Devices Profile for Web Services Version 1.1

Public Review Draft 01

27 January 2009

Specification URIs:
This Version:
http://docs.oasis-open.org/ws-dd/dpws/1.1/pr-01/wsdd-dpws-1.1-spec-pr-01.html
http://docs.oasis-open.org/ws-dd/dpws/1.1/pr-01/wsdd-dpws-1.1-spec-pr-01.docx (Authoritative Format)
http://docs.oasis-open.org/ws-dd/dpws/1.1/pr-01/wsdd-dpws-1.1-spec-pr-01.pdf

Previous Version:
http://docs.oasis-open.org/ws-dd/dpws/1.1/cd-01/wsdd-dpws-1.1-spec-cd-01.html
http://docs.oasis-open.org/ws-dd/dpws/1.1/cd-01/wsdd-dpws-1.1-spec-cd-01.docx
http://docs.oasis-open.org/ws-dd/dpws/1.1/cd-01/wsdd-dpws-1.1-spec-cd-01.pdf

Latest Version:
http://docs.oasis-open.org/ws-dd/dpws/wsdd-dpws-1.1-spec.html
http://docs.oasis-open.org/ws-dd/dpws/wsdd-dpws-1.1-spec.docx
http://docs.oasis-open.org/ws-dd/dpws/wsdd-dpws-1.1-spec.pdf

Technical Committee:
OASIS Web Services Discovery and Web Services Devices Profile (WS-DD) TC

Chair(s):
Toby Nixon (Microsoft Corporation)
Alain Regnier (Ricoh Company Limited)

Editor(s):
Dan Driscoll (Microsoft Corporation)
Antoine Mensch

Declared XML Namespace(s):
http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01

Abstract:
This profile defines a minimal set of implementation constraints to enable secure Web service messaging, discovery, description, and eventing on resource-constrained endpoints.

Status:
This document was last revised or approved by the OASIS Web Services Discovery and Web Services Devices Profile (WS-DD) TC on the above date. The level of approval is also listed above. Check the “Latest Version” or “Latest Approved Version” location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee’s email list. Others should send comments to the Technical Committee by using the “Send A Comment” button on the Technical Committee’s web page at http://www.oasis-open.org/committees/ws-dd/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (http://www.oasis-open.org/committees/ws-dd/ipr.php.

The non-normative errata page for this specification is located at http://www.oasis-open.org/committees/ws-dd/.

Notices

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/who/trademark.php for above guidance.

Table of Contents

1 Introduction

The Web services architecture includes a suite of specifications that define rich functions and that may be composed to meet varied service requirements. To promote both interoperability between resource-constrained Web service implementations and interoperability with more flexible client implementations, this profile identifies a core set of Web service specifications in the following areas:

- Sending secure messages to and from a Web service
- Dynamically discovering a Web service
- Describing a Web service
- Subscribing to, and receiving events from, a Web service

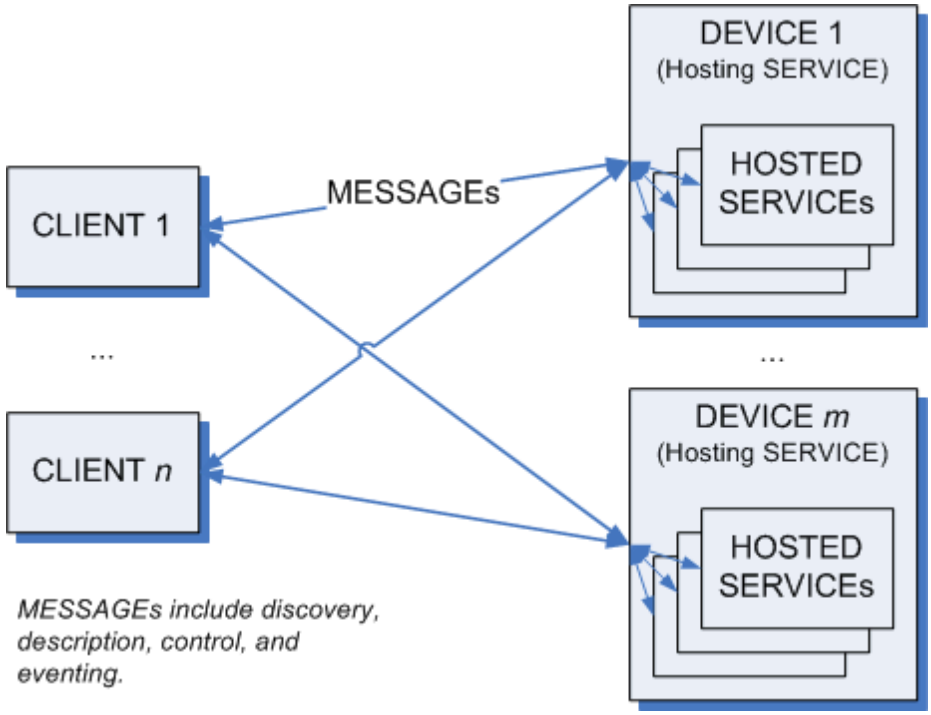
In each of these areas of scope, this profile defines minimal implementation requirements for compliant Web service implementations.

1.1 Requirements

This profile intends to meet the following requirements:

- Identify a minimal set of Web service specifications needed to enable secure messaging, dynamic discovery, description, and eventing.
- Constrain Web services protocols and formats so Web services can be implemented on peripheral-class and consumer electronics-class hardware.
- Define minimum requirements for compliance without constraining richer implementations.

1.2 Terminology



MESSAGE

Protocol elements that are exchanged, usually over a network, to affect a Web service. Always includes a SOAP ENVELOPE. Typically also includes transport framing information such as HTTP headers, TCP headers, and IP headers.

SOAP ENVELOPE

An XML Infoset that consists of a document information item [XML Infoset] with exactly one member in its [children] property, which MUST be the SOAP Envelope [SOAP 1.2] element information item.

MIME SOAP ENVELOPE

A SOAP ENVELOPE serialized using MIME Multipart Serialization [MTOM].

TEXT SOAP ENVELOPE

A SOAP ENVELOPE serialized as application/soap+xml.

CLIENT

A network endpoint that sends MESSAGEs to and/or receives MESSAGEs from a SERVICE.

SERVICE

A software system that exposes its capabilities by receiving and/or sending MESSAGEs on one or several network endpoints.

DEVICE

A distinguished type of SERVICE that hosts other SERVICEs and sends and/or receives one or more specific types of MESSAGEs.

HOSTED SERVICE

A distinguished type of SERVICE that is hosted by another SERVICE. The lifetime of the HOSTED SERVICE is a subset of the lifetime of its host. The HOSTED SERVICE is visible (not encapsulated) and is addressed separately from its host. Each HOSTED SERVICE has exactly one host. (The relationship is not transitive.)

SENDER

A CLIENT or SERVICE that sends a MESSAGE.

RECEIVER

A CLIENT or SERVICE that receives a MESSAGE.

1.3 Notational 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 **[RFC 2119]**.

- This specification uses the following syntax to define normative outlines for messages:
- The syntax appears as an XML instance, but values in italics indicate data types instead of literal values.
- Characters are appended to elements and attributes to indicate cardinality:
  - "?" (0 or 1)
  - "*" (0 or more)
  - "+" (1 or more)
- The character "|" is used to indicate a choice between alternatives.
- The characters "(" and ")" are used to indicate that contained items are to be treated as a group with respect to cardinality or choice.
- The characters "[" and "]" are used to call out references and property names.
- Ellipses (i.e., "...") indicate points of extensibility. Additional children and/or attributes MAY be added at the indicated extension points but MUST NOT contradict the semantics of the parent and/or owner, respectively. By default, if a receiver does not recognize an extension, the receiver SHOULD ignore the extension; exceptions to this processing rule, if any, are clearly indicated below.

- XML namespace prefixes (see Table 1) are used to indicate the namespace of the element being defined.

This specification uses the **[action]** and Fault properties [WS-Addressing] to define faults.

Normative statements in this profile are called out explicitly as follows:

Rnnn: Normative statement text goes here.

where "nnnn" is replaced by the statement number. Each statement contains exactly one requirement level keyword (e.g., "MUST") and one conformance target keyword (e.g., "MESSAGE").

1.4 XML Namespaces

The XML namespace URI that MUST be used be implementations of this specification is:

```
http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01
```

Table 1 lists XML namespaces that are used in this specification. The choice of any namespace prefix is arbitrary and not semantically significant.

Table 1: Prefixes and XML namespaces used in this specification.

Prefix	XML Namespace	Specification(s)
soap	http://www.w3.org/2003/05/soap-evelope	[SOAP 1.2]
wsa	http://www.w3.org/2005/08/addressing	[WS-Addressing]
wsd	http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01	[WS-Discovery]
dpws	http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01	This profile
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL 1.1]
wse	http://schemas.xmlsoap.org/ws/2004/08/eventing	[WS-Eventing]
wsp	http://www.w3.org/ns/ws-policy	[WS-Policy, WS-PolicyAttachment]
wsx	http://schemas.xmlsoap.org/ws/2004/09/mex	[WS-MetadataExchange]

1.5 Normative References

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

[AES/TLS] P.Chown, *Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)*, http://www.ietf.org/rfc/rfc3268.txt, IETF RFC 3268, June 2004.

[BP 1.1, Section 4] K. Ballinger, et al, *Basic Profile Version 1.1, Section 4: Service Description*, http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html#description, August 2004.

[HTTP/1.1] R.Fielding, et al, *Hypertext Transfer Protocol -- HTTP/1.1*, http://www.ietf.org/rfc/rfc2616.txt, IETF RFC 2616, June 1999.

[HTTP Authentication] J. Franks, et al, *HTTP Authentication: Basic and Digest Access Authentication*, http://www.ietf.org/rfc/rfc2617.txt, IETF RFC 2617, June 1999.

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

[MTOM] N. Mendelsohn, et al, *SOAP Message Transmission Optimization Mechanism*, http://www.w3.org/TR/2005/REC-soap12-mtom-20050125/, January 2005.

[RFC 4122] P. Leach, et al, *A Universally Unique IDentifier (UUID) URN Namespace*, http://www.ietf.org/rfc/rfc4122.txt, IETF RFC 4122, July 2005.

[SHA] *Secure Hash Standard*, http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf, October 2008.

[SOAP 1.2, Part 1] M. Gudgin, et al, *SOAP Version 1.2 Part 1: Messaging Framework*, http://www.w3.org/TR/2007/REC-soap12-part1-20070427/, April 2007.

