



Service Component Architecture Web Service Binding Specification Version 1.1

Committee Draft 02 ~~Revision 1 plus issue 61 updates,~~
~~rev3~~

~~2nd~~20th April~~June~~, 2009

Specification URIs:

This Version:

<http://docs.oasis-open.org/opencsa/sca-bindings/sca-binding-ws-1.1-spec-cd02.html>
<http://docs.oasis-open.org/opencsa/sca-bindings/sca-binding-ws-1.1-spec-cd02.doc>
<http://docs.oasis-open.org/opencsa/sca-bindings/sca-binding-ws-1.1-spec-cd02.pdf>
(Authoritative)

Previous Version:

<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-spec-cd01.html>
<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-spec-cd01.doc>
<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-spec-cd01.pdf> (Authoritative)

Latest Version:

<http://docs.oasis-open.org/opencsa/sca-bindings/sca-binding-ws-1.1-spec.html>
<http://docs.oasis-open.org/opencsa/sca-bindings/sca-binding-ws-1.1-spec.doc>
<http://docs.oasis-open.org/opencsa/sca-bindings/sca-binding-ws-1.1-spec.pdf> (Authoritative)

Latest Approved Version:

Technical Committee:

OASIS Service Component Architecture / Bindings (SCA-Bindings) TC

Chair(s):

Simon Holdsworth, IBM

Editor(s):

Simon Holdsworth, IBM
~~Khanderao Kand, Oracle~~
Anish Karmarkar, Oracle
~~Sanjay Patil, SAP~~
Piotr Przybylski, IBM

Related work:

This specification replaces or supercedes:

- Service Component Architecture Web Service Binding Specification Version 1.00, March 21 2007

This specification is related to:

- Service Component Architecture Assembly Model Specification Version 1.1
- Service Component Architecture Policy Framework Specification Version 1.1

Declared XML Namespace(s):

<http://docs.oasis-open.org/ns/opencsa/sca/200712200903>

Abstract:

The SCA Web Service binding specified in this document applies to the services and references of an SCA composites. It defines the manner in which a service can be made available as a web service, and in which a reference can invoke a web service.

This binding is a WSDL-based binding; that means it either references an existing WSDL binding or allows one to specify enough information to generate one. When an existing WSDL binding is not referenced, rules defined in this document allow one to generate a WSDL binding.

Status:

This document was last revised or approved by the OASIS Service Component Architecture / Bindings (SCA-Bindings) 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/sca-bindings/>.

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/sca-bindings/ipr.php>).

The non-normative errata page for this specification is located at <http://www.oasis-open.org/committees/sca-bindings/>.

Notices

Copyright © OASIS® 2005⁶, 2009⁸. All Rights Reserved.

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 names "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	6
1.1	Terminology	6
1.2	Normative References	7
1.3	Non-Normative References	7
1.4	Naming Conventions	7
2	Web Service Binding Schema	9
2.1	Compatibility of SCA Service Interfaces and WSDL portTypes	12
2.2	Endpoint URI resolution	12
2.3	Interface mapping	13
2.4	Production of WSDL description for an SCA service	14
2.5	Additional binding configuration data	14
2.6	Web Service Binding and SOAP Intermediaries	14
2.7	Support for WSDL extensibility	14
2.8	Intents listed in the bindingType	15
2.9	Intents and binding configuration	15
3	Web Service Binding Examples	16
3.1	Example Using WSDL documents	16
3.2	Examples Without a WSDL Document	17
4	Transport Binding	19
4.1	Intents	19
4.2	Default Transport Binding Rules	19
4.2.1	WS-I Basic Profile Alignment	19
4.2.2	Default Transport Binding Rules	20
5	Conformance	21
5.1	SCA WS Binding XML Document	21
5.2	SCA Runtime	21
A	Web Services XML Binding Schema: sca-binding-webservice.xsd	22
B	Conformance Items	23
C	Appendix - WSDL Generation	27
D	Acknowledgements	28
E	Non-Normative Text	29
F	Revision History	30
1	Introduction	5
1.1	Terminology	5
1.2	Normative References	6
1.3	Non-Normative References	6
2	Web Service Binding Schema	7
2.1	Endpoint URI resolution	8
2.2	Interface mapping	9
2.3	Production of WSDL description for an SCA service	9
2.4	Additional binding configuration data	9
2.5	Web Service Binding and SOAP Intermediaries	9
2.6	Support for WSDL extensibility	9

2.7 Intents listed in the bindingType	10
2.8 Intents and binding configuration	10
3 Web Service Binding Examples	11
3.1 Example Using WSDL documents	11
3.2 Examples Without a WSDL Document	12
3.3 Example PolicySet Providing The Conversation Intent	13
4 Transport Binding	14
4.1 Intents	14
4.2 Default Transport Binding Rules	14
4.2.1 WS-I Basic Profile Alignment	14
4.2.2 Default Transport Binding Rules	14
5 Conformance	16
A. Web Services Binding Schema	17
B. Appendix – WSDL Generation	18
C. Acknowledgements	19
D. Non-Normative Text	20
E. Revision History	21

1 Introduction

The SCA Web Service binding specified in this document applies to the services and references of composites and components [SCA-Assembly]. It defines the manner in which a service can be made available as a web service, and in which a reference can invoke a web service.

