://www.w3.org/TR/xmlenc-core/

[XML-SIG] "XML-Signature Syntax and Processing", D. Eastlake, J. R., D. Solo, M. Bartel, J. Boyer, B. Fox, E. Simon, W3C Recommendation, 12 february 2002, URL: http://www.w3.org/TR/xmldsig-core/

[UDDI] "UDDI Version 2.04 API Specification," UDDI Committee Specification, 19 July 2002, URL:http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm

[UDDIData] "UDDI Version 2.03 Data Structure Reference", UDDI Committee Specification, 19 July 2002, URL:http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.htm

[SAML1.0] "Assertions and Protocol for the Oasis Security and Assertions Markup Language (SAML)", OASIS Standard 5 November 2002, URL:http://www.oasis-open.org/committees/download.php/1371/oasis-sstc-saml-core-1.0.pdf

[SAMLConf] "Conformance Program Specification for the OASIS Security Assertions Markup Language", OASIS Standard, November 5, 2002, URL:http://www.oasis-open.org/committees/download.php/1374/oasis-sstc-saml-conform-1.0.pdf

[SAMLBind] "Bindings and Profiles for the Oasis Security and Assertions Markup Language (SAML)", Oasis Standard 5 November 2002, URL:http://www.oasis-open.org/committees/download.php/1372/oasis-sstc-saml-bindings-1.0.pdf

[SOAPwAtt] "SOAP Messages with Attachments," W3C Note, 11 December 2000, URL:http://www.w3.org/TR/SOAP-attachments

[RFC2396] "Uniform Resource Identifiers (URI): Generic Syntax", IETF RFC 2396, T. Berners-Lee, R. Fielding, L. Masinter, August 1998, URL:http://www.ietf.org/rfc/rfc2396.txt

[RFC2104] "HMAC: Keyed Hashing for Message Authentication", Krawczyk, H., Bellare, M. and R. Canetti, IETF RFC 2104, February 1997. URL:http://www.ietf.org/rfc/rfc2104.txt

[WS-SEC] "Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)", OASIS Standard 200401, March 2004, URL:http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf

[WSS-X509] "Web Services Security X509 Certificate Token Profile", OASIS Standard 200401, March 2004, URL:http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0.pdf

[WSS-Username] "Web Services Security Username Token Profile 1.0", OASIS Standard 200401 , March 2004, URL:http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf

2.2 Informative References

[WSDL1.2] "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language", R. Chinnici, M. Gudgin, J-J. Moreau, S. Weerawarana , W3C Working Group Draft, URL:http://www.w3.org/TR/wsdl20/

"Web Services Description Language (WSDL) Version 2.0 Part 2: Message Patterns", M. Gudgin, A. Lewis, J. Schlimmer, W3C Working Group Draft, URL:http://www.w3.org/TR/wsdl20-patterns/

"Web Services Description Language (WSDL) Version 1.2 Part 3: Bindings", J-J. Moreau, J. Schlimmer, W3C Working Group Draft, URL:http://www.w3.org/TR/wsdl12-bindings/

[XMLInfoSet] W3C Recommendation "XML Information Set", J. Cowan, R. Tobin, October 2001, URL:http://www.w3.org/TR/2001/REC-xml-infoset-20011024/

[SAML-Security] "Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML)", Committee Specification 01, 31 May 2002, URL:http://www.oasis-open.org/committees/download.php/1375/oasis-sstc-saml-sec-consider-1.0.pdf

[RFC2234] "Augmented BNF for Syntax Specifications: ABNF". IETF RFC 2234, D. Crocker, Ed., P. Overell, November 1997, URL:http://www.ietf.org/rfc/rfc2234.txt

[P3P] "The Platform for Privacy Preferences 1.0 (P3P1.0) Specification", Lorrie Cranor, Marc Langheinrich, Massimo Marchiori, Martin Presler-Marshall, Joseph Reagle, W3C Recommendation 16 April 2002, URL:http://www.w3.org/TR/2002/REC-P3P-20020416/

[DOS] "DOS-resistant Authentication with Client Puzzles", Tuomas Aura, Pekka Nikander and Jussipekka Leiwo, URL:http://research.microsoft.com/users/tuomaura/Publications/aura-nikander-leiwo-protocols00.pdf

[MIME] "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", IETF RFC 2045, November 1996, URL:http://www.ietf.org/rfc/rfc2045.txt

[OWSEROvw] "OMA Web Services Enabler (OWSER): Overview", Version 1.0, Open Mobile Alliance™, OMA-OWSER-Overview-V1_0, URL:http://www.openmobilealliance.org/

[OWSERBP] "OMA Web Services Enabler (OWSER) Best Practices: WSDL Style Guide", Version 1.0, Open Mobile Alliance™, OMA-OWSER-Best_Practice-WSDL_Style_Guide-V1_0, URL:http://www.openmobilealliance.org/

[SOAP1.2] "SOAP Version 1.2 Part 0: Primer", Nilo Mitra, W3C Recommendation, 24 June 2003, URL:http://www.w3c.org/TR/2003/REC-soap12-part0-20030624/

"SOAP Version 1.2 Part 1: Messaging Framework", Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen, W3C Recommendation, 24 June 2003, URL:http://www.w3c.org/TR/2003/REC-soap12-part1-20030624/

"SOAP Version 1.2 Part 2: Adjuncts", Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen, W3C Recommendation, 24 June 2003, URL:http://www.w3c.org/TR/2003/REC-soap12-part2-20030624/

[STG] "Security Taxonomy and Glossary", L. Wheeler, URL:http://www.garlic.com/~lynn/secure.htm

[WSGloss] "Web Services Glossary", W3C Working Draft , 8 August 2003, URL:http://www.w3.org/TR/2003/WD-ws-gloss-20030808/

[21CFR11] 21 CFR Part 11. Electronic Records; Electronic Signatures; Final Rule Electronic Submissions; Establishment of Public Docket; Notice, URL:http://21cfrpart11.com/files/library/government/21cfrpart11_final_rule.pdf

[OCSP] "X.509 Internet Public Key Infrastructure – Online Certificate Status Protocol - OCSP," M. Myers, et al., IETF RFC 2560, June 1999, URL:http://ietf.org/rfc/rfc2560.txt

[RFC2828] "Internet Security Glossary", R. Shirley, IETF RFC 2828, May 2000, URL:http://www.ietf.org/rfc/rfc2828.txt .

[WebArch] "Web Services Architecture Requirements", W3C Working Draft, Aug 8, 2003, URL:http://www.w3.org/TR/2003/WD-ws-arch-20030808/

[X800] CCITT Recommendation X.800 (1991), "Security architecture for Open Systems Interconnection for CCITT applications", URL:http://online.vsi.ru/library/ITU-T/original/recs/x/x.0800e.zip

[OCSPProf] "Online Certificate Status Protocol Mobile Profile, Version 1.0", Open Mobile Alliance™, OMA-WAP-OCSP-200211210-d, URL:http://www.openmobilealliance.org/

[OMADict] "Dictionary for OMA Specifications Version 1.0", Open Mobile Alliance™, OMA-Dictionary-V1_0, URL:http://www.openmobilealliance.org

[XACML] "eXtensible Access Markup Language (XACML) Version 1.0", OASIS Standard, 18 February 2003, URL:http://www.oasis-open.org/committees/download.php/2406/oasis-xacml-1.0.pdf

[OMASES] "OMA Service Enabler Strategy White Paper, Version 1.1", Open Mobile Alliance™, OMA-WP-SvcEnablerStrat-V1_1, URL:http://www.openmobilealliance.org/

[RFC3280] "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", R. Housley, W. Ford, W. Polk, D. Solo, IETF RFC 3280, April 2002 URL:http://www.ietf.org/rfc/rfc3280.txt

[WSS-SAML] "Web Services Security: SAML Token Profile", OASIS Working Draft, URL: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

[WSS-Kerberos] "Web Services Security Kerberos Token Profile", OASIS Working Draft, URL:http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

[XKMS] "XML Key Management Specifications (XKMS 2.0)", W3C Candidate Recommendation, 5 April 2004, URL:http://www.w3.org/TR/xkms2/

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

Examples and diagrams are always informative, unless there are special circumstances when they are used to illustrate some normative usage, in which such examples or diagrams will explicitly be marked as normative.