[SOAP 1.2, Part 2]

M. Gudgin, et al, *SOAP Version 1.2 Part 2: Adjuncts, Section 7: SOAP HTTP Binding*, http://www.w3.org/TR/2007/REC-soap12-part2-20070427/#soapinhttp, April 2007.

[SOAP-over-UDP] OASIS Committee Draft 02, *SOAP-over-UDP*, http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/cd-02/wsdd-soapoverudp-1.1-spec-cd-02.docx, 27 January 2009.

[TLS] T. Dierks, et al, *The TLS Protocol, Version 1.0*, http://www.ietf.org/rfc/rfc2246.txt, IETF RFC 2246, January 1999.

[WS-Addressing] W3C Recommendation, *Web Services Addressing 1.0 - Core*, http://www.w3.org/TR/2006/REC-ws-addr-core-20060509, 9 May, 2006.

[WS-Discovery] OASIS Committee Draft 02, *Web Services Dynamic Discovery (WS-Discovery)*, http://docs.oasis-open.org/ws-dd/discovery/1.1/cd-02/wsdd-discovery-1.1-spec-cd-02.docx, 27 January 2009.

[WSDL 1.1] E. Christensen, et al, *Web Services Description Language (WSDL) 1.1*, http://www.w3.org/TR/2001/NOTE-wsdl-20010315, March 2001.

[WSDL Binding for SOAP 1.2]

K. Ballinger, et al, *WSDL 1.1 Binding Extension for SOAP 1.2*, http://www.w3.org/Submission/2006/SUBM-wsdl11soap12-20060405/, 5 April 2006.

[WS-Eventing] D. Box, et al, *Web Services Eventing (WS-Eventing),* http://www.w3.org/Submission/2006/SUBM-WS-Eventing-20060315/, 15 March 2006.

[WS-MetadataExchange]

K. Ballinger, et al, *Web Services Metadata Exchange 1.1 (WS-MetadataExchange)*, http://www.w3.org/Submission/2008/SUBM-WS-MetadataExchange-20080813/, 13 August 2008.

[WS-Policy] W3C Recommendation, *Web Services Policy 1.5 - Framework*, http://www.w3.org/TR/2007/REC-ws-policy-20070904/, 4 September 2007.

[WS-PolicyAttachment]

W3C Recommendation, *Web Services Policy 1.5 - Attachment*, http://www.w3.org/TR/2007/REC-ws-policy-attach-20070904/, 4 September 2007.

[WS-Transfer] J. Alexander, et al, *Web Service Transfer (WS-Transfer)*, http://www.w3.org/Submission/2006/SUBM-WS-Transfer-20060927/, 27 September 2006.

[X.509.v3] *ITU-T X.509.v3 Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks (ISO/IEC/ITU 9594-8)*

[XML Schema, Part 1]

H. Thompson, et al, *XML Schema Part 1: Structures*, http://www.w3.org/TR/2001/REC-xmlschema-1/20010502/, May 2001.

[XML Schema, Part 2]

P. Biron, et al, *XML Schema Part 2: Datatypes*, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/, May 2001.

1.6 Non-Normative References

[IPv6 Autoconfig] S. Thomson, et al, *IPv6 Stateless Address Autoconfiguration*, http://www.ietf.org/rfc/2462.txt, IETF RFC 2462, December 1998.

[DHCP] R. Droms, et al, *Dynamic Host Configuration Protocol*, http://www.ietf.org/rfc/2131.txt, IETF RFC 2131, March 1997.

[XML Infoset] J. Cowan, et al, *XML Information Set (Second Edition)*, http://www.w3.org/TR/2004/REC-xml-infoset/20040204/, February 2004.

[WS-Security] OASIS Standard Specification, *Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)*, http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-SOAPMessageSecurity.pdf, 1 February 2006.

2 Messaging

The scope of this section is the following set of Web services specifications. All of the requirements in these specifications are included by reference except where superseded by normative statements herein:

- [SOAP 1.2, Part 1]
- [SOAP 1.2, Part 2]
- [SOAP-over-UDP]
- [HTTP/1.1]
- [WS-Addressing]
- [RFC 4122]
- [MTOM]

It is assumed that a DEVICE has obtained valid IPv4 and/or IPv6 addresses that do not conflict with other addresses on the network. Mechanisms for obtaining IP addresses are out of the scope of this profile. For more information, see [DHCP] and [IPv6 Autoconfig].

2.1 URI

R0025: A SERVICE MAY fail to process any URI with more than MAX_URI_SIZE octets.
R0027: A SERVICE SHOULD NOT generate a URI with more than MAX_URI_SIZE octets.

The constant MAX_URI_SIZE is defined in Appendix B -- Constants.

2.2 UDP

R0029: A SERVICE SHOULD NOT send a SOAP ENVELOPE that has more octets than the MTU over UDP.

To improve reliability, a SERVICE should minimize the size of SOAP ENVELOPEs sent over UDP. However, some SOAP ENVELOPEs may be larger than an MTU; for example, a signed Hello SOAP ENVELOPE. If a SOAP ENVELOPE is larger than an MTU, the underlying IP network stacks may fragment and reassemble the UDP packet.

R5018: A SERVICE MAY reject a SOAP ENVELOPE received over UDP that has more than MAX_UDP_ENVELOPE_SIZE octets.
R5019: A CLIENT MAY reject a SOAP ENVELOPE received over UDP that has more than MAX_UDP_ENVELOPE_SIZE octets.

Unlike TCP or HTTP messages, UDP datagrams must be received in one chunk, which may lead to excessive resource requirements when receiving large datagrams on small embedded systems. The constant MAX_UDP_ENVELOPE_SIZE is defined in Appendix B -- Constants.

2.3 HTTP

R0001: A SERVICE MUST support transfer-coding = "chunked".
R0012: A SERVICE MUST at least support the SOAP HTTP Binding.
R5000: A CLIENT MUST at least support the SOAP HTTP Binding.
R0013: A SERVICE MUST at least implement the Responding SOAP Node of the SOAP Request-Response Message Exchange Pattern (http://www.w3.org/2003/05/soap/mep/request-response/).

R0014: A SERVICE MAY choose not to implement the Responding SOAP Node of the SOAP Response Message Exchange Pattern (http://www.w3.org/2003/05/soap/mep/soap-response/).
R0015: A SERVICE MAY choose not to support the SOAP Web Method Feature.

R0014 and R0015 relax requirements in [SOAP 1.2].

R0030: A SERVICE MUST at least implement the Responding SOAP Node of an HTTP one-way Message Exchange Pattern where the SOAP ENVELOPE is carried in the HTTP Request and the HTTP Response has a Status Code of 202 Accepted and an empty Entity Body (no SOAP ENVELOPE).
R0017: A SERVICE MUST at least support Request Message SOAP ENVELOPEs and one-way SOAP ENVELOPEs that are delivered using HTTP POST.

2.4 SOAP Envelope

R0034: A SERVICE MUST at least receive and send SOAP 1.2 [SOAP 1.2] SOAP ENVELOPEs.
R0003: A SERVICE MAY reject a TEXT SOAP ENVELOPE with more than MAX_ENVELOPE_SIZE octets.
R0026: A SERVICE SHOULD NOT send a TEXT SOAP ENVELOPE with more than MAX_ENVELOPE_SIZE octets.

Large SOAP ENVELOPEs are expected to be serialized using attachments.

R5001: A SERVICE MUST at least support SOAP ENVELOPEs with UTF-8 encoding.
R5002: A SERVICE MAY choose not to accept SOAP ENVELOPEs with UTF-16 encoding.

2.5 WS-Addressing

R5005: A SERVICE MUST at least support WS-Addressing 1.0 [WS-Addressing].
R5006: A SERVICE MAY reject messages using other versions of WS-Addressing.

Some underlying specifications (e.g., WS-Transfer [WS-Transfer]) explicitly allow other versions of WS-Addressing. DPWS applications should rely solely on WS-Addressing 1.0.

R0004: A DEVICE SHOULD use a urn:uuid scheme IRI as the [address] property of its Endpoint Reference.
R0005: A DEVICE MUST use a stable, globally unique identifier that is constant across re-initializations of the device, and constant across network interfaces and IPv4/v6 addresses as the [address] property of its Endpoint Reference.
R0006: A DEVICE MUST persist the [address] property of its Endpoint Reference across re-initialization and changes in the metadata of the DEVICE and any SERVICEs it hosts.

Because the [address] property of an Endpoint Reference [WS-Addressing] is a SOAP-layer address, there is no requirement to use anything other than a UUID for the [address] property.

R0042: A HOSTED SERVICE SHOULD use an HTTP transport address as the [address] property of its Endpoint References.

Use of other possible values of [address] by a HOSTED SERVICE is out of scope of this profile.

R0031: A SERVICE MUST NOT generate a wsa:InvalidMessageInformationHeader SOAP Fault if the [address] of the [reply endpoint] of an HTTP Request Message SOAP ENVELOPE is "http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous".
R0041: If an HTTP Request Message SOAP ENVELOPE generates a SOAP Fault, a SERVICE MAY discard the SOAP Fault if the [address] of the [fault endpoint] of the HTTP Request Message is not "http://www.w3.org/2005/08/addressing/anonymous".

R0031 and R0041 ensure that messages with non-anonymous address in both the [reply endpoint] and the [fault endpoint] do not result in a fault being sent.

The SOAP HTTP Binding requires the Response Message SOAP ENVELOPE to be transmitted as the HTTP Response of the corresponding Request Message SOAP ENVELOPE.

R0019: A SERVICE MUST include a Message Information Header representing a [relationship] property of type wsa:Reply in each Response Message SOAP ENVELOPE the service generates.

Per WS-Addressing [WS-Addressing], a response SOAP ENVELOPE must include a wsa:RelatesTo SOAP ENVELOPE header block. Since “http://www.w3.org/2005/08/addressing/reply” is the default value for the [relationship] property, the RelationshipType attribute should be omitted from the wsa:RelatesTo SOAP ENVELOPE header block.

R0040: A SERVICE MUST include a Message Information Header representing a [relationship] property of “http://www.w3.org/2005/08/addressing/reply”in each SOAP Fault SOAP ENVELOPE the service generates.

2.6 Attachments

R0022: If a SERVICE supports attachments, the SERVICE MUST support the HTTP Transmission Optimization Feature.

The HTTP Transmission Optimization Feature implies support for the Optimized MIME Multipart Serialization and Abstract Transmission Optimization features.

R0036: A SERVICE MAY reject a MIME SOAP ENVELOPE if the Content-Transfer-Encoding header field mechanism of any MIME part is not "binary".