This binding is a WSDL-based binding; that means it either references an existing WSDL binding or can be configured to specify enough information to generate one. When an existing WSDL binding is not referenced, rules defined in this document allow one to generate a WSDL binding. [This specification only defines a binding using WSDL 1.1.](#)

The Web Service binding can point to an existing WSDL [\[WSDL11\]\[WSDL11\]](#) document, separately authored, that specifies the details of the WSDL binding to be used to provide or invoke the web service. In this case the SCA web services binding allows anything that is valid in a WSDL binding, including rpc-encoded style and binding extensions. It is the responsibility of the SCA system provider to ensure support for all options specified in the WSDL binding. Interoperation of such services is not guaranteed.

The SCA Web Service binding also provides attributes that can be used to provide the details of a WSDL SOAP binding. This allows a WSDL document to be synthesized in the case that one does not already exist. In this case only WS-I compliant mapping is supported.

The SCA Web Service binding can be further customized through the use of SCA Policy Sets. For example, a requirement to conform to a WS-I profile [\[WSI-Profiles\]](#) could be represented with a policy set.

1.1 Terminology

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 [\[RFC2119\]\[RFC2119\]](#).

This specification uses predefined namespace prefixes throughout; they are given in the following list. Note that the choice of any namespace prefix is arbitrary and not semantically significant.

Table 1-1 Prefixes and Namespaces used in this specification

Prefix	Namespace	Notes
xs	"http://www.w3.org/2001/XMLSchema"	Defined by XML Schema 1.0 specification
wsa	"http://www.w3.org/2005/08/addressing"	Defined by WS-Addressing 1.0
wsp	"http://www.w3.org/ns/ws-policy"	Defined by WS-Policy 1.5
wsrmp	"http://docs.oasis-open.org/ws-rx/wsrmp/200702"	Defined by WS-ReliableMessaging Policy 1.2
soap11	"http://schemas.xmlsoap.org/soap/envelope/"	Defined by SOAP 1.1
soap12	"http://www.w3.org/2005/08/addressing"	Defined by SOAP 1.2
wsdli	"http://www.w3.org/ns/wsdli-instance"	Defined by WSDL 2.0

wsoap11	"http://schemas.xmlsoap.org/wsd/soap/"	Defined by WSDL 1.1 [WSDL11]
wsoap12	"http://schemas.xmlsoap.org/wsd/soap12/"	Defined by [W11-SOAP12]
sca	"http://docs.oasis-open.org/ns/opencsa/sca/200712200903"	Defined by the SCA specifications

31

32 1.2 Normative References

- 33 [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
34 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 35 [SCA-Assembly] [http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-](http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-spec.html)
36 [spec.html](http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-spec.pdf)[http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-](http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-spec.pdf)
37 [spec.pdf](http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-spec.pdf)
- 38 [SCA-Policy] <http://docs.oasis-open.org/opencsa/sca-policy/sca-policy-1.1-spec.pdf>
- 39 [SCA-JCAA] [http://docs.oasis-open.org/opencsa/sca-j/sca-javacaa-1.1-](http://docs.oasis-open.org/opencsa/sca-j/sca-javacaa-1.1-spec.html)
40 [spec.html](http://docs.oasis-open.org/opencsa/sca-j/sca-javacaa-1.1-spec.pdf)<http://docs.oasis-open.org/opencsa/sca-j/sca-javacaa-1.1-spec.pdf>
- 41 [WSDL11] E. Christensen et al, *Web Service Description Language (WSDL) 1.1*,
42 <http://www.w3.org/TR/2001/NOTE-wsd-20010315>, W3C Note, March 15 2001.
- 43 [WSDL20] Chinnici et al, *Web Service Description Language (WSDL) Version 2.0 Part 1:*
44 *Core Language*, <http://www.w3.org/TR/2007/REC-wsd20-20070626/>, W3C
45 Recommendation, June 26 2007.
- 46 [WSI-Profiles] <http://www.ws-i.org/Profiles/BasicProfile-1.1.html>
47 <http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>
48 <http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0.html>
49 <http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html>
- 50 [JAX-WS] <http://jcp.org/en/jsr/detail?id=224>
- 51 [SOAP11] <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- 52 [SOAP] <http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>
53 <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- 54 [SOAP12Adjuncts] SOAP Version 1.2 Part 2: Adjuncts (Second Edition)
55 <http://www.w3.org/TR/soap12-part2/>
- 56 [WS-Addr] <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/>
- 57 [W11-SOAP12] <http://www.w3.org/Submission/wsd11soap12/>
- 58