Clarifications intended to augment some text are shown as follows:

> **NOTE:** This is an example of a note.

Such notes are always informative.

Text with nouns in capitals (e.g., Web Service, Policy etc.) in this document is used as defined in section 3.2.

3.2 Definitions

Access	To interact with a system entity in order to manipulate, use, gain knowledge of, and/or obtain a representation of some or all of a system entity's resources. (Source: [RFC2828])
Access Control	Protection of resources against unauthorized access; a process by which use of resources is regulated according to a security policy and is permitted by only authorized entities (users, programs, processes, or other systems) according to that policy. (Source: [RFC2828])
Anonymity	The quality or state of being anonymous, which is the condition of having a name or identity that is unknown or concealed. (Source: [RFC2828])
Authentication	The process of verifying an identity claimed by or for a system entity. (Source: [RFC2828])
Authorization	The process of determining, by evaluating applicable access control information, whether a subject is allowed to have the specified types of access to a particular resource. Usually, authorization is in the context of authentication. Once a subject is authenticated, it may be authorized to perform different types of access. (Source: [STG])
Data Confidentiality	The property that information is not made available or disclosed to unauthorized individuals, entities, or processes [i.e., to any unauthorized system entity]. ISDs SHOUL NOT use this term as a synonym for “privacy”, which is a different concept. (Source: [RFC2828])
Data Integrity	The property that data has not been changed, destroyed or lost in an unauthorized or accidental manner. [RFC2828]
Denial of Service	The prevention of authorized access to a system resource or the delaying of system operations and functions. [RFC2828]
Identity	The unique identifier for a person, organization, resource or service (Source: [WSGloss])
Key Management	The process of handling and controlling cryptographic keys and related material (such as initialization values) during their life cycle in a cryptographic system, including ordering, generating, distributing, storing, loading, escrowing, archiving, auditing, and destroying the material. [RFC2828]
Non-repudiation	The prevention of denying by one of the entities involved in a communication of having participated in all or part of the communication. (Source: [X800])
Privacy	The right of an entity (normally a person), acting on its won behalf, to determine the degree to which it will interact with its environment, including the degree to which the entity is willing to share information about itself with others. ISDs SHOULD NOT use this term as a synonym for “data confidentiality” or “data confidentiality service”, which are different concepts. Privacy is a reason for security rather than a kind of security. [RFC2828]
Policy	A set of statements or rules that describes a set of controls which determine an action.

Security Policy	A set of rules and practices that specify or regulate how a system or organization provides security services to protect sensitive and critical system resources. [RFC2828]
Web Service	A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. (Source: [WSGloss])
Web Service Enabler Release	The OMA Web Service Enabler Release is a set of common protocols, schemas, and processing rules using Web Service technologies that are the elements that can be used to create or interact with a number of different services.
Web Service Provider	An organizational entity that exposes some capability as a Web Service.
Web Service Requester	A software system that requests a Web Service.

3.3 Abbreviations

CRL	Certificate Revocation List
HMAC	Hashed Message Authentication Code
HTTP	Hyper Text Transfer Protocol
HTTPS	HTTP Secure (aka HTTP over SSL)
MAC	Message Authentication Code
MIME	Multipurpose Internet Mail Extensions
OASIS	Organization for the Advancement of Structured Information Standards
OCSP	Online Certificate Status Protocol
OMA	Open Mobile Alliance
OWSER	OMA Web Services Enabler
PKCS	Public Key Cryptography Standards
PKI	Public Key Infrastructure
SAML	Security Assertion Markup Language
SHA	Secure Hash Algorithm
SOAP	Simple Object Access Protocol[1]
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
URI	Uniform Resource Identifier
WAP	Wireless Application Protocol
WS	Web Services
WS-I	Web Service Interoperability Forum
WSDL	Web Service Description Language
WSP	Web Service Provider
WSR	Web Service Requester
XKMS	XML Key Management Service

[1] Note that starting from SOAP Version 1.2, SOAP will no longer be an acronym.

4. Introduction

This document describes the instantiation of the OWSER in terms of the normative use of Web Service and other supporting technologies to implement the functionalities identified in [OWSEROvw]. The [OWSEROvw] outlines key concepts for understanding this document.

Ths document is intended to provide normative guidance to designers of specific OMA Web Services and implementers thereof. Therefore, material of a tutorial nature is kept to a minimum.

Section 5 of this document describes the conceptual "stack" of technology specifications that together comprise the technical architecture of Web Services some of which will be used in this version of the OWSER.

Section 6 of this document provides the specifications of the supporting technologies that will form the common basis for all OMA Web Services. It specifies the basic set of technologies on which a service-oriented architecture based on Web Services are based, namely HTTP for message transfer, XML for data description, SOAP for message encapsulation, WSDL for service interface description, and UDDI for service discovery. Where options are provided by the referenced specification, this specification mandates certain choices to ensure interoperability for OMA applications.

Section 7 of this document provides the specifications for common functions identified in the OWSER.

- Section 7.1 discusses and provides specifications for common security features to be used by applications using the OWSER.
- Section 7.2 discusses and provides specifications for the management of end user privacy.

Section 8 of this document provides specifications for the management of a Web Service during its lifecycle.

Appendix A provides the static conformance requirements to this specification.

5. OMA MWS Protocol Architecture (informative)

The task of an OMA application designer is to understand not only a particular application domain, but also those parts of the OMA infrastructure that can be leveraged to reuse certain capabilities that are expected to be available in common to all OMA applications. To this end, the OWSER identifies such common capabilities that will be available to the designer as a part of the OWSER based on the analysis of a set of requirements and use cases. The OWSER addresses the following capabilities in the first release of this specification [OWSEROvw]:

- WSR authentication
- WSR authorization
- Message and data confidentiality
- Message and data integrity
- Service Registry Management
    - WSP registration
    - Web Service publication
    - Web Service discovery
    - Web Service binding

The functionality and use of these OWSER capabilities will be defined by a variety of specifications. Some of these specifications have been or are being developed in fora other than the OMA, while others may be developed within the OMA. Together, these specifications will provide the complete toolbox for the designer of an application that leverages the OWSER to use or provide Web Services in an OMA environment.

It is important for an application designer to understand the inter-relationship between these specifications, in particular the external specifications that form much of the basis for interoperable Web Services. Such specifications have been arrived at in the larger community with input from all major industry segments. Their adoption forms a stable base that will be widely implemented in a variety of platforms and tools, and which are expected to be available to and used by application designers, implementers and service providers.

Figure 1 shows a "stack" identifying the various specifications that offer the basis for interoperable Web Services. It is, in a sense, a protocol architecture for the OWSER showing how various Web Service specifications, whose use have been described normatively in subsequent sections of this document, inter-relate.

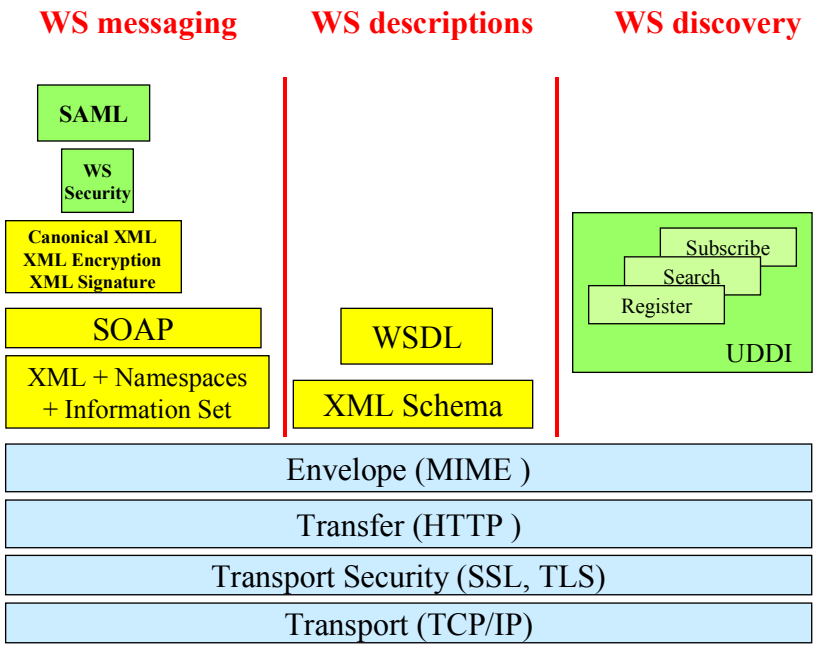

Figure 1: The Web Services "stack" described by the OWSER

The Web Services stack has been divided into three portions, representing the three aspects of Web Service interactions - discovery of service descriptions, the service description techniques, and the messaging to access services that form the basis of the service oriented architecture described in the OWSER Overview [OWSEROvw].