R0037: A SERVICE MUST NOT send a MIME SOAP ENVELOPE unless the Content-Transfer-Encoding header field mechanism of every MIME part is "binary".

Even for the SOAP Envelope, the "binary" Content-Transfer-Encoding mechanism is more appropriate than the "8bit" mechanism which is suitable only for data that may be represented as relatively short lines of at most 998 octets [MIME].

While DPWS-compliant services are required to support binary encoded MIME parts at a minimum, R0036 allows for them to support others (non-DPWS compliant clients) if they choose. While a service might choose to support more than what is required in DPWS, a DPWS-compliant client cannot assume that the service it is interacting with supports anything beyond binary MIME parts.

R0038: A SERVICE MAY reject a MIME SOAP ENVELOPE if the root part is not the first body part in the Multipart/Related entity.

R0039: A SERVICE MUST NOT send a MIME SOAP ENVELOPE unless root part is the first body part in the Multipart/Related entity.

Per MTOM, the root part of the MIME SOAP ENVELOPE contains an XML representation of the modified SOAP Envelope, with additional parts that contain binary representations of each attachment. This root part must be the first part so a RECEIVER does not have to buffer attachments.

3 Discovery

The scope of this section is the following set of Web services specifications. All of the requirements in these specifications are included by reference except where superseded by normative statements herein:

- [WS-Discovery]

If a CLIENT and a SERVICE are not on the same subnet, the CLIENT may not be able to discover the SERVICE. However, if a CLIENT has an Endpoint Reference and transport address for a SERVICE through some other means, the CLIENT and SERVICE should be able to communicate within the scope of this profile.

R1013: A DEVICE MUST be a compliant WS-Discovery [WS-Discovery] Target Service.

R1001: A HOSTED SERVICE SHOULD NOT be a Target Service.

If each SERVICE were to participate in WS-Discovery, the network traffic generated by a relatively small number of DEVICEs hosting a relatively small number of HOSTED SERVICEs could overwhelm a bandwidth-limited network. Therefore, only DEVICEs act as Target Services.

R1019: A DEVICE MUST at least support the "http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01/rfc3986" and "http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01/strcmp0" Scope matching rules.

R1020: If a DEVICE includes Types in a Hello, Probe Match, or Resolve Match SOAP ENVELOPE, it MUST include the dpws:Device Type.

Including the dpws:Device Type indicates a DEVICE supports the Devices Profile, and indicates a CLIENT may retrieve metadata about the DEVICE and its relationship to any HOSTED SERVICEs using Get [WS-Transfer].

R1009: A DEVICE MUST at least support receiving Probe and Resolve SOAP ENVELOPEs and sending Hello and Bye SOAP ENVELOPEs over multicast UDP.

R1016: A DEVICE MUST at least support sending Probe Match and Resolve Match SOAP ENVELOPEs over unicast UDP.

R1018: A DEVICE MAY ignore a multicast UDP Probe or Resolve SOAP ENVELOPE if the [address] of the [reply endpoint] is not "http://www.w3.org/2005/08/addressing/anonymous".

WS-Discovery acknowledges that a CLIENT may include reply information in UDP Probe and Resolve SOAP ENVELOPEs to specify a transport other than SOAP over UDP. However, to establish a baseline for interoperability, DEVICEs are required only to support UDP responses.

R1015: A DEVICE MUST support receiving a Probe SOAP ENVELOPE as an HTTP Request at any HTTP transport address where the DEVICE endpoint is available.

R5021: A DEVICE MAY reject a unicast Probe SOAP ENVELOPE received as an HTTP Request if the [address] property of the [destination] is not "urn:docs-oasis-open:ws-dd:ns:discovery:2009:01".

To support the scenario where a DEVICE has a known HTTP transport address, a CLIENT may send an ad-hoc Probe over HTTP to that address and expect to receive a ProbeMatches response, using the same message pattern as defined by the ProbeOp operation of the DiscoveryProxy portType in [WS-Discovery]. This requirement does not imply that the DEVICE must perform as a Discovery Proxy.

How the client obtains the DEVICE HTTP address is not defined in this specification, and this HTTP address does not necessarily relate to HOSTED SERVICE addresses.

R1021: If a DEVICE matches a Probe SOAP ENVELOPE received as an HTTP Request, it MUST send a Probe Matches SOAP ENVELOPE response containing a Probe Match section representing the DEVICE.

R1022: If a DEVICE does not match a Probe SOAP ENVELOPE received as an HTTP Request, it MUST send a Probe Matches SOAP ENVELOPE response with no Probe Match sections.
R5022: If a DEVICE includes a Probe Match section as an HTTP Response as described in R1021, it MUST include all of its Types and Scopes in the Probe Match section.

DEVICEs may omit their Types and Scopes in their UDP WS-Discovery messages to reduce message size and prevent fragmentation. However, they are obligated to return all Types and Scopes in their HTTP ProbeMatches messages as increased risk of packet loss due to fragmentation is not a consideration.

4 Description

The scope of this section is the following set of Web services specifications. All of the requirements in these specifications are included by reference except where superseded by normative statements herein:

- [XML Schema Part 1, Part 2]
- [WSDL 1.1]
- [BP 1.1, Section 4]
- [WSDL Binding for SOAP 1.2]
- [WS-MetadataExchange]
- [WS-Policy]
- [WS-PolicyAttachment]
- [WS-Transfer]

A DEVICE acts primarily as a metadata resource for device-wide data, and for the HOSTED SERVICES on the device. A CLIENT retrieves the XML representation of these characteristics by sending a WS-Transfer Get SOAP ENVELOPE to the DEVICE. The resulting metadata contains characteristics of the device and topology information relating the DEVICE to its HOSTED SERVICEs. WS-Transfer Get is used here because the device-wide metadata is the XML representation of the DEVICE.

CLIENTs may also retrieve metadata for individual HOSTED SERVICEs by sending a WS-MetadataExchange GetMetadata SOAP ENVELOPE to the HOSTED SERVICE. The resulting metadata contains limited topology information about the HOSTED SERVICE, its hosting DEVICE, its WSDL, and any additional sections specific to the type of service. GetMetadata is used here because the XML representation of the HOSTED SERVICE (possibly accessible with WS-Transfer Get) is not defined.

Through WSDL, this description also conveys the MESSAGEs a HOSTED SERVICE is capable of receiving and sending. Through WS-Policy, description conveys the capabilities and requirements of a HOSTED SERVICE, particularly the transports over which it may be reached and its security capabilities.

R5007: A DEVICE MUST support receiving a WS-Transfer Get SOAP ENVELOPE using the HTTP binding defined in this profile.

R2044: In a Get Response SOAP ENVELOPE, a DEVICE MUST include only a wsx:Metadata element in the SOAP ENVELOPE Body.

All metadata from the device should be contained in the wsx:Metadata element in the Get Response.

R2045: A DEVICE MAY generate a wsa:ActionNotSupported SOAP Fault in response to a Put, Delete, or Create SOAP ENVELOPE.

A DEVICE is not required to support all of the operations defined in [WS-Transfer].

R5008: A HOSTED SERVICE MUST support receiving a WS-MetadataExchange GetMetadata SOAP ENVELOPE using the HTTP binding defined in this profile.

4.1 Characteristics

To express DEVICE characteristics that are typically fixed across all DEVICEs of the same model by their manufacturer, this profile defines extensible ThisModel metadata as follows:

```
<dpws:ThisModel ...>
    <dpws:Manufacturer xml:lang="..."? >xs:string</dpws:Manufacturer>+
    <dpws:ManufacturerUrl>xs:anyURI</dpws:ManufacturerUrl>?
    <dpws:ModelName xml:lang="..."? >xs:string</dpws:ModelName>+
    <dpws:ModelNumber>xs:string</dpws:ModelNumber>?
    <dpws:ModelUrl>xs:anyURI</dpws:ModelUrl>?
    <dpws:PresentationUrl>xs:anyURI</dpws:PresentationUrl>?
```

```
    ...
</dpws:ThisModel>
```

The following describes additional, normative constraints on the outline above:

dpws:ThisModel/ dpws:Manufacturer

Name of the manufacturer of the DEVICE. It MUST have fewer than MAX_FIELD_SIZE Unicode characters, SHOULD be localized, and SHOULD be repeated for each supported locale.

dpws:ThisModel/ dpws:ManufacturerUrl

URL to a Web site for the manufacturer of the DEVICE. It MUST have fewer than MAX_URI_SIZE octets.

dpws:ThisModel/ dpws:ModelName

User-friendly name for this model of device chosen by the manufacturer. It MUST have fewer than MAX_FIELD_SIZE Unicode characters, SHOULD be localized, and SHOULD be repeated for each supported locale.

dpws:ThisModel/ dpws:ModelNumber

Model number for this model of DEVICE. It MUST have fewer than MAX_FIELD_SIZE Unicode characters.

dpws:ThisModel/ dpws:ModelUrl

URL to a Web site for this model of DEVICE. It MUST have fewer than MAX_URI_SIZE octets.

dpws:ThisModel/ dpws:PresentationUrl

URL to a presentation resource for this DEVICE. It MAY be relative to the HTTP transport address over which metadata was retrieved, and MUST have fewer than MAX_URI_SIZE octets. If PresentationUrl is specified, the DEVICE MAY provide the resource in multiple formats, but MUST at least provide an HTML page. CLIENTs and DEVICEs MAY use HTTP content negotiation [HTTP/1.1] to determine the format and content of the presentation resource.

DEVICEs that use a relative URL MAY use HTTP Redirection 3xx codes [HTTP/1.1] to direct CLIENTs to a dedicated web server running on another port.

CORRECT:

```
<dpws:ThisModel
      xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01" >
    <dpws:Manufacturer>ACME Manufacturing</dpws:Manufacturer>
    <dpws:ModelName xml:lang="en-GB" >ColourBeam 9</dpws:ModelName>
    <dpws:ModelName xml:lang="en-US" >ColorBeam 9</dpws:ModelName>
</dpws:ThisModel>
```

A Dialect [WS-MetadataExchange] equal to "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/ThisModel" indicates an instance of the ThisModel metadata format.

No Identifier [WS-MetadataExchange] is defined for instances of the ThisModel metadata format.

R2038: A DEVICE MUST have one Metadata Section with Dialect equal to "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/ThisModel" for its ThisModel metadata.

R2012: In any Get Response SOAP ENVELOPE, a DEVICE MUST include the Metadata Section with Dialect equal to "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/ThisModel".

Get [WS-Transfer] is the interoperable means for a CLIENT to retrieve the resource representation data for a DEVICE – which includes the ThisModel metadata for a DEVICE. A DEVICE may also provide other means for a CLIENT to retrieve its ThisModel metadata.