Field Code Changed

Field Code Changed

59 1.3 Non-Normative References

- 60 [WSI-AP] <http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>
- 61 [MTOM] <http://www.w3.org/TR/2005/REC-soap12-mtom-20050125/>
- 62 [WS-RM] <http://docs.oasis-open.org/ws-rx/wsrn/200702/wsrn-1.2-spec-cd-01.html>

Formatted: Heading 2,H2

63 1.4 Naming Conventions

64 This specification follows some naming conventions for artifacts defined by the
65 specification. In addition to the conventions defined by section 1.3 of the Assembly
66 [SCA-Assembly] specification, this specification adds three additional conventions:

- 67 1. Where the names of elements and attributes consist partially or wholly of
68 acronyms, the letters of the acronyms use the same case. When the acronym
69 appears at the start of the name of an element or an attribute, or after a period,
70 it is in lower case. If it appears elsewhere in the name of an element or an

Formatted: Numbered + Level: 1 +
Numbering Style: 1, 2, 3, ... + Start at: 1 +
Alignment: Left + Aligned at: 0.5" + Indent at:
0.75"

- 71 | attribute, it is in upper case. For example, an attribute might be named "uri" or
72 | "jndiURL".
- 73 | 2. Where the names of types consist partially or wholly of acronyms, the letters of
74 | the acronyms are in all upper case. For example, an XML Schema type might be
75 | named "JCABinding" or "MessageID".
- 76 | 3. Values, including local parts of QName values, follow the rules for names of
77 | elements and attributes as stated above, with the exception that the letters of
78 | acronyms are in all upper case. For example, a value might be "JMSDefault" or
79 | "namespaceURI".

2 Web Service Binding Schema

The Web Service binding element is defined by the following pseudo-schema.

```
<binding.ws name="xs:NCName"?
  requires="list of xs:QName"?
  policySets="list of xs:QName"?
  uri="xs:anyURI"?
  wsdlElement="xs:anyURI"?
  wsdl:wsdlLocation="list of xs:anyURI pairs"?
  ...>
  <wireFormat/>?
  <operationSelector/>?
  <endpointReference>...</endpointReference>*
  ...
</binding.ws>
```

- **/binding.ws/@name** - as defined in the SCA Assembly Specification [SCA-Assembly].
- **/binding.ws/@requires** - as defined in the SCA Assembly Specification [SCA-Assembly].
- **/binding.ws/@policySets** - as defined in the SCA Assembly Specification [SCA-Assembly].
- **/binding.ws/@uri** - the resolution algorithm of Section 2.24 below describes how this attribute is interpreted. For an SCA reference, the @uri attribute MUST be an absolute value. [BWS20001]
- **/binding.ws/@wsdlElement** - when present this attribute specifies the URI of a WSDL element. The value of the @wsdlElement attribute MUST identify an element in an existing WSDL 1.1 document. [BWS20002] The URI can have the following forms:

• **_Service:**

```
<WSDL-namespace-URI>#wsdl.service(<service-name>)
```

If the binding is for an SCA service, the wsdlElement attribute MUST NOT specify the wsdl.service form of URI. [BWS20003]

If the binding is for an SCA reference, the set of available ports for the reference consists of the ports in the WSDL service that have portTypes which are compatible supersets of the SCA reference as defined in the SCA Assembly Model specification [SCA-Assembly] and satisfy all the policy constraints of the binding.

If the wsdl.service form of wsdlElement is used on an SCA reference binding, the set of available ports for the reference MUST contain at least one port. [BWS20004] The set of available ports represents a single SCA reference binding with respect to the multiplicity of that SCA reference. If the wsdl.service form of wsdlElement is used on an SCA reference binding, the SCA runtime MUST raise an error if there are no available ports that it supports. [BWS20005] When an invocation is made using an SCA reference binding with the wsdl.service form of wsdlElement, the SCA runtime MUST use exactly one port from the set of available ports for the reference (with port selection on a per-invocation basis permitted). [BWS20006]

Formatted: Highlight

Formatted: Highlight

Formatted: Indent: Left: 0.5", Bulleted + Level: 1 + Aligned at: 0.25" + Tab after: 0.5" + Indent at: 0.5", Tab stops: 0.75", List tab + Not at 0.5"

Formatted: Highlight

Formatted: Highlight

Formatted: Indent: Left: 0.75", No bullets or numbering, Tab stops: Not at 0.75"

Formatted: Highlight

Formatted: Highlight

Formatted: Font color: Red

126 In this case, the SCA runtime MUST make all the ports in the WSDL Service that
127 have equivalent portTypes with the SCA service or reference available to the SCA
128 service or reference.