Underlying all three aspects are the data transport mechanisms. These include the means by which Web Service messages may be enveloped in various application layer constructs (e.g., MIME), or the means by which Web Service messages may be transferred as payload of other application layer protocols (e.g., HTTP). Note that it is NOT necessary that both the envelope and transfer mechanism always be used.

It is assumed that TCP/IP provides the transport mechanism for all upper layer data. At the application layer, various enveloping mechanisms are possible for application payload, including those for Web Service messages. One technique, namely MIME [MIME] which is the most widely deployed, is shown in the figure, although this choice does not preclude the use of other techniques. Specifically, a MIME/ multipart construct may be used to carry a soap:Envelope together with "attachments" referenced from within the envelope [SOAPwAtt]. This version of the OWSER mandates the use of [SOAPwAtt] for encapsulation of documents bundled with a SOAP 1.1 message, but it should be noted that future version might move to the output of the ongoing work in the W3C to standardize a "concrete" attachments specification.

Web Service messaging is most prevalent using HTTP as a transfer mechanism, although the messaging specification is defined to be transport neutral. It is also possible (although not provided in this version of the OWSER Specifications) to use other transfer mechanisms.

Moving to those aspects specific to Web Services, as opposed to any application layer protocol, Web Service messages are described using XML [XML1.0] (which includes the XML Namespace [XMLNS] and XML Information Set [XMLInfoSet] recommendations from the W3C). The actual messaging framework is provided by the SOAP specification [SOAP1.1] [SOAP1.2], which provides an extensible message framework comprising an overall envelope containing an optional header and a mandatory body. This version of the OWSER specifications mandates the use of [SOAP1.1] as the messaging envelope (see section 6.3), but it is expected that future versions may move to the W3C Recommendation, [SOAP1.2], when there is widespread industry support for it in platforms and development tools.

The SOAP header element forms the basis for message-level extensibility as it allows other specifications to define additional functionality (e.g., security, etc.) that may be carried as a part of the header to provide value-added services. In Figure 1, the

boxes on top of SOAP represent those specifications that the OWSER has identified to provide security-related, value-added functionality. This version of the OWSER specification will define the normative security headers that will convey security-related data needed to provide the common security capabilities of authentication, authorization, data confidentiality and integrity.

For example, referring to Figure 1, it is very likely that a Web Service request will require authorization for the interaction. SAML is one type of authorization assertion that can be used and it can be encapsulated within a WS-Security header as specified in the SAML Profile [WSS-SAML]. Similarly, there is a need to encrypt or digitally sign parts of a SOAP message. The XML Encryption [XML-ENC] and XML Digital Signature [XML-SIG] specifications provide the mechanisms to do so.

In order to make security techniques such as signing of messages with XML content operate properly, it is necessary to use XML Canonicalization [XMLCanon]. This is because changes in the XML serialization (e.g., string representation) may be introduced by different XML implementations, even when retaining the same XML information content. Signing requires a canonical form so that verification by another party can succeed. Potential changes include, for example, spacing or attribute ordering. Exclusive XML Canonicalization [XML-XCanon] is necessary to handle XML Namespaces when portions of XML content are moved from one XML context (e.g., a message) to another, and is the preferred form of XML canonicalization. So, in a sense, moving up the "stack", each specification provides support for certain features needed by those above.

Moving to the description aspect of Web Services, WSDL is the means to offer descriptions of Web Service interfaces. The data exchanged over such an interface is described using XML Schema. Future specifications are expected in this area that will offer the ability to describe how Web Services may be chained together or perform their activities in a coordinated manner. This version of the OWSER specifications mandates the use of [WSDL1.1] (see section 6.4), but it is expected that future versions may move to the W3C Recommendation, [WSDL1.2], when there is widespread industry support for it in development tools.

A set of specifications is available for the discovery aspects of Web Services, namely the UDDI specifications [UDDI] that cover the whole gamut of how Web Services may be published and discovered in a secure manner, and how service registries hosting such service descriptions may be federated.

6. Specification of common supporting technologies

This section provides specifications for the common supporting capabilities that must be available to access and use Web Services. This provides the common infrastructure for OMA Web Service based interactions. Referring to the protocol "stack" in Figure 1, this section provides specifications for Web Service message transfer, data description, message construct, service interface descriptions and, where applicable, service registration and discovery.

6.1 XML

The OWSER will use XML-based messages for communications between various entities. Conforming specifications of such messages MUST be based on [XML1.0].

[XML1.0] allows for the use of UTF-8 and UTF-16 encodings. A conformant sending implementation MAY choose either of these encodings, but a receiver MUST be capable of accepting both.

[XML1.0] allows UTF-8 encoding to include a byte order mark (BOM); therefore, receivers of messages MUST be prepared to accept them.

6.2 XML Namespaces

The XML namespace URIs that MUST be used by implementations of this specification are as follows:

http://www.openmobilealliance.org/schema/mws/
http://www.openmobilealliance.org/schema/common

> **NOTE:** The namespace conventions shown above are a placeholder until such time as the work on naming conventions in the OMA is complete. In particular, it is expected that the work on such matters will include both the definition of a top-level OMA namespace as well as a versioning convention.

The following namespaces are used in this document:

Prefix	Namespace
xenc	http://www.w3.org/2001/04/xmlenc#
ds	http://www.w3.org/2000/09/xmldsig#
wsse	http://schemas.xmlsoap.org/ws/2003/06/secext
soap	http://schemas.xmlsoap.org/soap/envelope/
uddi	urn:uddi-org:api_v2

Note that the use of any namespace prefix is arbitrary and not semantically significant.

6.3 SOAP

SOAP provides the basic message format for communications between a Web Service and a WSR. It defines the structure of the overall soap:Envelope, which contains an optional soap:Header and a mandatory soap:Body.

OWSER-based Web Service interactions MUST use [SOAP1.1] for Web Service messages as constrained by the WS-I Basic Profile 1.0 [WSI1.0] (see specifically section 4 therein).

> **NOTE:** Future versions of this specification may require the use of the forthcoming W3C Recommendation SOAP Version 1.2 [SOAP1.2]. Note that there are some syntactic and semantic incompatibilities between the two versions. [WSI1.0] has attempted to remove some of the ambiguities in the specification of [SOAP1.1]. It is expected that this profile will have widespread support from implementers in the near future.

> **NOTE:** The WS-I 1.0 profile of SOAP 1.1 requires the serialization of a SOAP message as a well-formed XML 1.0 document. In future, if the use of SOAP 1.2 is mandated, it may be possible to provide alternative serializations for SOAP messages.

See also section 6.9 on the use of WS-I profiles.

6.4 WSDL

The Web Service Description Language (WSDL) provides the description of a Web Service as a set of end points operating on SOAP messages containing either document-oriented or procedure-oriented information.

OWSER-based specifications SHOULD provide their service descriptions using WSDL 1.1 as constrained by the WS-I Basic Profile 1.0.