R2001: If a DEVICE changes any of its ThisModel metadata, it MUST increment the Metadata Version exposed in Hello, Probe Match, and Resolve Match SOAP ENVELOPEs as wsd:MetadataVersion.

Caching for the ThisModel metadata is controlled by the wsd:MetadataVersion construct [WS-Discovery].

To express DEVICE characteristics that typically vary from one DEVICE to another of the same kind, this profile defines extensible ThisDevice metadata as follows:

```
<dpws:ThisDevice ...>
    <dpws:FriendlyName xml:lang="..."? >xs:string</dpws:FriendlyName>+
    <dpws:FirmwareVersion>xs:string</dpws:FirmwareVersion>?
    <dpws:SerialNumber>xs:string</dpws:SerialNumber>?
    ...
</dpws:ThisDevice>
```

The following describes additional, normative constraints on the outline above:

dpws:ThisDevice/dpws:FriendlyName

User-friendly name for this DEVICE. It MUST have fewer than MAX_FIELD_SIZE Unicode characters, SHOULD be localized, and SHOULD be repeated for each supported locale.

dpws:ThisDevice/dpws:FirmwareVersion

Firmware version for this DEVICE. It MUST have fewer than MAX_FIELD_SIZE Unicode characters.

dpws:ThisDevice/dpws:SerialNumber

Manufacturer-assigned serial number for this DEVICE. It MUST have fewer than MAX_FIELD_SIZE Unicode characters.

CORRECT:

```
<dpws:ThisDevice
      xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01" >
    <dpws:FriendlyName xml:lang="en-GB" >
      ACME ColourBeam Printer
    </dpws:FriendlyName>
    <dpws:FriendlyName xml:lang="en-US" >
      ACME ColorBeam Printer
    </dpws:FriendlyName>
</dpws:ThisDevice>
```

A Dialect [WS-MetadataExchange] equal to "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/ThisDevice" indicates an instance of the ThisDevice metadata format.

No Identifier [WS-MetadataExchange] is defined for instances of the ThisDevice metadata format.

R2039: A DEVICE MUST have a Metadata Section with Dialect equal to "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/ThisDevice" for its ThisDevice metadata.

R2014: In any Get Response SOAP ENVELOPE, a DEVICE MUST include the Metadata Section with Dialect equal to "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/ThisDevice".

CORRECT:

```
<soap:Envelope
    xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
    xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01"
    xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
    xmlns:wsa="http://www.w3.org/2005/08/addressing" >
 <soap:Header>
  <wsa:Action>
    http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse
  </wsa:Action>
  <wsa:RelatesTo>
    urn:uuid:82204a83-52f6-475c-9708-174fa27659ec
  </wsa:RelatesTo>
  <wsa:To>
    http://www.w3.org/2005/08/addressing/anonymous
  </wsa:To>
```

```
 </soap:Header>
 <soap:Body>
  <wsx:Metadata>
   <wsx:MetadataSection
Dialect="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/ThisModel"
   >
    <dpws:ThisModel>
     <dpws:Manufacturer>ACME Manufacturing</dpws:Manufacturer>
     <dpws:ModelName xml:lang="en-GB" >
       ColourBeam 9
     </dpws:ModelName>
     <dpws:ModelName xml:lang="en-US" >
       ColorBeam 9
     </dpws:ModelName>
    </dpws:ThisModel>
   </wsx:MetadataSection>
   <wsx:MetadataSection
Dialect="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/ThisDevice"
   >
    <dpws:ThisDevice>
     <dpws:FriendlyName xml:lang="en-GB" >
       ACME ColourBeam Printer
     </dpws:FriendlyName>
     <dpws:FriendlyName xml:lang="en-US" >
       ACME ColorBeam Printer
     </dpws:FriendlyName>
    </dpws:ThisDevice>
   </wsx:MetadataSection>

   <!-- Other Metadata Sections omitted for brevity. -->

  </wsx:Metadata>
 </soap:Body>
</soap:Envelope>
```

Get [WS-Transfer] is the interoperable means for a CLIENT to retrieve the resource representation data for a DEVICE – which includes the ThisDevice metadata for a DEVICE. A DEVICE may also provide other means for a CLIENT to retrieve its ThisDevice metadata.

R2002: If a DEVICE changes any of its ThisDevice metadata, it MUST increment the Metadata Version exposed in Hello, Probe Match, and Resolve Match SOAP ENVELOPEs as wsd:MetadataVersion.

Caching for the ThisDevice metadata is controlled by the wsd:MetadataVersion construct [WS-Discovery].

4.2 Hosting

To express the relationship between a HOSTED SERVICE and its hosting DEVICE, this profile defines relationship metadata as follows:

```
<dpws:Relationship Type="xs:anyURI" ... >
 (<dpws:Host>
    <wsa:EndpointReference>endpoint-reference</wsa:EndpointReference>
    <dpws:Types>list of xs:QName</dpws:Types>?
    ...
  </dpws:Host>)?
 (<dpws:Hosted>
    <wsa:EndpointReference>endpoint-reference</wsa:EndpointReference>+
    <dpws:Types>list of xs:QName</dpws:Types>
    <dpws:ServiceId>xs:anyURI</dpws:ServiceId>
```

```
    ...
  </dpws:Hosted>)*
  ...
</dpws:Relationship>
```

The following describes additional, normative constraints on the outline above:

dpws:Relationship

This is a general mechanism for defining a relationship between two or more SERVICEs.

dpws:Relationship/@Type

The type of the relationship. The nature of the relationship and the content of the dpws:Relationship element are determined by this value. This value should be compared directly, as a case-sensitive string, with no attempt to make a relative URI into an absolute URI, to unescape, or to otherwise canonicalize it.

dpws:Relationship/@Type = "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/host"

This is a specific, hosting relationship type to indicate the relationship between a HOSTED SERVICE and its hosting DEVICE. This relationship type defines the following additional content:

dpws:Relationship/dpws:Host

This is a section describing a hosting DEVICE. At least one of ./dpws:Host or ./dpws:Hosted MUST be included.

dpws:Relationship/dpws:Host/wsa:EndpointReference

Endpoint Reference for the host, which includes the stable identifier for the host which MUST be persisted across re-initialization (see also R0005 and R0006). If ./dpws:Host is omitted, implied value is the Endpoint Reference of the DEVICE that returned this metadata in a Get Response SOAP ENVELOPE.

dpws:Relationship/dpws:Host/dpws:Types

Unordered set of Types implemented by the host. (See [WS-Discovery].) If omitted or ./dpws:Host is omitted, no implied value.

dpws:Relationship/dpws:Hosted

This is a section describing a HOSTED SERVICE. . It MUST be included by a DEVICE for each of its HOSTED SERVICEs. It MUST be included by a HOSTED SERVICE for itself. For the hosting relationship type, if a host has more than one HOSTED SERVICE, including one relationship that lists all HOSTED SERVICEs is equivalent to including multiple relationships that each list some subset of the HOSTED SERVICEs.

dpws:Relationship/dpws:Hosted/wsa:EndpointReference

Endpoint References for a HOSTED SERVICE.

dpws:Relationship/dpws:Hosted/dpws:Types

Unordered set of Types implemented by a HOSTED SERVICE. All implemented Types SHOULD be included.

dpws:Relationship/dpws:Hosted/dpws:ServiceId

Identifier for a HOSTED SERVICE which MUST be persisted across re-initialization and MUST NOT be shared across multiple Hosted elements. ServiceId MUST be unique within a DEVICE. This value should be compared directly, as a case-sensitive string, with no attempt to make a relative URI into an absolute URI, to unescape, or to otherwise canonicalize it.

CORRECT:

```
<dpws:Relationship
    Type="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/host"
    xmlns:img="http://printer.example.org/imaging"
    xmlns:wsa="http://www.w3.org/2005/08/addressing"
    xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01" >
  <dpws:Hosted>
```

```
    <wsa:EndpointReference>
      <wsa:Address>http://172.30.184.244/print</wsa:Address>
    </wsa:EndpointReference>
    <dpws:Types>
      img:PrintBasicPortType img:PrintAdvancedPortType
    </dpws:Types>
    <dpws:ServiceId>
      http://printer.example.org/imaging/PrintService
    </dpws:ServiceId>
  </dpws:Hosted>
</dpws:Relationship>
```

A Dialect [WS-MetadataExchange] equal to "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/Relationship" indicates an instance of the Relationship metadata format.

No Identifier [WS-MetadataExchange] is defined for instances of the Relationship metadata format.

R2040: If a DEVICE has any HOSTED SERVICEs, it MUST have at least one Metadata Section with Dialect equal to "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/Relationship" for its Relationship metadata.

R2029: In any Get Response SOAP ENVELOPE, a DEVICE MUST include any Metadata Section(s) with Dialect equal to "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/Relationship".

Get [WS-Transfer] is the interoperable means for a CLIENT to retrieve the resource representation data for a DEVICE – which includes the relationship metadata for itself and HOSTED SERVICEs.

R5020: A HOSTED SERVICE MUST have one Metadata Section with http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/Relationship".

GetMetadata [WS-MetadataExchange] is the interoperable means for a CLIENT to retrieve metadata for a HOSTED SERVICE – which includes the relationship metadata for itself and its hosting DEVICE.

A DEVICE or HOSTED SERVICE may provide other means for a CLIENT to retrieve its relationship metadata.

CORRECT:

```
<soap:Envelope
    xmlns:gen="http://example.org/general"
    xmlns:img="http://printer.example.org/imaging"
    xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
    xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01"
    xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
    xmlns:wsa="http://www.w3.org/2005/08/addressing" >
  <soap:Header>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse
    </wsa:Action>
    <wsa:RelatesTo>
      urn:uuid:82204a83-52f6-475c-9708-174fa27659ec
    </wsa:RelatesTo>
    <wsa:To>
      http://www.w3.org/2005/08/addressing/anonymous
    </wsa:To>
  </soap:Header>
  <soap:Body>
    <wsx:Metadata>
      <wsx:MetadataSection
        Dialect
      ="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/Relationship"
      >
        <dpws:Relationship
```

```
          Type="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/host" >
          <dpws:Hosted>
            <wsa:EndpointReference>
              <wsa:Address>http://172.30.184.244/print</wsa:Address>
            </wsa:EndpointReference>
            <wsa:EndpointReference>
              <wsa:Address>http://[fdaa:23]/print1</wsa:Address>
            </wsa:EndpointReference>
            <dpws:Types>
              img:PrintBasicPortType img:PrintAdvancedPortType
            </dpws:Types>
            <dpws:ServiceId>
              http://printer.example.org/imaging/PrintService
            </dpws:ServiceId>
          </dpws:Hosted>
          <dpws:Hosted>
            <wsa:EndpointReference>
              <wsa:Address>http://172.30.184.244/scan</wsa:Address>
            </wsa:EndpointReference>
            <wsa:EndpointReference>
              <wsa:Address>http://[fdaa:24]/scan</wsa:Address>
            </wsa:EndpointReference>
            <dpws:Types>img:ScanBasicPortType</dpws:Types>
            <dpws:ServiceId>
              http://printer.example.org/imaging/ScanService
            </dpws:ServiceId>
          </dpws:Hosted>
        </dpws:Relationship>
      </wsx:MetadataSection>

      <!-- Other Metadata Sections omitted for brevity. -->

    </wsx:Metadata>
  </soap:Body>
</soap:Envelope>
```

R2030: If a DEVICE changes any of its relationship metadata, it MUST increment the Metadata Version exposed in Hello, Probe Match, and Resolve Match SOAP ENVELOPEs as wsd:MetadataVersion.