129 ● Port (WSDL 1.1):

130 <WSDL-namespace-URI>#wsdl.port(<service-name>/<port-name>)

131 If the binding is for an SCA service, the portType associated with the specified
132 WSDL port MUST be compatible with the SCA service interface as defined in
133 section 2.1, and the port MUST satisfy all the policy constraints of the binding.
134 [BWS20007] The SCA runtime MUST expose an endpoint for the specified WSDL
135 port, or raise an error if it does not support the WSDL port. [BWS20008]

136 If the binding is for an SCA reference, the portType associated with the
137 specified WSDL port MUST be a compatible superset of the SCA reference
138 interface as defined in the SCA Assembly Model specification [SCA-Assembly],
139 and the port MUST satisfy all the policy constraints of the binding. [BWS20009]
140 The SCA runtime MUST use the specified WSDL port for invocations made using
141 the SCA reference, or raise an error if it does not support the WSDL port.
142 [BWS20010]

143 In this case, the port in the WSDL 1.1 Service identified by the <binding.ws>
144 element MUST implement a portType that is equivalent to the one specified for
145 the SCA service or reference. The identified port MUST be made available to the
146 SCA service or reference by the SCA runtime.

147 ● Endpoint (WSDL 2.0):

148 <WSDL-namespace-URI>#wsdl.endpoint(<service-name>/<endpoint-name>)

149 In this case, the endpoint in the WSDL 2.0 Service identified by the
150 <binding.ws> element MUST have an equivalent portType with the SCA service
151 or reference. The identified endpoint MUST be made available to the SCA service
152 or reference by the SCA runtime.

153

154 ● Binding:

155 <WSDL-namespace-URI>#wsdl.binding(<binding-name>)

156 If the binding is for an SCA service, the portType associated with the specified
157 WSDL binding MUST be compatible with the SCA service interface as defined in
158 section 2.1, and the WSDL binding MUST satisfy all the policy constraints of the
159 binding. [BWS20011] The SCA runtime MUST expose an endpoint for the
160 specified WSDL binding, or raise an error if it does not support the WSDL binding.
161 [BWS20012]

162 If the binding is for an SCA reference, the portType associated with the specified
163 WSDL binding MUST be a compatible superset of the SCA reference interface as
164 defined in the SCA Assembly Model specification [SCA-Assembly], and the WSDL
165 binding MUST satisfy all the policy constraints of the binding. [BWS20013] The
166 SCA runtime MUST use the specified WSDL binding for invocations made using
167 the SCA reference, or raise an error if it does not support the WSDL binding.
168 [BWS20014] In this case, the WSDL binding identified by the <binding.ws>
169 element MUST implement a portType that is equivalent to the one specified for
170 the SCA service or reference. The SCA runtime MUST make the service or
171 reference available via the specified WSDL binding.

Formatted: Bulleted + Level: 1 + Aligned at: 0.5" + Tab after: 0.75" + Indent at: 0.75"

Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

Formatted: Indent: Left: 0.75", No bullets or numbering, Tab stops: Not at 0.75"

Formatted: Highlight

Formatted: Bulleted + Level: 1 + Aligned at: 0.5" + Tab after: 0.75" + Indent at: 0.75"

Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

172 ~~When the *wsdl.binding* form of *wsdlElement* is used in this case, the endpoint~~
173 ~~address URI for an SCA reference MUST be specified by either the *@uri* attribute~~
174 ~~on the binding or a WS-Addressing *EndpointReference* element, except where the~~
175 ~~SCA Assembly Model specification [SCA-Assembly] states that the *@uri* attribute~~
176 ~~can be omitted. [BWS20015]. The endpoint address URI for an SCA service or~~
177 ~~the callback element of an SCA reference is determined as specified in section~~
178 ~~2.1. For the *callback* element of an SCA service, the binding MUST NOT specify~~
179 ~~an endpoint address URI or a WS-Addressing *EndpointReference*.~~

Formatted: Highlight

180 • **/binding.ws/@wsdl:wsdlLocation** – when present this attribute specifies the
181 location(s) of the WSDL document(s) associated with specific namespace(s).

182 The *@wsdl:wsdlLocation* attribute MAY be specified by the binding in the
183 event that the *<WSDL-namespace-URI>* in the 'endpoint' attribute is not
184 dereferencable, or when the intended WSDL document is to be found at a different
185 location than the one pointed to by the *<WSDL-namespace-URI>*.
186 The *@wsdl:wsdlLocation* attribute MAY be specified by the binding in the event that
187 the *<WSDL-namespace-URI>* in the 'endpoint' attribute is not dereferencable, or
188 when the intended WSDL document is to be found at a different location than the
189 one pointed to by the *<WSDL-namespace-URI>*. [BWS20001][BWS20016]

Formatted: Indent: Left: 0.5", No bullets or numbering

190 If the *@wsdl:wsdlLocation* attribute is used the *@wsdlElement* attribute
191 MUST also be specified. [BWS20017].
192 If the *@wsdl:wsdlLocation* attribute is used the *@wsdlElement* attribute MUST also
193 be specified. [BWS20009].

Formatted: Font color: Red

194 The semantics of this attribute are specified in Section 7.1 of WSDL 2.0
195 [WSDL20][WSDL20]. The value of the *@wsdl:wsdlLocation* attribute MUST identify
196 an existing WSDL 1.1 document. [BWS20018]

Formatted: Font color: Red

Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

197 • **/binding.ws/wireFormat** – as defined in the SCA Assembly Specification [SCA-
198 Assembly]. This specification does not define any new wireFormat elements.