> **NOTE:** It is recognised that WSDL does not provide a complete description of all the additional information needed to fully describe a Web Service. For instance, it is not currently possible in WSDL to describe aspects of the security context, and other policies governing the interactions between a WSR and a Web Service. Work on additional specifications in these areas is progressing in other standards bodies, and may be incorporated into future releases of the OWSER.
>
> **NOTE:** Future versions of this specification may require the use of the forthcoming W3C Recommendation WSDL Version 1.2 [WSDL1.2]. Note that there are some syntactic and semantic incompatibilities between the two versions. [WSI1.0] has attempted to remove some of the ambiguities in the specification of [WSDL1.1]. It is expected that this profile will have widespread support from implementers in the near future.

See also section 6.9 on the use of WS-I profiles.

6.5 UDDI

The UDDI specifications provide Web Service interfaces towards the publication of a Web Service interface by a WSP, and discovery of such interfaces by potential users. OWSER based Web Services MAY use UDDI for the registration and publication of Web Services. Note that the use of UDDI is not necessary in every instance of Web Service interactions. There may be many instances where the additional information needed for WSRs to interact with WSPs may be obtained by other means. However, when there is a need for mechanisms for distributed publication and discovery, this specification mandates the use of UDDI as the selected mechanism.

In particular, if Web Service interface description publication and discovery mechanisms are required, OWSER-based WSRs and WSPs MUST use the UDDI 2.0X series of specifications as profiled by the WS-I Basic Profile 1.0 [WSI1.0] (see specifically section 6 therein).

See also section 6.9 on the use of WS-I profiles.

See also section 8.2 on the normative use of the various UDDI specifications for the different aspects of service management.

6.6 HTTP

The Web Services messaging protocol, SOAP, has been defined to be transport neutral. As a result it is possible to use many standard or non-standard transport mechanisms. However, this version of the OWSER focuses on HTTP, as this is the most prevalent transfer mechanism available for Web Service messaging, and also one that is defined as a part of the [WSI1.0] Basic Profile.

This specification requires that a SOAP message MUST be sent using either HTTP/1.1 or HTTP/1.0. It SHOULD be sent using HTTP/1.1 [RFC2616].

> **NOTE:** HTTP/1.1 has several performance advantages over HTTP/1.0. Also, support for HTTP/1.0 is implied in HTTP/1.1. For more information on HTTP versioning, see [RFC2145].
>
> **NOTE:** As noted in section 5.6.2 of [WSI1.0], the use of HTTPS is not prohibited.

This specification requires that the SOAP binding to HTTP MUST conform to the constraints laid out in section 4.2 of [WSI1.0].

6.7 Other transport bindings

The use of other transport bindings for SOAP messages in an OMA environment is not excluded. However, no transport bindings other than as in section 6.6 are specified in this version of the OWSER specification. If and when other transport bindings are provided, their specifications will be referenced or described in a future version of this specification.

6.8 SOAP with Attachments

When there is a need to bundle documents referenced from within a soap:Envelope for transport as a compound structure, OMA Web Service based enablers SHOULD use SOAP with Attachments [SOAPwAtt].

> **NOTE:** The WS-I.org is working on a profile for [SOAPwAtt].
>
> **NOTE:** The W3C is working on standardizing an attachments specification. Future versions of this specification may conform to such a specification.

6.9 Use of WS-I profiles

The Web Services Interoperability Forum (WS-I) Basic Profile 1.0 [WSI1.0] provides clarifications, resolution of ambiguities and the creation of profiles for a set of base specifications that are necessary to promote interoperability. In particular, it shows how implementations of SOAP1.1, WSDL1.1 and UDDI2.0 conforming to the stated profile may be used together in an interoperable manner. The forum is also expected to produce test material to verify Web Service compliance with the Basic Profile.

By restricting the use of these three base technologies in OMA to the profile specified in [WSI1.0], it is expected that interoperability at this level will be assured.

> **NOTE:** The MWS Working Group is aware of work that is starting in WS-I.org on a Basic Security Profile, and will explore its applicability to the OWSER at some future date after that work has reached some level of maturity.

7. Specification of common functions of the OWSER

Section 5 of this specification provided a protocol architecture showing the relationships between the principal specifications that will be referenced and used by the OWSER. Section 6 provided the common infrastructure that must be available to all participants in Web Service based interactions. This section provides the specification of the common capabilities, categorised into areas related to security, service management etc. Consulting Figure 1 in section 5, such specifications correspond to those that reuse the basic Web Service messaging and description infrastructures provided by SOAP and WSDL.

7.1 Security

This section provides an informative sub-section describing security services and a normative sub-section for technologies that may be used to implement such security services. This specification incorporates existing standards by reference, and adds additional profiling where needed. In some cases, the referenced standards are drafts, subject to change. They are referenced because of high maturity or acceptance, with the knowledge that they will change before final release.

The following list the goals and non-goals for providing security specifications for the OWSER.

Goals:

- Specify Web Services security technology suitable for heterogeneous infrastructure environments including Web Service nodes that are limited in processing power and network access, as well as nodes that are not restricted by such factors.

  NOTE: This version of the specification does not specifically address the requirements of Web Service nodes that have such limiting factors.

- Use existing and emerging XML and Web Services security standards.
- Specify Web Services security technology suitable for environments where resource-constrained devices are possible Web Service nodes.

  NOTE: This version of the specification does not consider aspects of Web Service nodes that have any constraining factors.

Non-Goals:

- Define terminal specific definitions to meet terminal requirements (these may be specified in a different specification, or a subsequent version of this specification).
- Define PKI for mobile (see WAP work).
- Provide a tutorial introduction to XML and Web Services security.
- Specify a complete solution for non-repudiation.
- Address security or privacy aspects of identity management

Security considerations include the following:

- Support for security features to address security threats, risk and vulnerabilities. These features are discussed below and include support for confidentiality, integrity, authentication, authorization, access control, privacy, key management and security policy. Considerations to address the threat of non-repudiation are also discussed.
- Allow multiple security token formats, multiple security algorithms and technologies, extensible mechanisms.
- Flexibility of choice of protocol layer for applying security, enabling cost effective choices to meet varying risks.
- Consideration of different device capabilities and requirements.
- Consistency with other efforts, including the W3C Web Services Architecture working group, the OASIS WS-Security TC, WS-I basic security profile and other efforts.

The general approach, consistent with that recommended in the Web Services Architecture Requirements document [WebArch] is to

- Construct a threat model.

- Establish security policies to mitigate threats.
- Construct security model that captures security policies.
- Realize security model in Web Services framework.

Section 7.1.2 provides the normative security standards used to implement the security features described in section 7.1.1.

7.1.1 Security Services (informative)

The security services in this section are intended to identify the security requirements of OMA Web Services and to describe the traditional security goals of reducing vulnerabilities of information, assets and resources.

Important security services include confidentiality, integrity, authentication, authorization, access control, non-repudiation, privacy, key management and security policy. An important consideration is the mitigation of denial of service attacks. This section summarizes the features and their relevant technologies allowing the normative section that follows to focus on profiling of standard technologies for OMA.

7.1.1.1 Authentication

Authentication is used to verify that a party is who they assert to be and may be used, for example, to identify the sender of a message, a recipient, or the signer of some content. The authentication services provided by the OWSER shall address device, application, service requester and service requester authentication as well as data origin authentication.

Mutual authentication (the authentication of both parties in an exchange) is necessary to avoid man-in-the-middle attacks, and the use of timely information such as challenge response should be used to avoid replay attacks.

One widely accepted mechanism to authenticate communicating parties is the use of X.509 certificates with SSL/TLS for server authentication. SSL/TLS also allows the server to require client certificate-based authentication. This mechanism allows parties to authenticate to each other, assuming certificate management is handled properly. Credentials associated with authentication may be short or long-lived. If long-lived, then validation of credentials such as certificates is required of a recipient to ensure that revocation has not occurred. This may be done using OCSP, XKMS or CRLs to give some examples.

SSL/TLS may also further protect HTTP basic or digest authentication as well as application username and password authentication by providing integrity and confidentiality services.