Caching for relationship metadata is controlled by the wsd:MetadataVersion construct [WS-Discovery].

R2042: A DEVICE MUST NOT change its relationship metadata based on temporary changes in the network availability of the SERVICEs described by the metadata.

Relationship metadata is intended to model fairly static relationships and should not change if a SERVICE becomes temporarily unavailable. As in the general case, any CLIENT attempting to contact such a SERVICE will need to deal with an Endpoint Unavailable Fault [WS-Addressing], connection refusal, or other network indication that the SERVICE is unavailable.

4.3 WSDL

R2004: If a HOSTED SERVICE exposes Notifications, its portType MUST include Notification and/or Solicit-Response Operations describing those Notifications.

R2004 relaxes R2303 in [BP 1.1, Section 4].

R2019: A HOSTED SERVICE MUST at least include a document-literal Binding for SOAP 1.2 over HTTP for each portType in its WSDL.

Because the document-literal SOAP Binding is more general than an rpc-literal SOAP Binding, there is no requirement to use anything other than the document-literal Binding.

R2028: A HOSTED SERVICE is not required to include any WSDL bindings for SOAP 1.1 in its WSDL.

Since this profile brings SOAP 1.2 into scope, it is sufficient to bind to that version of SOAP. There is no requirement to bind to other SOAP versions and thus R2028 updates R2401 in [BP 1.1, Section 4] to SOAP 1.2.

Addressing information for a HOSTED SERVICE is included in relationship metadata. For the mandatory SOAP 1.2 binding (R2019), there is no requirement to re-express this information in a WSDL Service and Port, since the endpoint reference used in the relationship metadata refers to this binding by default. The use of WSDL Services and Ports may still be necessary for other bindings not covered by this profile.

R2023: If a HOSTED SERVICE receives a MESSAGE that is inconsistent with its WSDL description, the HOSTED SERVICE SHOULD generate a SOAP Fault with a Code Value of "Sender", unless a "MustUnderstand" or "VersionMismatch" Fault is generated.

R2024: If a HOSTED SERVICE receives a MESSAGE that is inconsistent with its WSDL description, the HOSTED SERVICE MUST check for "VersionMismatch", "MustUnderstand", and "Sender" fault conditions in that order.

Statements R2023 and R2024 update R2724 and R2725 [BP 1.1, Section 4] to SOAP 1.2.

R2031: A HOSTED SERVICE MUST have at least one Metadata Section with Dialect="http://schemas.xmlsoap.org/wsdl/".

For clarity, separation of levels of abstraction, and/or reuse of standardized components, WSDL may be authored in a style that separates different elements of a Service Definition into separate documents which may be imported or included as needed. Each separate document may be available at the URL in the xs:include/@schemaLocation, xs:import/@schemaLocation, or wsdl:import/@location or may be included in a separate XML Schema or WSDL Metadata Section.

GetMetadata [WS-MetadataExchange] is the interoperable means for a CLIENT to retrieve metadata for a HOSTED SERVICE – which includes the WSDL for a HOSTED SERVICE. A HOSTED SERVICE may provide other means for a CLIENT to retrieve its WSDL.

There is no requirement for a HOSTED SERVICE to store its WSDL and include it in-line in a Get Response SOAP ENVELOPE. The WSDL may be stored at a different location, and the HOSTED SERVICE may include a reference to it in a Get Response SOAP ENVELOPE.

CORRECT:

```
<soap:Envelope
    xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
    xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
    xmlns:wsa="http://www.w3.org/2005/08/addressing" >
  <soap:Header>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Response
    </wsa:Action>
    <wsa:RelatesTo>
      urn:uuid:82204a83-52f6-475c-9708-174fa27659ec
    </wsa:RelatesTo>
    <wsa:To>
      http://www.w3.org/2005/08/addressing/anonymous
    </wsa:To>
  </soap:Header>
  <soap:Body>
    <wsx:Metadata>
      <wsx:MetadataSection
          Dialect="http://schemas.xmlsoap.org/wsdl" >
        <wsx:MetadataReference>
          <wsa:Address>http://172.30.184.244/print</wsa:Address>
```

```
          <wsa:ReferenceParameters>
            <x:Acme xmlns:x="urn:acme.com:webservices">
              WSDL
            </x:Acme>
          </wsa:ReferenceParameters>
        </wsx:MetadataReference>
      </wsx:MetadataSection>

      <!-- Other Metadata Sections omitted for brevity. -->

    </wsx:Metadata>
  </soap:Body>
</soap:Envelope>
```

4.4 WS-Policy

To indicate that a SERVICE is compliant with this profile, this profile defines the following WS-Policy [WS-Policy] assertion:

```
<dpws:Profile wsp:Optional="true"? ... />
```

The following describes additional, normative constraints on the outline above:

dpws:Profile

Assertion indicating compliance with this profile is required. This assertion has Endpoint Policy Subject [WS-PolicyAttachment]: a policy expression containing this assertion MAY be attached to a wsdl:port, SHOULD be attached to a wsdl:binding, but MUST NOT be attached to a wsdl:portType; the latter is prohibited because the assertion specifies a concrete behavior whereas the wsdl:portType is an abstract construct.

dpws:Profile/@wsp:Optional="true"

Per WS-Policy [WS-Policy], this is compact notation for two policy alternatives, one with and one without the assertion. The intuition is that the behavior indicated by the assertion is optional, or in this case, that the SERVICE supports but does not require compliance with this profile.

CORRECT:

```
<wsp:Policy
    xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01"
    xmlns:wsp="http://www.w3.org/ns/ws-policy" >
  <dpws:Profile />
</wsp:Policy>
```

R2037: A SERVICE MUST include the dpws:Profile assertion in its policy.

This assertion has Endpoint Policy Subject: a policy expression containing this assertion MAY be attached to a wsdl:port, SHOULD be attached to a wsdl:binding, but MUST NOT be attached to a wsdl:portType; the latter is prohibited because this assertion specifies concrete behavior whereas the wsdl:portType is an abstract construct.

R2041: If a SERVICE uses wsp:PolicyReference/@URI to attach a policy identified by an absolute IRI, the SERVICE MUST have a Metadata Section with Dialect equal to "http://www.w3.org/ns/ws-policy" and Identifier equal to that IRI.

R2025: If a SERVICE uses wsp:PolicyReference/@URI to attach a policy identified by an absolute IRI, then in a Get Response SOAP ENVELOPE, the SERVICE MUST include the Metadata Section with Dialect equal to"http://www.w3.org/ns/ws-policy" and Identifier equal to that IRI.

R2035: If a SERVICE uses wsp:PolicyReference/@URI to attach a policy identified by a relative IRI, the SERVICE MUST embed that policy as a child of wsdl:definitions, and the policy MUST have a @wsu:Id containing that IRI.

R2036: A SERVICE MUST NOT use @wsp:PolicyURIs to attach policy.

Because all components in WSDL are extensible via elements [BP 1.1, Section 4], attachment using wsp:PolicyReference/@URI is sufficient.

Get [WS-Transfer] is the interoperable means for a CLIENT to retrieve attached policy.

CORRECT:

```
<soap:Envelope
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01"
  xmlns:wsp="http://www.w3.org/ns/ws-policy"
  xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
  xmlns:wsa="http://www.w3.org/2005/08/addressing" >
 <soap:Header>
  <wsa:Action>
    http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse
  </wsa:Action>
  <wsa:RelatesTo>
    urn:uuid:82204a83-52f6-475c-9708-174fa27659ec
  </wsa:RelatesTo>
  <wsa:To>
    http://www.w3.org/2005/08/addressing/anonymous
  </wsa:To>
 </soap:Header>
 <soap:Body>
  <wsx:Metadata>
   <wsx:MetadataSection
     Dialect="http://schemas.xmlsoap.org/wsdl/" >
    <wsdl:definitions
      targetNamespace="http://acme.example.com/colorbeam"
      xmlns:image="http://printer.example.org/imaging" >
     <wsp:Policy wsu:Id="DpPolicy" >
      <dpws:Profile />
     </wsp:Policy>

     <!-- Other WSDL components omitted for brevity. -->

     <wsdl:binding name="PrintBinding" type="image:PrintPortType" >
       <wsp:PolicyReference URI="#DpPolicy"
           wsdl:required="true" />
       <!-- Other WSDL components omitted for brevity. -->
     </wsdl:binding>
    </wsdl:definitions>
   </wsx:MetadataSection>

   <!-- Other Metadata Sections omitted for brevity. -->

  </wsx:Metadata>
 </soap:Body>
</soap:Envelope>
```

5 Eventing

The scope of this section is the following set of Web services specifications. All of the requirements in these specifications are included by reference except where superseded by normative statements herein:

- [WS-Eventing]

5.1 Subscription

R3009: A HOSTED SERVICE MUST at least support Push Delivery Mode indicated by "http://schemas.xmlsoap.org/ws/2004/08/eventing/DeliveryModes/Push".

The Push Delivery Mode [WS-Eventing] is the default Delivery Mode and indicates the Event Source (HOSTED SERVICE) will push Notifications to the Event Sink (CLIENT).

R3017: If a HOSTED SERVICE does not understand the [address] of the Notify To of a Subscribe SOAP ENVELOPE, the HOSTED SERVICE MUST generate a wsa:DestinationUnreachable SOAP Fault in place of a SubscribeResponse message.

R3018: If a HOSTED SERVICE does not understand the [address] of the End To of a Subscribe SOAP ENVELOPE, the HOSTED SERVICE MUST generate a wsa:DestinationUnreachable SOAP Fault in place of a SubscribeResponse message.