199 • **/binding.ws/operationSelector** – as defined in the SCA Assembly Specification [SCA-
200 Assembly]. This specification does not define any new operationSelector elements.

201 • **/binding.ws/endpointReference** – when present this element provides the WS-
202 Addressing [WS-Addr] EndpointReference that specifies the endpoint for the service
203 or reference.

204 When the *endpointReference* element is present along with the *@wsdlElement*
205 attribute on the parent element, the *@wsdlElement* attribute value MUST be of the
206 'Binding' form *<WSDL-namespace-URI> #wsdl.binding(<binding-name>)*
207 [BWS20002].

208 • **/binding.ws/@{any}** – this is an extensibility mechanism to allow extensibility via
209 attributes.

210 • **/binding.ws/any** – this is an extensibility mechanism to allow extensibility via
211 elements.

212 A *binding.ws* element MUST NOT contain more than one of any of the following:

Formatted: Highlight

- 213 • the *@uri* attribute
- 214 • the *@wsdlElement* attribute referring to a WSDL port or to a WSDL service
- 215 • the *endpointReference* element

Formatted: Bulleted + Level: 1 + Aligned at: 0.5" + Indent at: 0.75"

216 [BWS20019]

217 The endpoint address URI for an SCA service or the callback element of an SCA
218 reference is determined as specified in section 2.2. For the *callback* element of an SCA

Formatted: Highlight

219 service, the binding MUST NOT specify an endpoint address URI or a WS-Addressing
 220 EndpointReference. [BWS20020],

221 The SCA runtime MUST support all the attributes of the <binding.ws> element, namely
 222 @name, @uri, @requires, @policySets @wsdlElement, and @wsdl:wsdlLocation.
 223 [BWS20021],

224 The SCA runtime SHOULD support the element <endpointReference>. [BWS20022], If an
 225 SCA runtime does not support the element <endpointReference>, then it MUST reject
 226 an SCA WS Binding XML document (as defined in Section 5.1) that contains the element.
 227 [BWS20023],

228 The <binding.ws> element MUST conform to the XML schema defined in sca-binding-
 229 webservice.xsd. [BWS20024],

- Formatted: Font color: Red
- Formatted: Font: Verdana, Font color: Auto, Highlight
- Formatted: Font: Verdana, Font color: Auto
- Formatted: Font: Verdana, Font color: Auto, Highlight
- Formatted: Font: Verdana, Font color: Auto
- Formatted: Font: Verdana, Font color: Auto, Highlight
- Formatted: Font: Verdana, Font color: Auto
- Formatted: Font: Verdana, Font color: Auto, Highlight
- Formatted: Font: Verdana, Font color: Auto

230 **2.1 Compatibility of SCA Service Interfaces and WSDL portTypes**

231 A WSDL portType is compatible with an SCA service interface if and only if all of the
 232 following conditions are satisfied:

- 233 1. The SCA service interface is remotable.
- 234 2. The operations on the portType are the same as the operations on the SCA
 235 service interface, with the same operation name, same input types (taking order
 236 as significant), same output types (taking order as significant), and same
 237 fault/exception types. If the SCA service interface is not a WSDL portType, it is
 238 mapped to a WSDL portType for the purposes of this comparison. The mapping
 239 is defined in the relevant SCA specification for the interface type. If the interface
 240 cannot be mapped to WSDL, the SCA service interface is not compatible with the
 241 WSDL portType.
- 242 3. WSDL 1.1 message parts can point to an XML Schema element declaration or an
 243 XML Schema type. When determining compatibility between two WSDL
 244 operations, a message part that points to an XML Schema element is considered
 245 to be incompatible with a message part that points to an XML Schema type.
- 246 4. If either the portType or the SCA service interface declares an SCA callback
 247 interface, then both the portType and the SCA service interface declare callback
 248 interfaces and these callback interfaces are compatible according to points 1
 249 through 3 above.

250 **2.2 Endpoint URI resolution**

251 This specification does not mandate any particular way to determine the URI for a web
 252 services binding on an SCA service. An absolute URI can be indicated by the @uri
 253 attribute, by the URI in a wsa:Address element within an endpointReference element, or
 254 by the URI indicated in a WSDL Port via a @wsdlElement attribute. Implementations
 255 can use the specified URI as the service endpoint URI or they can use a different URI
 256 which might include portions of the specified URI. For example, the service endpoint
 257 URI might be produced by modifying any or all of the host name, the port number, and
 258 a portion of the path.

259 Note that if no absolute URI is indicated by any of these elements, implementations can
 260 use the structural URI for the binding as a portion of the URI for the eventual deployed
 261 endpoint. In addition, the @uri attribute value could be relative; implementations are
 262 encouraged to combine this value with the structural URI for the service in determining
 263 a deployed URI.