The SOAP Message Security specification [WS-SEC] being developed in the OASIS Web Services Security technical committee enables security tokens to be conveyed with and be bound to SOAP messages. This allows a client to authenticate to a server in conjunction with a SOAP request, and a server to authenticate to a client in the SOAP response, to give an example using the request-response message exchange paradigm. The authentication information conveyed in the token is bound to the message using an XML signature, with either public key or symmetric key technologies. [WS-SEC] supports the carriage of a variety of token types, including the X.509, Kerberos, SAML, and Username types.

Beyond the application protocols described above, the fact that a party has been authenticated may be conveyed within a Web Services environment to enable features such as single sign-on. Such authentication assertions may be conveyed by technologies such as SAML authentication assertions.and can be used to establish, for example, a network identity.

Management of the credentials associated with authentication may depend on trusted third parties, such as in a PKI, or may only require bilateral arrangements. Regardless, this issuance and lifecycle management of credentials is out of the scope of this authentication section.

> **NOTE:** Additional considerations essential to effective security, such as PKI design and management, administration policies, password management and other issues are out of scope of this document but must be considered for a comprehensive security design.
>
> Such additional information may be registered in a UDDI registry as additional uddi:tModel associated with the Web Service interface.

7.1.1.2 Data Integrity

Integrity of information refers to the ability of a receiver to detect whether the content has been changed since creation, either maliciously or by accident. A checksum is not enough, since it could be maliciously replaced to mislead. Instead, a much stronger mechanism such as a digital signature or a MAC with the use of keying material can be used for the detection of any change in the content.

Data integrity may be provided at different protocol layers.

- Transport integrity, such as provided by SSL/TLS, provides transient integrity for a connection. It offers no persistent record and no integrity for information once received by the destination TCP/IP node. Integrity is provided for all the information conveyed and cannot be applied to portions of the information conveyed over the secure session..
- SOAP Application Messaging integrity uses XML Digital Signatures [XML-SIG] to enable the integrity of all or part of the soap:Header and soap:Body. This offers end-to-end integrity between SOAP nodes and may be used to provide integrity appropriately when SOAP intermediaries are used. It also provides protection while SOAP messages are stored.
- Finally, payload application-level integrity uses [XML-SIG] to ensure the integrity of the payload or portions of the XML payload. These signatures may be used for both in-transit and stored integrity with granularity suitable for the application.

  NOTE: It is not appropriate to use encryption for the purpose of integrity. Note also that [XML-SIG] and [XML-ENC] recommendations allow the use of different algorithms.

[XML-SIG] defines how digital signatures may be applied to XML and other non-XML content and how signatures may be verified, taking into account XML specific issues of canonicalization. [XML-SIG] includes the definition of an XML Signature schema to package signature information and processing rules for signature creation and verification. [XML-SIG] defines the ds:KeyInfo element for conveying key information. [WS-SEC] profiles [XML-SIG] for SOAP message integrity and also defines an extended mechanism for conveying key information, using security tokens. This definition of SOAP header tokens is extensible and supports a wide variety of security mechanisms.

7.1.1.3 Confidentiality

Confidentiality is the property that unauthorized parties cannot view information. Typically confidentiality is provided using encryption technologies, such as symmetric and asymmetric encryption. The topic of confidentiality includes the choice and specifications of encryption algorithms, packaging of encryption metadata with encrypted content, and the relationship to the content and protocol model. Confidential communications are often necessary to preserve the privacy of information.

Confidentiality may be deployed at different layers in the protocol stack, depending on application requirements. Use at different layers has different benefits and issues.

- Confidentiality at the IPSec layer provides encryption of a variety of different higher-level protocols without additional effort, but may not function with network address translation equipment.
- SSL/TLS offers point-to-point confidentiality above the TCP layer, is easy to deploy, and is widely adopted. SSL/TLS does not provide adequate confidentiality when messages are routed through application intermediaries since decryption is necessary to route the message.
- [WS-SEC] provides end-to-end security between SOAP endpoints, enabling the use of confidentiality when using SOAP intermediaries. SOAP message confidentiality may be applied to any combination of soap:Header and soap:Body content as long as the SOAP message structure is maintained, providing a granularity to meet SOAP message and application requirements.
- Application level confidentiality using [XML-ENC] for payload content offers end-to-end application confidentiality, enables fine granularity (e.g. only encrypt the credit card number), and allows confidentiality of information stored at servers as well as in-transit.

Two specifications are critical to Web Services confidentiality - XML Encryption [XML-ENC] and SOAP Message Security [WS-SEC]. [XML-ENC] defines a process for encrypting XML or other data and representing the result in XML. This results in an xenc:EncryptedData element that either contains or references the ciphertext as well as additional information for processing, such as the encryption algorithm and the key information. XML Encryption may be used to encrypt XML elements and element content, replacing such content with an xenc:EncryptedData element. [XML-ENC] may be used for application level security and is also the basis for SOAP message security as defined by [WS-SEC].

> **NOTE:** There are two uses of these specifications, namely application and Web Services confidentiality. Application level security uses [XML-SIG] and [XML-ENC] to sign/encrypt portions of the payload, based on application understanding. As such, which features are chosen by an application are outside the scope of this specification..

[WS-SEC] defines how to use [XML-ENC] to encrypt any combination of SOAP soap:Header as well as soap:Body content. It also defines security tokens to convey key information.

> **NOTE:** Which parts of the payload are signed and encrypted is out of scope of this specification, as this is an application decision. The technique is extensible and flexible to meet any application requirements.
>
> Note also that which parts of a message are signed or encrypted are a part of the overall application specification, which may be made available as a uddi:tModel in a UDDI service registry or communicated by out-of-band means to WSRs.

Profiling direct use of [XML-SIG] and [XML-ENC] by an application to secure application data (e.g., conveyed in a soap:Body element) is out of scope of this specification.

Although SOAP messaging security and application security both use [XML-ENC], it is used differently in each case. For example, [XML-ENC] defines how the ds:KeyInfo element may be used to convey key information. SOAP Message Security [WS-SEC] defines mechanisms based on security tokens conveyed in the soap:Header and referenced using an extension of ds:KeyInfo, using SecurityToken references from the ds:KeyInfo elements. [WS-SEC] also defines additional processing rules, such as the use of the xenc:ReferenceList elements in the wsse:Security to reference all encrypted content destined for that role.

7.1.1.4 Key Management

The security and reliability of any communication process is directly dependent on the quality of key management and protection afforded to the keys. The functions of key management are to provide secure key generation, storage, renewal, revocation, exchange and use. The security of encrypted or authenticated data is strictly dependent upon the prevention of unauthorized disclosure, substitution, deletion and use of keys. If keys are compromised, the security of the data can no longer be assured.

Key management includes establishing a security context for creating, registering, sharing and validating keys. Key sharing can be performed differently depending on application requirements, including out of band communication. Scalable solutions may require a back end infrastructure, such as a public key infrastructure (PKI) or a Kerberos system. Differences in the methods and technologies result in different mechanisms, but the goals are the same, to reduce the risks of inappropriate key use and to provide a uniform, scalable system for key management.

The XML Key Management Specification [XKMS] defines a Web Service interface for public key registration, location and validation. [XKMS] is designed to work with the ds:KeyInfo mechanism, used by both [XML-SIG] and [XML-ENC] . [XKMS] is designed to hide the details of a backend key management system from an application, and to impose minimal client requirements, enabling lightweight clients to take advantage of key management systems. [XKMS] defines a public key registration web interface, and a key location and validation interface, providing a layer of indirection between applications and a PKI or other security infrastructure implementation.

> **NOTE:** The backend infrastructure (such as, for example, the choice of certificate authority and associated mechanisms) in a Web Service provider network must be agreed to by the WSR and WSP, including legal agreements. The usage of ds:KeyInfo is defined when the PKI is established.
>
> Note also that the mechanism for providing additional information such as the above may be done by providing a particular uddi:tModel referencing a specification in the UDDI registry entry for the Web Service.

[WS-SEC] Web Servicerelies on security tokens that may also require validation. This may lead to additional XML-based key management interfaces in the future.