R3017 and R3018 do not ensure that a HOSTED SERVICE can contact an event sink, but they do provide a mechanism for the event source to fault on unsupported URI schemes or addresses it knows it cannot contact.

R5003: If a HOSTED SERVICE generates a wsa:DestinationUnreachable SOAP Fault under R3017 or R3018, the SOAP Fault Detail MUST be the EndTo or NotifyTo Endpoint Reference Address that the HOSTED SERVICE did not understand.

R5003 allows a client to distinguish between a SOAP Fault generated due to an unreachable [destination] information header in the Subscribe message, and a SOAP Fault generated due to an unreachable NotifyTo or EndTo address.

R3019: If a HOSTED SERVICE cannot deliver a Notification SOAP ENVELOPE to an Event Sink, the HOSTED SERVICE MAY terminate the corresponding Subscription.

R5004: If a HOSTED SERVICE terminates a subscription (per R3019), the HOSTED SERVICE SHOULD send a Subscription End SOAP ENVELOPE with a Status of "http://schemas.xmlsoap.org/ws/2004/08/eventing/DeliveryFailure".

5.1.1 Filtering

To enable subscribing to one or more Notifications exposed by a HOSTED SERVICE, this profile defines a Filter Dialect designated "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/Action".

- A Filter in this Dialect contains a white space-delimited list of URIs that indicate the [action] property of desired Notifications.
- The content of a Filter in this Dialect is defined as xs:list/@itemType="xs:anyURI" [XML Schema Part 2].
- A Filter in this Dialect evaluates to true for an Output Message of a Notification or Solicit-Response operation if and only if a URI in the Filter matches the [action] property of the Message using the "http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01/rfc3986" matching rule [WS-Discovery].
- A Filter in this Dialect with no URIs specified will always evaluate to false for all messages.

The Action Dialect uses the RFC 3986 prefix matching rule so CLIENTs can subscribe to a related set of Notifications by including the common prefix of the [action] property of those Notifications. Typically, the

Notifications within a WSDL portType [WSDL 1.1] will share a common [action] property prefix, and specifying that prefix with the Action Dialect will be a convenient means to subscribe to all Notifications defined by a portType.

R3008: A HOSTED SERVICE MUST at least support Filtering by the Dialect "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/Action".

CORRECT:

```
<soap:Envelope
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing" >
 <soap:Header>
  <wsa:Action>
    http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe
  </wsa:Action>
  <wsa:MessageID>
    urn:uuid:314bea3b-03af-47a1-8284-f495497f1e33
  </wsa:MessageID>
  <wsa:ReplyTo>
   <wsa:Address>
     http://www.w3.org/2005/08/addressing/anonymous
   </wsa:Address>
  </wsa:ReplyTo>
  <wsa:To>http://172.30.184.244/print</wsa:To>
 </soap:Header>
 <soap:Body>
  <wse:Subscribe>
   <wse:Delivery>
    <wse:NotifyTo>
     <wsa:Address>
       urn:uuid:3726983d-02de-4d41-8207-d028ae92ce3d
     </wsa:Address>
    </wse:NotifyTo>
   </wse:Delivery>
   <wse:Expires>PT10M</wse:Expires>
   <wse:Filter
Dialect="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/Action"
   >
http://printer.example.org/imaging/PrintBasicPortType/JobEndState
http://printer.example.org/imaging/PrintBasicPortType/PrinterState
   </wse:Filter>
  </wse:Subscribe>
 </soap:Body>
</soap:Envelope>
```

R3011: A HOSTED SERVICE MUST NOT generate a wse:FilteringNotSupported SOAP Fault in response to a Subscribe SOAP ENVELOPE.

A HOSTED SERVICE must support filtering, at least by [action], so the Filtering Not Supported SOAP Fault is not appropriate.

To indicate that a HOSTED SERVICE does not expose any Notifications that would match the contents of a Filter with the Action Dialect, this profile defines the following SOAP Fault:

[action]	http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/fault
[Code]	Soap:Sender
[Subcode]	dpws:FilterActionNotSupported

[Reason]	E.g., “no notifications match the supplied filter”
[Detail]	(None defined.)

R3020: If none of the Notifications exposed by a HOSTED SERVICE match the [action] values in a Subscribe SOAP ENVELOPE Filter whose Dialect is "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/Action", the HOSTED SERVICE SHOULD generate a dpws:FilterActionNotSupported SOAP Fault.

5.2 Subscription Duration and Renewal

R3016: A HOSTED SERVICE MUST NOT generate a wse:UnsupportedExpirationType SOAP Fault in response to a Subscribe or Renew SOAP ENVELOPE with an Expiration type of xs:duration.

R3013: A HOSTED SERVICE MAY generate a wse:UnsupportedExpirationType SOAP Fault in response to a Subscribe or Renew SOAP ENVELOPE with an Expiration of type xs:dateTime.

Event Sources are required to have an internal clock, but there is no requirement that the clock be synchronized with clients or other HOSTED SERVICEs. Event Sources are only required to support Expirations expressed in duration, but they should attempt to match the type of the Subscription Expiration when possible. If the value or type of the Expiration is unacceptable, the Event Source may select an appropriate Expiration and return it in the Subscribe Response or Renew Response.

R3015: A HOSTED SERVICE MAY generate a wsa:ActionNotSupported SOAP Fault in response to a Get Status SOAP ENVELOPE.

Event Sources are not required to support retrieving subscription status.

6 Security

This section defines a RECOMMENDED baseline for interoperable security between a DEVICE and a CLIENT. A DEVICE (or CLIENT) is free to support other security mechanisms in place of this mechanism as specified by WSDL [WSDL 1.1], policies [WS-Policy], or by other means.

In the absence of an explicit indication stating that a different security mechanism is to be used, the default security mechanism is determined by the transport addresses of the DEVICE: HTTP transport addresses (URLs) indicate the device supports no security, and HTTPS transport addresses indicate the device supports the security profile defined in this section.

A DEVICE may support at most one security profile.

This scope of this section is the following set of Web services specifications. All of the requirements in these specifications are included by reference except where superseded by normative statements herein:

- [AES/TLS]
- [HTTP Authentication]
- [SHA]
- [TLS]
- [RFC 4122]
- [X.509.v3]
- [WS-Discovery]

6.1 Terminology

Compact Signature

> A WS-Discovery Compact Signature [WS-Discovery] is evidence of authenticity of the unencrypted contents of a WS-Discovery message. The Compact Signature is included inside the unencrypted message.

Secure Channel

> A Secure Channel is a point-to-point transport-level TLS/SSL connection established between a CLIENT and a SERVICE. Messages transmitted through a Secure Channel receive some security protection, but that protection does not extend beyond the CLIENT and SERVICE that established the channel.

6.2 Model

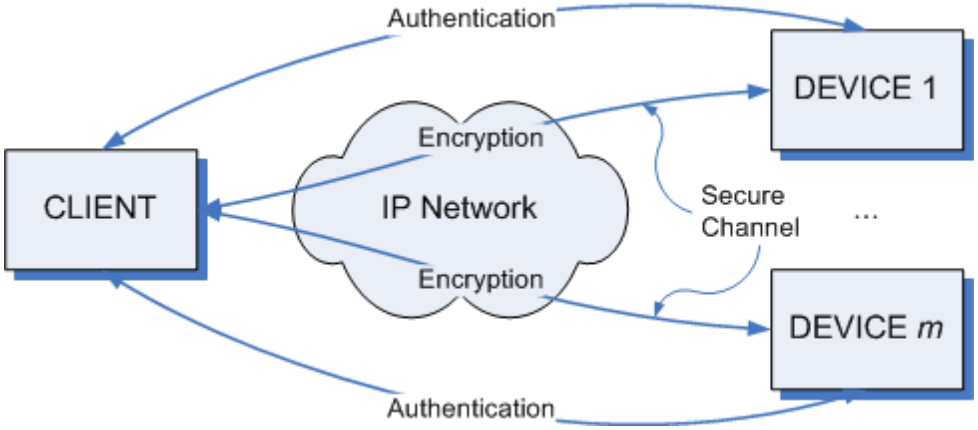
The security profile defined in this section has two parts: optional message-level signatures for UDP WS-Discovery traffic, and mandatory transport-level encryption for metadata and control traffic.

WS-Discovery Compact Signatures allow a CLIENT to verify the integrity of multicast or unicast WS-Discovery messages, and to identify WS-Discovery traffic that was signed by a DEVICE with a specific cryptographic credential.

TLS/SSL is used to establish a point-to-point Secure Channel between a CLIENT and a DEVICE, and provides a mechanism for each participant to authenticate the identity of the other, and to verify the integrity of the exchanged messages. It also provides confidentiality for all messages sent in the Secure Channel established between the CLIENT and the DEVICE.

A DEVICE uses an x.509.v3 certificate as its credential, and it uses this credential to sign WS-Discovery messages and to establish TLS/SSL connections. A DEVICE may require CLIENT authentication in the form of x.509.v3 certificates negotiated in the TLS/SSL connection, or username/password credentials communicated through HTTP Authentication after the TLS/SSL connection is established.

A DEVICE uses TLS/SSL to secure its HTTP traffic, and HOSTED SERVICES may also use TLS/SSL to secure their HTTP traffic. A DEVICE may use a physical HTTPS address, or a logical address and HTTPS xAddrs. If a DEVICE and its HOSTED SERVICEs are all reachable at the same address and port, a CLIENT and DEVICE may reuse a TLS/SSL connection for multiple operations.



The organization of CLIENT and DEVICE credentials, mechanism for provisioning them, and criteria for distinguishing valid and invalid credentials is out of scope of this profile.

6.3 Integrity

Integrity is the process that protects MESSAGEs against tampering while in transit. Integrity MUST adhere to the following requirements:

R5015: If a SERVICE uses TLS/SSL or WS-Discovery Compact Signatures, it MUST provide Integrity (as defined in this section) for any TLS/SSL connections or signatures, respectively.
R4000: A SERVICE MUST not send a SOAP ENVELOPE without protecting the integrity of any Message Information Header blocks matching the following XPath expressions: (a) /soap:Envelope/soap:Header/wsa:Action, (b) /soap:Envelope/soap:Header/wsa:MessageID, (c) /soap:Envelope/soap:Header/wsa:To, (d) /soap:Envelope/soap:Header/wsa:ReplyTo, (e) /soap:Envelope/soap:Header/wsa:RelatesTo, and (f) /soap:Envelope/soap:Header/[@isReferenceParameter='true'].*
R4063: A SERVICE MAY reject a SOAP ENVELOPE that has unprotected Message Information Header blocks.
R4001: A SERVICE MUST not send a SOAP ENVELOPE (including SOAP Faults) without protecting the integrity of the SOAP ENVELOPE Body in conjunction with any Message Information Block(s) from R4000.
R4064: A SERVICE MAY reject a SOAP ENVELOPE that does not protect the integrity of the SOAP ENVELOPE Body.