264 The target address for a reference binding is defined as one of the following:

Formatted: Outline numbered + Level: 2 +
 Numbering Style: 1, 2, 3, ... + Start at: 1 +
 Alignment: Left + Aligned at: 0" + Indent at:
 0.4"

- 265 A. The value of the @uri attribute
- 266 B. The value of the wsa:Address element of the endpointReference element
- 267 C. The value of the address element of the WSDL port referenced by the
- 268 @wsdlElement attribute
- 269 D. The value of the address element of one of the set of available WSDL ports as
- 270 specified under the definition of the @wsdlElement attribute when it references a
- 271 WSDL service element

Formatted: Numbered + Level: 1 +
Numbering Style: A, B, C, ... + Start at: 1 +
Alignment: Left + Aligned at: 0.5" + Indent at:
0.75"

- 272 If there is no target address for a reference binding, the SCA runtime MUST raise an
- 273 error. [BWS20025]
- 274 For a reference binding, the SCA runtime MUST use the target address. [BWS20026]

Formatted: Highlight

Formatted: Highlight

275 The rules for resolving the URI at which an SCA service is hosted, or SCA reference

276 targets, when used with binding.ws (in precedence order) are:

- 277 1. The URIs in the endpoint(s) of the referenced WSDL
- 278 or
- 279 The URI specified by the wsa:Address element of the endpointReference,
- 280 2. The explicitly stated URI in the @uri attribute of the binding.ws element, which
- 281 can be relative,
- 282 3. The structural URI as defined by the Assembly specification

283 An SCA runtime MUST follow rules listed above in determining the URI at which an SCA

284 service is hosted or an SCA reference is targeted.

285 The URI in the WSDL endpoint or in the wsa:Address of an EPR MAY be a relative URI, in

286 which case it is relative to the URI defined in (2) or (3). The wsa:Address element MAY

287 be the empty relative URI, in which case it uses the URI defined in (2) or (3) directly.

288 This enables the EPR writer to specify reference parameters, metadata and other EPR

289 contents while letting the deployer choose the URI.

290 To reference a WSDL document and also specify an EPR, the @wsdlElement attribute

291 MUST refer to a binding element in the WSDL.

292 2-12.3 Interface mapping

293 When binding.ws is used on a service or reference with an interface that is not defined

294 by interface.wsdl, the SCA runtime MUST derive a WSDL portType for the service or

295 reference from the interface using the rules defined for that SCA interface type.

296 [BWS20027] When binding.ws is used on a service or reference with an interface that is

297 not defined by interface.wsdl, the SCA runtime MUST derive a WSDL portType for the

298 service or reference from the interface using the rules defined for that SCA interface

299 type. [BWS20010].

Formatted: Font color: Red

300 An SCA runtime MUST raise an error if the interface on a service or reference element

301 with a binding.ws element does not map to a WSDL portType.

302 An SCA runtime MUST raise an error if the interface on a service or reference element

303 with a binding.ws element does not map to a WSDL portType [BWS20028][BWS20003].

Formatted: Font color: Red

304 For example, for *interface.java*, the mapping to a WSDL portType is as defined in the

305 SCA Java Common Annotations and API Specification [SCA-JCAA].

306 *binding.ws* implementations can use appropriate standards, for example WS-I AP 1.0

307 [WSI-AP] or MTOM [MTOM], to map interface parameters to binary attachments

308 transparently to the target component.

309

310 2.22.4 Production of WSDL description for an SCA service

311 ~~Any service hosted by an SCA runtime with one or more web service bindings with HTTP~~
312 ~~endpoints SHOULD return a WSDL description of the service in response to an HTTP GET~~
313 ~~request with the "?wsdl" suffix to that HTTP endpoint. Any service hosted by an SCA~~
314 ~~runtime with one or more web service bindings with HTTP endpoints SHOULD return a~~
315 ~~WSDL description of the service in response to an HTTP GET request with the "?wsdl"~~
316 ~~suffix to that HTTP endpoint. [BWS20004][BWS20029]~~

317 ~~If none of the web service bindings for an SCA service have HTTP endpoints, then the~~
318 ~~SCA runtime SHOULD provide some other means of obtaining the WSDL description of~~
319 ~~the service.~~

320 ~~If none of the web service bindings for an SCA service have HTTP endpoints, then some~~
321 ~~other means of obtaining the WSDL description of the service SHOULD be provided by~~
322 ~~the SCA runtime [BWS20030][BWS20005]. This can include out of band mechanisms,~~
323 ~~for example publication to a UDDI registry.~~

324 Refer to section 4 for a detailed definition of the rules that are used for generating the
325 WSDL description of an SCA service with one or more web service bindings.

326

327 2.32.5 Additional binding configuration data

328 ~~SCA runtime implementations MAY provide additional metadata that is associated with a~~
329 ~~web service binding. SCA runtime implementations MAY provide additional metadata that~~
330 ~~is associated with a web service binding. [BWS20031][BWS20006].~~

331 This can be used for example to enable JAX-WS [JAX-WS] handlers to be executed as
332 part of the target component dispatch. The specification of such metadata is SCA
333 runtime-specific and is outside of the scope of this document.