Traditional PKI clients, including mobile terminals, may also support traditional PKI registration mechanisms such as the Cryptographic Request Syntax (PKCS#10), Cryptographic Message Syntax (CMS/PKCS#7) as well as the Online Certificate Status Protocol [OCSP] for online certificate validation. Different techniques are appropriate for different circumstances.

Most mobile terminal implementations may support the [OCSP], for example. [OCSP] is being profiled [OCSPProf] by the OMA Security Group for this purpose.

Although key and certificate validation are often essential to make meaningful use of key-based security services they are not always required by applications. In addition, the choice of technique will depend on the platform and implementation constraints. Thus, this specification recommends use of one of the following techniques, and if used imposes some normative requirements in section 7.1.2.4:

- OCSP
- XKMS

Note that the description of OCSP in the normative section refers to the OMA profiling of OCSP.

7.1.1.5 Access Control/Authorization

Access Control and Authorization are security mechanisms that provide the appropriate access to a system or application. They may also be provided at different levels of the protocol stack. The network may make coarse-grained decisions about access to the network, systems may provide services to manage access to their resources, or the resources themselves may restrict who is able to use them. In some topologies, an authorization server may determine whether an authenticated party is allowed to access a resource or perform some action.

An authorization assertion may be expressed in XML using SAML and conveyed to other parties, or other mechanisms may be used to convey authorization asssertions. How the decision is made regarding authorization is server dependent, but one way to express rules in XML is using the Extensible Access Markup Language [XACML]. How servers perform authorization is out of scope of this document. How they share this information in an interoperable manner is in scope.

Although appropriate access control should be required of endpoint implementations, the scope of this specification is limited to interoperability of the Web Services protocols and mechanisms. As such, information necessary for access control decisions may be conveyed to Web Service requesters and Web Service providers by a variety of mechanisms. As a result this specification recommends but does not mandate use of different techniques. If used, this specification imposes some normative requirements as specified in section 7.1.2.6:

- SAML
- Web Services Security SOAP Message Security SAML Token Profile

7.1.1.6 Non-Repudiation

Repudiation is defined as the “Denial by one of the entities involved in a communication of having participated in all or part of the communication”. (Source: [X800]). Non-repudiation is the use of technology, business rules and legal mechanisms to reduce the risk of repudiation to an acceptable level.

Discussion of non-repudiation in a pure technology sense is not meaningful since the issue is intrinsically linked to business and legal issues. Non-repudiation technologies can be correctly considered to support dispute resolution and support for reduction of repudiation risk.

Endorsement using long-lived digital signatures may be used to provide evidence that the signing party has agreed to a contract, approved an action, read some material or agreed to some other statement (verbal or written) when creating the signature. Non-repudiation requires that only the signer have access to their signing material, that appropriate information is included with the signature (such as a timestamp and the reason for signing) and that the signature be persistent. This means that signatures for non-repudiation cannot be transitory signatures such as used in SSL/TLS, but must be long-lived signatures suitable for dispute resolution.

> **NOTE:** The verification of a digital signature does not prove that the alleged signatory actually affixed the actual digital signature. One of the fundamental issues being debated is whether or not a digital signature should be treated differently than a traditional signature and whether or not the current technical definitions of non-repudiation services (as per the ISO/IEC 13888-1,-2,-3) take into account the possibility of private key theft or identity theft. In the legal sense (according to the rights that exist within common law jurisdictions), someone who "signs" a document is always able to repudiate a signature that has been attributed to him or her by claiming the signature is a forgery or that if the signature is not forged, that it was obtained under duress or fraudulent circumstances. The burden of proof then falls on the relying party to prove that the signature was obtained correctly.

Non-repudiation imposes requirements on key management, including due diligence on key registration and certificate issuance and the management of the certificate lifecycle. Separate encryption and signing keys are required since encryption keys may be backed up (to allow recovery of information) while signing keys must remain under the sole control of the signer (for non-repudiation). Effective non-repudiation requires more than technology, but also the business, legal and process controls necessary to make it meaningful.

The signing technologies needed to support non-repudiation and dispute resolution includes digital signatures together with additional information such as the signing timestamp and the reason for signing, the full name of the signer. What is required depends on the application, for exampleFDA 21 CFR 11 [21CFR11] defines information required for online FDA submissions. Currently work is progressing in the Oasis Digital Signature Services technical committee and other forums to define how signatures can be generally meaningful for non-repudiation. The work in OASIS is based on [XML-SIG] and imposes some normative requirements in section 7.1.2.5 on the following optional techniques, if used.

The following mechanisms for supporting non-repudiation are specified normatively in section 7.1.2.5:

- XML Digital Signature
- Definitions of additional content to be included with XML Digital Signatures

Non-repudiation requires effective key and credential management, as discussed in section 7.1.2.4.

7.1.1.7 Denial Of Service Threat Mitigation

Another security concern is that a service should be available and access not denied by attacks against the server. This is known as denial of service attacks. Denial of service attacks can disable a server from providing services to legitimate users by overloading it with request processing or other incoming events that overwhelm the server. It is appropriate to take measures to mitigate the risks associated with denial of service attacks that may degrade or disable the ability of a server to respond to requests for service, when this is a concern. However, this is out of scope of this specification.

Some approaches to reducing the threat include:

1. Requiring clients to authenticate below the application protocol level
    a. SOAP over HTTPS with client-side certificates gives some traceability, providing a measure of deterence.
    b. Requiring signed requests

       The signature should contain a timestamp to reduce the window of possibility for replay attacks

       NOTE: Signature verification still imposes a processing requirement on the server but can prevent it from taking inappropriate action or even more intensive application processing.
2. Implementation level guards for resource consumption

Other techniques such as client puzzles may also be used to reduce the risk of denial of service attacks. (See [DOS])

The following mechanisms may contribute to availability if used correctly:

- SSL/TLS (7.1.2.1)
- WS-Security (7.1.2.2)

7.1.2 Normative security technologies

All compliant implementations are required to provide support integrity and confidentiality and to meet this requirement they MUST support either transport-level or message-level security.

The following approaches to integrity are appropriate for Web Services integrity and their use is specified normatively:

- Transport Integrity: SSL/TLS (7.1.2.1)
- SOAP Messaging Integrity: WS-Security (7.1.2.2)

The following standards are appropriate to Web Services confidentiality and their use are specified normatively:

- Transport Confidentiality: SSL/TLS (7.1.2.1)
- SOAP Messaging Confidentiality: WS-Security (7.1.2.2)

NOTE: Additionally, readers are urged to examine the ongoing work of the WS-I Basic Security Profile work group.

7.1.2.1 Transport Level Security – SSL/TLS

Implementations MAY use SSL/TLS to provide transport-level security.When HTTP over SSL/TLS is used, following specifications MUST be supported (note these are required by the WS-I basic profile):

- RFC2818: HTTP Over TLS, http://www.ietf.org/rfc/rfc2818
- RFC2246: The TLS Protocol Version 1.0, http://www.ietf.org/rfc/rfc2246
- The SSL Protocol Version 3.0, http://wp.netscape.com/eng/ssl3/draft302.txt
- RFC2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile, http://www.ietf.org/rfc/rfc2459

When HTTP over SSL/TLS is used, use of one of the following ciphersuites MUST be used (consistent with SAML guidelines [SAML-Security]):

- TLS_RSA_WITH_3DES_EDE_CBC_SHA (when using TLS)
- TLS_RSA_WITH_RC4_128_SHA (when using TLS)
- SSL_RSA_WITH_3DES_EDE_CBC_SHA (when using SSL)
- Forward looking RECOMMENDED: TLS_RSA_WITH_AES_128_CBC_SHA
- Forward looking RECOMMENDED: TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA

When HTTP over SSL/TLS is used, SSL 3.0 or TLS 1.0 is REQUIRED. SSL 2.0 MUST NOT be used.

Client authentication with client certificates is RECOMMENDED when prevention of denial of service attacks is a concern.

When SSL/TLS is used to create an authenticated server-to-server link, both client and server authentication MUST be provided.