In this profile, the integrity of UDP discovery SOAP ENVELOPEs is protected using message-level signatures, while the integrity of other MESSAGEs is protected using a Secure Channel.

6.4 Confidentiality

Confidentiality is the process by which sensitive information is protected against unauthorized disclosure while in transit. Confidentiality MUST adhere to the following requirements:

R5016: If a SERVICE uses TLS/SSL, it MUST provide Confidentiality (as defined in this section) for any TLS/SSL connections.
R4002: A SERVICE MUST NOT send a SOAP ENVELOPE without encrypting the SOAP ENVELOPE Body.

R4067: A SERVICE MAY reject a SOAP ENVELOPE that does not encrypt the SOAP ENVELOPE Body.

In this profile, UDP WS-Discovery MESSAGEs are not treated as confidential. Confidential MESSAGEs are encrypted using a Secure Channel.

6.5 Authentication

Authentication is the process by which the identity of the sender is determined by the recipient. Authentication MUST adhere to the following requirements:

R4004: A SENDER MUST authenticate itself to a RECEIVER using credentials acceptable to the RECEIVER.

In this profile, authentication is done using certificates or a combination of certificates and HTTP authentication. If multicast messages are secured, the following additional requirements apply:

R4005: On multicast MESSAGEs, a CLIENT MUST use an authentication credential that is suitable for all DEVICEs that could legitimately process the multicast MESSAGE.

R5023: If a SERVICE uses TLS/SSL, it MUST provide Authentication (as defined in this section) for any TLS/SSL connections.

6.6 Trust

The distribution of the credentials needed for establishing the trust relationship is out of the scope of this profile.

R4008: A SERVICE MAY use additional mechanisms to verify the authenticity of the SENDER of any received MESSAGE by analyzing information provided by the lower networking layers.

For example, a SERVICE may authenticate only CLIENTs whose IP address exists in a preconfigured list.

6.7 DEVICE Behavior

R4014: A DEVICE MAY require authentication of a CLIENT.

R4017: A CLIENT MAY ignore MESSAGEs received during discovery that have no signature or a nonverifiable signature.

R4018: A DEVICE SHOULD cache authentication information for a CLIENT as valid as long as the DEVICE is connected to the CLIENT.

R5009: If a DEVICE uses a physical transport address for the [address] property of its Endpoint Reference, it MUST be an HTTPS scheme IRI.

R5010: A SERVICE MAY use an HTTP scheme IRI for the [address] property of its Endpoint Reference.

6.8 Security for Discovery

In the discovery phase, the client learns of the existence of the device on the network. Subsequently, the identity of the device is verified, and the device is connected to the client.

R5011: A DEVICE SHOULD sign its UDP discovery traffic using WS-Discovery Compact Signatures [WS-Discovery] to provide CLIENTs with a mechanism to verify the integrity of the messages, and to authenticate the DEVICE as the signor of the messages.

WS-Discovery Compact Signatures use WS-Security [WS-Security] to generate a cryptographic signature that can be used by a CLIENT to verify the validity of the unencrypted message.

In cases where CLIENTs persist enough information about the credentials and presence of security on a DEVICE to protect against impersonation, the DEVICE may not sign its discovery messages.

R5012: A DEVICE MUST NOT advertise HTTP scheme addresses the xAddrs fields of WS-Discovery messages.

Probe

A CLIENT initiates the discovery process by probing the network for a DEVICE it is interested in.

R4032: A DEVICE MUST NOT send a Probe Match SOAP ENVELOPE if the DEVICE is outside the local subnet of the CLIENT, and the Probe SOAP ENVELOPE was sent using the multicast binding as defined in WS-Discovery section 2.4.

R4065: A CLIENT MUST discard a Probe Match SOAP ENVELOPE if it is received MATCH_TIMEOUT seconds or more later than the last corresponding Probe SOAP ENVELOPE was sent.

Resolve

R4036: A DEVICE MUST NOT send a Resolve Match SOAP ENVELOPE if the DEVICE is outside the local subnet of the CLIENT, and the Resolve SOAP ENVELOPE was sent using the multicast binding as defined in WS-Discovery section 2.4

R4066: A CLIENT MUST discard a Resolve Match SOAP ENVELOPE if it is received MATCH_TIMEOUT seconds or more later than the last corresponding Resolve SOAP ENVELOPE was sent.

6.9 Authentication

The authentication step that follows discovery verifies the credentials of the DEVICE and CLIENT in a secure manner. Credentials may be cached on the DEVICE and/or CLIENT to simplify subsequent authentications.

6.9.1 Transport Layer Security (TLS/SSL)

TLS/SSL provides mutual authentication of CLIENT and DEVICE as well as the establishment of a Secure Channel over which MESSAGEs are exchanged in a secure manner.

R4039: A CLIENT MUST initiate authentication with the DEVICE by setting up a TLS/SSL session.

R4042: Following the establishment of a TLS/SSL Secure Channel, subsequent MESSAGE exchanges over HTTP SHOULD use the existing TLS/SSL session.

6.9.2 Certificates

R4043: Each DEVICE SHOULD have its own, unique Certificate.

The Certificate contains information pertinent to the specific device including its public key. Typically, certificates are issued by a trusted authority or a delegate (2nd tier) or a delegate of the delegate.

R4045: The format of the certificate MUST follow the common standard X.509v3.

An example of a self-signed X.509 certificate is shown below. in this case, the Subject field contains the UUID in string representation format (i.e., not represented numerically).

Type	Element	Usage	Example
Basic Elements	Version	TLS	3
	Certificate Serial Number		1234567
	Certificate Algorithm Identifier		RSA
	Issuer		a7731471-4b54-4a64-942c-7d481dcb9614
	Validity Period		11/09/2001 - 01/07/2015
	Subject	UUID	a7731471-4b54-4a64-942c-7d481dcb9614

	Subject Public Key Information		rsaEncryption 1024 10888232e76740bd873462ea2c64ca1d a6f9112656a34b949d32cede0e476547 84ba0f7e62e143429d3217ee45ce5304 308e65a6eee6474cb4d9a3c0295c8267 761661ccba7546a09d5f03a8ea3b1160 dac9fb6e6ba94e54b6c8ee892e492f4c e3a96bbd9d7b4c4bb98b7c052ff361ba cee01718122c4f0d826efc123bb1b03d
Extensions	Extended Key Usage	Server Authentication	1.3.6.1.5.5.7.3.1
		Client Authentication	1.3.6.1.5.5.7.3.2
Signature	Certificate Authority's Digital Signature		5938f9908916cca32321916a184a6e75 2becb14fb99c4f33a03b03c3c752117c 91b8fb163d3541fca78bca235908ba69 1f7e36004a2d499a8e23951bd8af961d 36be05307ec34467a7c66fbb7fb5e49c 25e8dbdae4084ca9ba244b5bc1a377e5 262b9ef543ce47ad8a6b1d28c9138d0a dc8f5e3b469e42a5842221f9cf0a50d1

Certificate management is out of the scope of this profile.

6.9.3 DEVICE Authentication with TLS/SSL

X.509 certificates are the only mechanism for a CLIENT to authenticate a DEVICE or a HOSTED SERVICE (if TLS/SSL is supported on that HOSTED SERVICE).

R5017: If a SERVICE uses TLS/SSL, it MUST authenticate itself to a CLIENT by supplying an X.509v3 certificate during the TLS/SSL handshake.

6.9.4 CLIENT Authentication with TLS/SSL

R4071: If the CLIENT and the SERVICE exchanged certificates during the TLS/SSL handshake, and the SERVICE as well as the CLIENT were able to verify the certificates, the CLIENT and SERVICE are mutually authenticated, and no further steps SHALL be required.

R4046: A SERVICE MAY require HTTP Authentication step after the TLS/SSL handshake, if the SERVICE was not able to verify the certificate, or if the CLIENT did not provide a certificate during the TLS/SSL handshake.

X.509 certificates are the preferred mechanism for authenticating a client, but in cases where x.509 client certificates are unavailable or where validation is infeasible, the DEVICE may use HTTP Authentication to request client credentials.

R4048: If the HTTP authentication is successful, and the CLIENT presents a certificate to the SERVICE, the SERVICE SHOULD cache the certificate in its local certificate store of trusted certificates for future authentication of the CLIENT.

R4048 avoids the need for HTTP authentication for subsequent connections.

6.9.5 CLIENT Authentication with HTTP Authentication

HTTP authentication requires credentials in the form of username and password. It is assumed that how the CLIENT and SERVICE share knowledge of the username and password is out-of-band and beyond the scope of this profile.

Because the authentication is performed over the Secure Channel established during TLS/SSL handshake and after the CLIENT has authenticated the SERVICE, HTTP Basic authentication may be used safely.

R4050: If a SERVICE requires HTTP authentication, the SERVICE SHALL challenge the CLIENT using the HTTP 401 response code.
R4051: A CLIENT MUST authenticate using one of the options listed in the HTTP-Authenticate header.
R4052: HTTP Authentication MUST use the following parameters for username and password of the HTTP Request: UserName, PIN / Password.

The UserName is supplied to the SERVICE during HTTP authentication and MAY be used for establishing multiple access control classes, such as administrators, users, and guests. The naming and use of UserName is implementation-dependent and out of the scope of this profile.

R4053: If no UserName is provided, "admin" SHALL be used as the default UserName.

The purpose of the PIN / Password is to authenticate the CLIENT to the DEVICE during the HTTP authentication.

R4054: The RECOMMENDED size of a PIN / Password is at least 8 characters using at least a 32 character alphabet.
R4055: The PIN / Password that is unique to the SERVICE SHALL be conveyed to the CLIENT out-of-band. The methods of conveying the PIN out-of-band are out of the scope of this profile.
R4056: To reduce the attack surface, the SERVICE and CLIENT MAY limit the number of failed authentication attempts as well as the time interval successive attempts are made for one TLS/SSL session.

6.10 Secure Channel

A Secure Channel at the transport level is used to secure traffic between CLIENT and SERVICE.