334

335 2.42.6 Web Service Binding and SOAP Intermediaries

336 The Web Service binding does not provide any direct or explicit support for SOAP
337 intermediaries [SOAP].

338

339 2.52.7 Support for WSDL extensibility

340 When a binding.ws element uses the @wsdlElement attribute, the details of the binding
341 are specified by the WSDL element referenced by the value of the attribute. Per the
342 WSDL specification, WSDL allows for extensibility via elements as well as attributes, and
343 it specifies rules for processing such elements. This specification does not constrain the
344 use of such extensibility in WSDL and relies on the rules specified in the WSDL
345 specification for processing such extended elements.

346 ~~An SCA runtime MUST support the WSDL extensions defined in the namespace~~
347 ~~associated with the prefix "sca" (as defined in section 1.1).~~

348 ~~An SCA runtime MUST support the WSDL extensions defined in the namespace~~
349 ~~associated with the prefix "sca" [BWS20007] (as defined in section 1.1). [BWS20032]~~

Formatted: Font color: Red

Formatted: Font color: Red

Formatted: Font color: Red

Formatted: Font color: Red

Formatted: Highlight

350 The SCA runtime MUST support the WSDL 1.1 binding extension for SOAP 1.1 over HTTP
351 [WSDL11], as identified by the WSDL element wsoap11:binding that has the @transport
352 attribute with a value of "http://schemas.xmlsoap.org/soap/http". [BWS20033]

Formatted: Highlight

353 The SCA runtime SHOULD support the WSDL 1.1 binding extension for SOAP 1.2 over
354 HTTP [WSDL11-SOAP12], as identified by the WSDL element wsoap12:binding that has
355 the @transport attribute with a value of "http://schemas.xmlsoap.org/soap/http".
356 [BWS20034]

Formatted: Highlight

357 Because a WSDL document might contain extension elements that cannot be supported
358 by the SCA runtime, when using the @wsdlElement form of binding.ws it is not possible
359 to determine whether the binding is supported by the SCA runtime without parsing the
360 referenced WSDL element and its dependent elements.

361 2.62.8 Intents listed in the bindingType

362 This specification places no requirements on the intents that are listed as either
363 @alwaysProvides or @mayProvides in the bindingType for binding.ws.

364 2.72.9 Intents and binding configuration

365 This binding mandates support for SOAP 1.1 and encourages SOAP 1.2 support. The
366 <bindingType> element associated with this binding MUST include the SOAP 1.1 intent
367 in its @mayProvides or @alwaysProvides attributes. [BWS20035] The <bindingType>
368 element associated with this binding SHOULD include the SOAP 1.2 intent in its
369 @mayProvides attribute. [BWS20036] For more details on the <bindingType> element
370 see [SCA-Policy].

Formatted: Highlight

Formatted: Highlight

371 The SCA runtime MUST raise an error if a web service binding is configured with a policy
372 intent(s) that conflicts with the binding instance's configuration. The SCA runtime MUST
373 raise an error if a web service binding is configured with a policy intent(s) that conflicts
374 with the binding instance's configuration [BWS20037][BWS20008].

Formatted: Font color: Red

375 For example, it is an error to use the SOAP policy intent in combination with a WSDL
376 binding that does not use SOAP.

377 3 Web Service Binding Examples

378 The following snippets show the sca.composite file for the MyValueComposite file
379 containing the service element for the MyValueService and reference element for the
380 StockQuoteService. Both the service and the reference use a Web Service binding.
381

382 3.1 Example Using WSDL documents

383 This example shows a service and reference using the SCA Web Service binding, using
384 existing WSDL documents in both cases. In each case there is a single binding element,
385 whose name defaults to the service/reference name.

386 The service's binding is defined by the WSDL document associated with the given URI.
387 This service conforms to WS-I Basic Profile 1.1.

388 The first reference's first-binding is defined by the specified WSDL service in the WSDL
389 document at the given location. The reference can use any of the WSDL service's
390 ports/endpoints to invoke the target service. The second reference's second-binding is
391 defined by the specified WSDL binding. The specific endpoint URI to be invoked is
392 provided via the @uri attribute.
393

```
394 <?xml version="1.0" encoding="ASCII"?>  
395 <composite xmlns="http://docs.oasis-open.org/ns/opencsa/sca/200712200903"  
396   name="MyValueComposite">  
397   <service name="MyValueService">  
398     <interface.java interface="services.myvalue.MyValueService"/>  
399     <binding.ws wsdlElement="http://www.example.org/MyValueService#  
400  
401 wsdli:wsdlLocation="http://www.example.org/MyValueServiceSOAP" />  
402     ...  
403   </service>  
404  
405   ...  
406  
407   <reference name="StockQuoteReference1">  
408     <interface.java interface="services.stockquote.StockQuoteService"/>  
409     <binding.ws wsdlElement="http://www.example.org/StockQuoteService#  
410       wsdl.service(StockQuoteService) "  
411     wsdli:wsdlLocation="http://www.example.org/StockQuoteService  
412       http://www.example.org/StockQuoteService.wsdl"/>  
413   </reference>  
414  
415   <reference name="StockQuoteReference2">  
416     <interface.java interface="services.stockquote.StockQuoteService"/>  
417     <binding.ws wsdlElement="http://www.example.org/StockQuoteService#  
418       wsdl.binding(StockQuoteBinding) "  
419     wsdli:wsdlLocation="http://www.example.org/StockQuoteService  
420       http://www.example.org/StockQuoteService.wsdl"  
421     uri="http://www.example.org/StockQuoteService5"/>  
422   </reference>  
423 </composite>
```