7.1.2.2 Message Level Security: SOAP Message Security

Implementations MAY use SOAP Message Security to provide message level security.

- Web Services Security: SOAP Message Security [WS-SEC]

When message level security is applied, the ubse of the appropriate token profile defined by the OASIS Web Services Security TC MUST be used if that token type is used:

- Web Services Security: Kerberos Token Profile
- Web Services Security: SAML Token Profile
- Web Services Security: X509 Token Profile
- Web Services Security: Username Token Profile

When SOAP Message Security together with one of the above token profiles is used for end-to-end authentication, including the soap:Body element as part of the signed target in a security header signature in combination with an X.509 authentication token is recommended. Server support for X.509 authentication tokens MUST be provided.

7.1.2.3 Application Level Security

7.1.2.3.1 XML Digital Signatures

When digital signatures need to be provided, XML Digital Signature [xxxxx] MUST be used.

- XML Digital Signature Recommendation

  It is RECOMMENDED that Exclusive Canonicalization be the Signature CanonicalizationMethod, but the choice of canonicalization method is application dependent.

NOTE: Exclusive canonicalization is required if signed XML is to be moved between XML contexts, but in some cases inclusive canonicalization is more appropriate, for example when QNames are used in the XML schema and namespace propagation is not an issue; for this reason this specification does not mandate either although Exclusive canonicalization is recommended.

Processing rules:

1. For SOAP Message Security the rules in [WS-SEC] should be followed and best practices from WS-I Basic Security Profile observed (if and when available).
2. Canonicalization SHOULD be a Transform within every Signature Reference that refers to XML content, with the choice of canonicalization application dependent
3. Use of InclusiveNamespaces PrefixList for ExclusiveCanonicalization (for more details, see the SAML sig guidelines)
4. When using X.509 public key certificates it is strongly RECOMMENDED that certificate path validation be performed in accordance to the PKIX Profile as specified in [RFC3280]
5. The signer must not assume that the signed element will be the root element during verification. XML signatures must use proper URI fragments for the URI attribute of the Reference element. This URI fragment should reference the id attribute of an element in the same document using an XPointer shortcut reference.

The use of the Web Services Security: Minimalistic Profile (submitted to OASIS and elsewhere) is OPTIONAL, but it can be used as a means to address concerns about canonicalization. Note that this is a draft that has not yet been officially accepted for development at an open standards organization.

7.1.2.3.2 XML Encryption

When XML encryption is used, the following specifications are required and incorporated by reference

- XML Encryption Recommendation
- XML Decryption Transform

The Decryption Transform should be used when both signatures and encryption are used on XML content.

Implementation of XML encryption should meet the MUST/REQUIRED components of the XML Encryption recommendation. When used for SOAP message confidentiality, XML Encryption should be used as specified in WS-Security. In particular

- Use of the Decryption transform is recommended when portions of an XML document are signed using XML Digital Signature and these sections contain encrypted portions (either before or after signing).
- Use of CipherData instead of CipherReferences is recommended to simplify processing, although for large cipher text, references may be suitable.

7.1.2.4 Key Management

Public key and certificate validation are recommended when public keys are used for authentication, non-repudiation or other uses of signing where the consequence of invalid certificates may have a negative impact upon parties. Different mechanisms may be used such as OCSP or [XKMS].

When OCSP is used, adherence to the OMA OCSP profile is REQUIRED.

When [XKMS] is used, all required features of that specification must be met, including the compliance requirements listed in that specification.

7.1.2.5 Signing for Non-Repudiation

Non-repudiation requires the use of XML Digital Signature as profiled above. In addition, the XML Digital Signature SHOULD include a signature Reference to signature properties, including a timestamp specifying the time of signing, a signature purpose, and the full name of the signer. Requirements on the signature properties MAY be derived from

regulations, such as the 21CFR11 FDA regulation regarding the electronic submission of statements to the FDA [21CFR11]. Non-repudiation also requires appropriate due diligence upon credential issuance and appropriate revocation procedures. Such aspects are out of scope of this specification, but have been addressed in PKI business, legal and technology standards.

7.1.2.6 Authorization/Access Control

When SAML is used to convey information suitable for access control, the SAML 1.1 Conformance [SAMLConf] requirements MUST be met.

7.2 Privacy management functions (informative)

Privacy has three aspects: personal, territorial and informational. In an OMA context, personal privacy deals with mechanisms to ensure that end users are not exposed to whatever violates their moral senses, while territorial privacy is about protecting the user's property – e.g. the user equipment – from being invaded by undesired content, such as SMS or email messages. Informational privacy is about data protection, and the user's right to determine how, when and to what extent information about her is communicated to other parties, and the execution of this right might be based on her knowledge about what the other party's intention is.

Privacy is a broad area of which the topic of information privacy is but one aspect, and is the main issue in OMA Web Services security. The information protected by information privacy can include personal identifiable information, as well as other personal information such as preferences, age, gender, account numbers, user equipment-related information and so on. Legislation is a major driver of privacy requirements in some countries. Data protection aspects must be considered before a service is used, during service usage, and after it is used so that the service provider does not misuse the data collected.

Privacy is an issue before a service is used (how the other entity may be contacted, if at all), during service usage (what is revealed about an entity to another entity), and after service usage (how the revealed information may be used afterwards).

Privacy enhancing technologies include mechanisms to ensure anonymity, meaning that the identity of a party is not known to another party. Pseudonyms in conjunction with appropriate authorization may be used to protect information privacy. An example of such a pseudonym usage is a web site offering some content to be downloaded from a third party. A pseudonym is given to the third party, so that the party can share information about the download with the web site, but cannot track the downloads of individual parties over time.

Privacy policies may be used to proactively define how and when personal information may be released, to whom and, for how long. Privacy policies may be specified in different ways, one example is the W3C Platform for Privacy Preferences (P3P). P3P policies provide the user with information about the privacy practices of a server. In all cases, information may be compromised if appropriate confidentiality of the information is not maintained, such as during transit or storage. As a result, mechanisms designed to ensure confidentiality may be used to reduce the risks of inappropriate information disclosure.

A common privacy solution is desirable for different enablers, including Presence, SyncML, MMS, and Location, for example.

The implementation of privacy controls requires the means to state and convey privacy policies as well as the means to enforce them. This document references mechanisms to express and convey privacy policy statements. Policy enforcement mechanisms are out of scope.

The Platform for Privacy Preferences (P3P) may be used to make statements about how personally identifiable information about an end-user is used by a server. P3P also defines how a client might query a server to obtain this information. The compact form of P3P is recommended.

Privacy issues are also related to identity mangement systems that may be used to enable transactions without releasing personally identifiable information.

Related to privacy, the following specifications MAY be required and, if so, incorporated by reference

- Platform for Privacy Preferences (P3P)

8. Service management functions

This section of the OWSER discusses and provides normative specifications for common Service Management features. The scope taken in this chapter is essentially the service life cycle support for a Web Service. The features of service management specified in this section are those that support the tripartite relationship – publish, find and bind – between the WSR, WSP and the service registry as described in the OWSER Overview document (see section 5.1 therein).

> **NOTE:** Not all aspects of Web Service management features are specified in this document. Future OMA Web Service Enabler releases will provide specifications to cover additional features.

8.1 Service registry features (informative)

8.1.1 Web Service Provider registration

The first step in making a Web Service available for consumption in a distributed environment is the registration of the WSP in the registry. This specification does not mandate any database schema for the service registry.

The WSP registration comprises of the information of the WSP as a business entity. It may also include a description of the Web Service in terms of its functions, capabilities and interfaces. (Note that the Web Service description is not available for discovery by applications until it has been published.)

The three functions related to service registration are:

- Registration,
- De-registration,
- Modification

Security regarding authentication and authorization of the service representative registering the WSP in the registry has to be provided. Note that this security can be realized by means of transport level security, e.g., SSL.