R4057: All secure communication for Description, Control, and Eventing between the CLIENT and SERVICE MUST use a Secure Channel.
R4072: A SERVICE MUST support receiving and responding to a Probe SOAP ENVELOPE over HTTP using a Secure Channel.
R4073: A SERVICE MAY ignore a Probe SOAP ENVELOPE sent over HTTP that does not use a Secure Channel.
R5013: A CLIENT MAY use a Secure Channel to contact multiple SERVICEs if they can be reached at the same address and port. As prescribed by R1015, a CLIENT may send a Probe over HTTP; this Probe and ProbeMatches are sent using the Secure Channel.

6.11 TLS/SSL Ciphersuites

R4059: It is the responsibility of the sender to convert the embedded URL to use HTTPS as different transport security mechanisms can be negotiated.
R4060: A SERVICE MUST support the following TLS Ciphersuite: TLS_RSA_WITH_RC4_128_SHA.
R4061: It is recommended that a SERVICE also support the following TLS Ciphersuite: TLS_RSA_WITH_AES_128_CBC_SHA.
R4062: Additional Ciphersuites MAY be supported. They are negotiated during the TLS/SSL handshake.

Where appropriate, DEVICEs are encouraged to support additional Ciphersuites that rely on more robust security technology, such as the SHA-2 [SHA] family of hashing standards.

R5014: A SERVICE SHOULD NOT negotiate any of the following TLS/SSL Ciphersuites: (a) TLS_RSA_WITH_NULL_SHA, (b) SSL_RSA_WITH_NULL_SHA, (c) any Ciphersuite with DH_anon in their symbolic name, (d) any Ciphersuites with MD5 in their symbolic name.

7 Conformance

An endpoint MAY implement more than one of the roles defined herein. An endpoint is not compliant with this specification if it fails to satisfy one or more of the MUST or REQUIRED level requirements defined herein for the roles it implements.

Normative text within this specification takes precedence over normative outlines, which in turn take precedence over the XML Schema [XML Schema Part 1, Part 2] descriptions, which in turn take precedence over examples.

A. Acknowledgements

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

Geoff Bullen, Microsoft Corporation
Steve Carter, Novell
Dan Conti, Microsoft Corporation
Doug Davis, IBM
Scott deDeugd, IBM
Dan Driscoll, Microsoft Corporation
Colleen Evans, Microsoft Corporation
Max Feingold, Microsoft Corporation
Travis Grigsby, IBM
Francois Jammes, Schneider Electric
Ram Jeyaraman, Microsoft Corporation
Mike Kaiser, IBM
Supun Kamburugamuva, WSO2
Devon Kemp, Canon Inc.
Akira Kishida, Canon Inc.
Mark Little, Red Hat
Dr. Ingo Lueck, Technische Universitaet Dortmund
Jonathan Marsh, WSO2
Carl Mattocks
Antoine Mensch
Jaime Meritt, Progress Software
Vipul Modi, Microsoft Corporation
Anthony Nadalin, IBM
Tadahiro Nakamura, Canon Inc.
Masahiro Nishio, Canon Inc.
Toby Nixon, Microsoft Corporation
Shin Ohtake, Fuji Xerox Co., Ltd.
Venkat Reddy, CA
Alain Regnier, Ricoh Company, Ltd.
Hitoshi Sekine, Ricoh Company, Ltd.
Hiroshi Tamura, Ricoh Company, Ltd.
Minoru Torii, Canon Inc.
Asir S Vedamuthu, Microsoft Corporation
David Whitehead, Lexmark International Inc.
Don Wright, Lexmark International Inc.
Prasad Yendluri, Software AG, Inc.
Elmar Zeeb, University of Rostock
Gottfried Zimmermann

Co-developers of the initial contributions:

This document is based on initial contributions to the OASIS WS-DD Technical Committee by the follow co-devlopers:

Shannon Chan, Microsoft Corporation
Dan Conti, Microsoft Corporation
Chris Kaler, Microsoft Corporation
Thomas Kuehnel, Microsoft Corporation
Alain Regnier, Ricoh Company Limited
Bryan Roe, Intel Corporation

Dale Sather, Microsoft Corporation
Jeffrey Schlimmer, Microsoft Corporation (Editor)
Hitoshi Sekine, Ricoh Company Limited
Jorgen Thelin, Microsoft Corporation (Editor)
Doug Walter, Microsoft Corporation
Jack Weast, Intel Corporation
Dave Whitehead, Lexmark International Inc.
Don Wright, Lexmark International Inc.
Yevgeniy Yarmosh, Intel Corporation

B. Constants

The following constants are used throughout this profile. The values listed below supersede other values defined in other specifications listed below.

Constant	Value	Specification
APP_MAX_DELAY	2,500 milliseconds	[WS-Discovery]
DISCOVERY_PORT	3702	[WS-Discovery]
MATCH_TIMEOUT	10 seconds	[WS-Discovery]
MAX_ENVELOPE_SIZE	32,767 octets	This profile
MAX_UDP_ENVELOPE_SIZE	4,096 octets	This profile
MAX_FIELD_SIZE	256 Unicode characters	This profile
MAX_URI_SIZE	2,048 octets	This profile
MULTICAST_UDP_REPEAT	1	[SOAP-over-UDP]
UDP_MAX_DELAY	250 milliseconds	[SOAP-over-UDP]
UDP_MIN_DELAY	50 milliseconds	[SOAP-over-UDP]
UDP_UPPER_DELAY	450 milliseconds	[SOAP-over-UDP]
UNICAST_UDP_REPEAT	1	[SOAP-over-UDP]

C. Declaring Discovery Types in WSDL

Solutions built on DPWS often define portTypes implemented by Hosted Services, and a discovery-layer portType implemented by the Host Service so the presence of these functional services can be determined at the discovery layer. The binding between a service-layer type and its discovery-layer type can be defined purely in descriptive text, but this appendix provides an optional mechanism to declare a discovery-layer type inside WSDL that can be consumed and understood by tools.

This appendix defines an @dpws:DiscoveryType attribute to annotate the WSDL 1.1 portType [WSDL 1.1] for the service-layer type. The normative outline for @dpws:DiscoveryType is:

```
<wsdl:definitions …>
    [<wsdl:portType [dpws:DiscoveryType="xs:QName"]? >
        …
    </wsdl:portType>]*
</wsdl:definitions>
```

The following describes additional, normative constraints to the outline listed above:

/wsdl:definitions/wsdl:portType/@dpws:DiscoveryType

The name of the portType to be advertised by the Host Service to indicate that this device supports the annotated Hosted Service portType.

If omitted, no implied value

This mechanism is only suitable in cases where a functional service type is bound to a single discovery-layer type, and authors of more complex type topologies may express the relationship between service and discovery types through normative text or through other means.

Example usage follows. PrintDeviceType is the discovery-layer type for PrintPortType.

```
<wsdl:definitions
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01"
    targetNamespace="http://printer.example.com/imaging"
    xmlns:tns="http://printer.example.com/imaging">

    <wsdl:portType name="PrintPortType"
        dpws:DiscoveryType="tns:PrintDeviceType">

        <!-- Contents omitted for brevity -->

    </wsdl:portType>

    <!-- Define PrintDeviceType to be empty -->
    <wsdl:portType name="PrintDeviceType" />

</wsdl:definitions>
```

D. Revision History

[optional; should not be included in OASIS Standards]

Revision	Date	Editor	Changes Made
wd-01	09/16/2008	Dan Driscoll	Converted input specification to OASIS template.
wd-02	10/08/2008	Dan Driscoll	Resolved the following issues: • 001: Clarify R4032 and R4036 w.r.t. other multicast bindings • 002: Define matching for empty Action filter • 003: Fault Action should use lowercase 'f' • 004: Faulting to non-anonymous endpoints • 005: SOAP Binding should apply to clients • 013: Restrict encoding of SOAP messages to UTF-8 • 016: Edit R0042 • 028: Review constants • 045: EndpointReference subelement • 061: Assign an OASIS namespace for the specifications
wd-02	10/14/2008	Dan Driscoll	• Changed document format from doc to docx • Fixed "authoritative reference"
wd-02	10/14/2008	Dan Driscoll	• Changed version number to 1.1 • Removed "related work" section
wd-02	10/14/2008	Dan Driscoll	• Changed copyrights from 2007 to 2008
wd-03	12/12/2008	Dan Driscoll	• Changed draft from cd-01 to wd-03 • Updated dates to 2008/12/12 • Updated namespace to 2009/01 • Issue 098: Update namespace • Editorial: Changed 'wsdp' prefix to 'dpws'
wd-03	12/12/2008	Dan Driscoll Antoine Mensch	• 011: Fix SERVICE terminology • 015: Remove R0007 • 024: Fix Directed Discovery

			• 029: Fix SERVICE/DEVICE for WS-Policy • 038: Contents of Host EPR • 039: Recursive hosting • 055: WS-Addressing 1.0 • 070: HTTP content negotiation for PresentationUrl • 071: Update to WS-Policy 1.5 • 073: Clarify “stable” identifier • 074: Clarify R0036/R0037 • 075: Clarify “Target Service” • 077: Remove R3010 as redundant • 080: Secure all WS-A headers • 084: Faulting behavior on Subscribe • 085: Get/GetMetadata • 092: Split R3019 • 093: Remove R3012 • 094: Clean up expiration type/value switching • 095: Clarify expiration value switching • 109: Update references
wd-03	1/2/2009	Dan Driscoll	• 032: Describe security composability • 051: Generalize security • 112: Remove WS-Security reference • 113: Cleanup Network Model • 114: Remove security negotiation • 115: Replace R4070 with switches on HTTPS ID/xAddrs • 138: Create introduction and concrete description of security profile • 139: Remove protocol negotiation • 140: Clean up HTTP Authentication
wd-03	1/21/2009	Antoine Mensch	• Issue 012 • Issue 040 • Issue 046 • Issue 117 • Issue 127 • Issue 128 • Issue 135 • Issue 143
cd-02	1/21/2009	Dan Driscoll	• Changed draft from wd-03 to cd-02

Candidate			• Updated date, copyrights • Updated WS-Discovery and SOAP-over-UDP references to CD-02 • 072: Fix HOSTEDSERVICE • 083: Fix R0031 and wsa:ReplyTo • 130: Make FilterActionNotSupported recommended, not mandatory • 132: Define relative PresentationUrl • 134: Make Types/Scopes mandatory in directed ProbeMatches • 137: Add Appendix C • More security edits (see Section 7)
cd-02 Candidate	1/26/2009	Dan Driscoll	• Fixed WS-DD committee site links • Added TC participants to Appendix A; remove company names to meet OASIS rules • Removed “Last Approved Version”
cd-02	1/27/2009	Dan Driscoll	• Updated to reflect CD-02 status

1265