8.1.2 Web Service publication

Publication of a registeredWeb Service is the act of making a service description or a reference to a service description visible to consuming applications, i.e. exposure of the Web Service description in a service repository that is accessible to external parties. The Web Service description that is published can differ from the Web Service description that was registered during service registration, because of marketing reasons and/or the fact that service registration and publication may be performed in different business domains, where the publication domain can *re*-present the service descrition in its publication.

The functions related to Web Service publication are:

- Publish,
- Unpublish

After a Web Service has been published, it is discoverable by consuming applications.

Security considerations related to service publication include the authentication and authorization of the service publisher.

8.1.3 Web Service discovery

Web Service discovery is the activity of an application to find what Web Services and their associated capabilities are available for consumption, and under what conditions.

The functions related to Web Service discovery are:

- Discover

Authentication and authorization of the application (or representative) is optional during the discovery phase.

The SLA can contain integrity management related obligations.

8.2 Web Service registry specifications

This section provides normative guidance on the use of UDDI as the Web Services registry, as indicated in section 6.5.

8.2.1 Web Service registration

When UDDI is used, a business entity MUST use the UDDI 2.03 Data Structures schema [UDDIData] for describing a Web Service that it offers.

A WSP must have established an identity with an UDDI node to be allowed to register and publish information to that node. Write access is permitted only to authenticated users and write access controls are applied to UDDI entries.

In the case of registration at a public UDDI node, the information may be replicated at other public UDDI nodes but replication does not impose additional security requirements on the WSP. Note that the WSP identity applies to the public UDDI node with which the identity is initially established; it is not shared with the other replicated UDDI nodes.

A private UDDI node MAY impose the same access security model as a public UDDI node.

8.2.2 Web Service publication

When UDDI is used, a WS MUST use the UDDI 2.04 Publishing API [UDDI] for service publication.

OMA enablers that offer Web Service MAY publish a WSDL description. As a part of the publication process, a new uddi:tModel within the OMA UDDI namespace MUST be created to reference the WSDL document corresponding to the enabler release.

A uddi:tModel provides a generic mechanism for associating arbitrary metadata with services and other entities in a UDDI registry.

Additional uddi:tModel MAY be created by WSPs to identify additional information that may be necessary to completely describe the overall context and policies under which the service is offered. Such uddi:tModel will identify specifications that SHOULD be in textual format.

There are essentially two approaches for registering such additional information in UDDI. The one is to directly reference remotely accessible information in UDDI entities, the other is to register such additional information as distinct uddi:tModel and then reference these uddi:tModel in each UDDI entity that is using that information.

UDDI registries MUST generate a unique uddi:tModelKey to uniquely identify each uddi:tModel that is defined or referenced so that UDDI registry users can expect the same behavior across different UDDI registries. Such a key is typically provided by a UUID.

> **NOTE:** Future releases of the OWSER may specify technologies by which such additional information may be expressed in a formal notation that would allow programmatic consumption of such information by potential WSRs. Such notational support is not available at this time in a standardized format.

8.2.3 Web Service discovery

When UDDI is used, a WSR MUST use the UDDI 2.04 Inquiry API [UDDI] for service discovery.

> **NOTE:** The use of UDDI for service discovery is optional, because additional service information can be obtained by many out-of-band means, many of which may not have a formal expression in a protocol. However, if a distributed service discovery mechanism is used, this specification mandates the use of the UDDI mechanism as described in this section.

UDDI Release 2 does not impose any read access controls on registry entries. Therefore all information in a UDDI registry is available without requiring authentication. However, a private UDDI node MAY impose read access controls to all or certain entries in the registry.

Appendix A. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [CREQ].

All

Item	Function	Reference	Status	Requirement
OWSER-All-001	Message transfer- HTTP	6.6	M	
OWSER-All-002	XML messaging - SOAP	6.3	M	
OWSER-All-003	Attachments – SOAP with Attachments	6.8	O	
OWSER-All-004	XML encryption	7.1.2.3.2	O	
OWSER-All-005	XML digital signature	7.1.2.3.1	O	

WSR

Item	Function	Reference	Status	Requirement
OWSER-WSR-001	Service discovery	6.5, 8.2.3	O	
OWSER-WSR-002	Message integrity	7.1.2.1, 7.1.2.2	M	OWSER-WSR-003 OR OSWER-WSR-004
OWSER-WSR-003	Transport level integrity	7.1.2.1	O	
OWSER-WSR-004	Message level integrity	7.1.2.2	O	OWSER-All-005
OWSER-WSR-005	Message confidentiality	7.1.2.1, 7.1.2.2	M	OWSER-WSR-006 OR OWSER-WSR-007
OWSER-WSR-006	Transport level confidentiality	7.1.2.1	O	
OWSER-WSR-007	Message level confidentiality	7.1.2.2	O	OWSER-All-004
OWSER-WSR-008	Application level security – digital signature	71.2.3.1	O	OWSER-All-005
OWSER-WSR-009	Application level security – encryption	7.1.2.3.2	O	OWSER-All-004
OWSER-WSR-010	Key management	7.1.2.4	O	
OWSER-WSR-011	Signing for non-repudiation	7.1.2.5	O	
OWSER-WSR-012	Access control	7.1.2.6	O	

Web Service (WS)

Item	Function	Reference	Status	Requirement
OWSER-WS-001	Service publication	6.5, 8.2.2	O	
OWSER-WS-002	Message integrity	7.1.2.1, 7.1.2.2	M	OWSER-WS-003 OR OSWER-WS-004
OWSER-WS-003	Transport level integrity	7.1.2.1	O	
OWSER-WS-004	Message level integrity	7.1.2.2	O	OWSER-All-005
OWSER-WS-005	Message confidentiality	7.1.2.1, 7.1.2.2	M	OWSER-WS-006 OR OWSER-WS-007
OWSER-WS-006	Transport level confidentiality	7.1.2.1	O	

Item	Function	Reference	Status	Requirement
OWSER-WS-007	Message level confidentiality	7.1.2.2	O	OWSER-All-004
OWSER-WS-008	Application level security – encryption	7.1.2.3.2	O	OWSER-All-004
OWSER-WS-009	Application level security – digital signature	71.2.3.1	O	OWSER-All-005
OWSER-WS-010	Key management	7.1.2.4	O	
OWSER-WS-010	Signing for non-repudiation	7.1.2.5	O	
OWSER-WS-012	Access control	7.1.2.6	O	

SOAP Intermediary

Item	Function	Reference	Status	Requirement
OWSER-INT-001	Message integrity	7.1.2.1, 7.1.2.2	M	OWSER-INT-002 OR OSWER-INT-003
OWSER-INT-002	Transport level integrity	7.1.2.1	O	
OWSER-INT-003	Message level integrity	7.1.2.2	O	OWSER-All-005
OWSER-INT-004	Message confidentiality	7.1.2.1, 7.1.2.2	M	OWSER-INT-005 OR OWSER-INT-006
OWSER-INT-005	Transport level confidentiality	7.1.2.1	O	
OWSER-INT-006	Message level confidentiality	7.1.2.2	O	OWSER-All-004
OWSER-INT-007	Application level security – encryption	7.1.2.3.2	O	OWSER-All-004
OWSER-INT-008	Application level security – digital signature	71.2.3.1	O	OWSER-All-005

Appendix B. Document History (informative)

B.1 Approved Version History

Reference	Date	Description
OMA-OWSER-Core-Specification-V1_0	15 Jul 2004	Initial document to address the basic starting point Ref TP Doc# OMA-TP-2004-0238-OWSER-V1_0-for-final-approval

Draft/Candidate Version 1.1 History

Document Identifier	Date	Sections	Description
OMA-TS-OWSER-Core-Specification-V1_1-20050824	24 Aug 2005	Separation of NI function (now addressed in its own enabler)	OMA-TS-OWSER-Core-Specification-V1_1
OMA-TS-OWSER-Core-Specification-V1_1-2001205	5 December 2005	B	Correction of document history.
Candidate Version OMA-TS-OWSER-Core-Specification-V1_1	20 Dec 2005		Status changed to Candidate by TP TP ref # OMA-TP-2005-0394-OWSER-V1_1-for-Candidate-approval