424 3.2 Examples Without a WSDL Document

425 The next example shows the simplest form of the binding element without WSDL
426 document, assuming all defaults for portType mapping and SOAP binding synthesis. The
427 service and reference each have a single binding element, whose name defaults to the
428 service/reference name.

429 The service is to be made available at a location determined by the deployment of this
430 component. It will have a single port address and SOAP binding, with a simple WS-I
431 BasicProfile 1.1 compliant binding, and using the default options for mapping the Java
432 interface to a WSDL portType.

433 The reference indicates a service to be invoked which has a SOAP binding and portType
434 that matches the default options for binding synthesis and interface mapping. One
435 particular use of this case would be where the reference is to an SCA service with a web
436 service binding which itself uses all the defaults.

```
437  
438 <?xml version="1.0" encoding="ASCII"?>  
439 <composite xmlns="http://docs.oasis-open.org/ns/opencsa/sca/200712200903"  
440 name="MyValueComposite">  
441  
442 <service name="MyValueService">  
443 <interface.java interface="services.myvalue.MyValueService"/>  
444 <binding.ws/>  
445 ...  
446 </service>  
447  
448 ...  
449  
450 <reference name="StockQuoteService">  
451 <interface.java interface="services.stockquote.StockQuoteService"/>  
452 <binding.ws uri="http://www.example.org/StockQuoteService"/>  
453 </reference>  
454 </composite>
```

455 The next example shows the use of the binding element without a WSDL document, with
456 multiple SOAP bindings with non-default values. The SOAP 1.2 binding name defaults to
457 the service name, the SOAP 1.1 binding is given an explicit name. The reference has a
458 web service binding which uses SOAP 1.2, but otherwise uses all the defaults for SOAP
459 binding. The reference binding name defaults to the reference name.
460

```
461  
462 <?xml version="1.0" encoding="ASCII"?>  
463 <composite xmlns="http://docs.oasis-open.org/ns/opencsa/sca/200712200903"  
464 name="MyValueComposite">  
465  
466 <service name="MyValueService">  
467 <interface.java interface="services.myvalue.MyValueService"/>  
468 <binding.ws name="MyValueServiceSOAP11" requires="SOAP.1_1"/>  
469 <binding.ws requires="SOAP.1_2"/>  
470 ...  
471 </service>  
472  
473 ...  
474  
475 <reference name="StockQuoteService">  
476 <interface.java interface="services.stockquote.StockQuoteService"/>  
477 <binding.ws uri="http://www.example.org/StockQuoteService"  
478 requires="SOAP.1_2"/>  
479 </reference>
```

480 </composite>

481

482 **3.3 Example PolicySet Providing The Conversation Intent**

483 The following policy set applies to *binding.ws* and provides the conversation intent. The
484 conversation intent is provided by using WS-ReliableMessaging [WS-RM] protocol which
485 has a concept of a Sequence. This Sequence (which appears as a *wsrmp:Sequence* SOAP
486 header in the message) is used as a correlation mechanism, on the wire, to implement
487 conversational semantics.

```
488 <policySet name="WSRM-Sequence-based-conversation"  
489           provides="sca:conversation"  
490           appliesTo="sca:binding.ws">  
491   <wsp:Policy>  
492     <wsrmp:RMAssertion  
493       xmlns:wsrmp="http://docs.oasis-open.org/ws-rx/wsrmp/200608"/>  
494   </wsp:Policy>  
495 </policySet>
```

496

497 4 Transport Binding

498 The binding.ws element provides numerous ways to specify exactly how messages ought
499 to be transmitted from or to the reference or service. Those ways include references to
500 WSDL binding elements from the @wsdlElement attribute, policy intents, and even
501 vendor extensions within the binding.ws element. This section describes the defaults to
502 be used if the specific transport details are not otherwise specified.

503 4.1 Intents

504 So as to narrow the range of choices for how messages are carried, the following policy
505 intents affect the transport binding:

- 506 • SOAP
507 ~~When the SOAP intent is required, the SCA runtime MUST transmit and receive~~
508 ~~messages using SOAP. One or more SOAP versions can be used. [BWS40001]~~~~When~~
509 ~~the SOAP intent is required, the SCA runtime MUST transmit and receive messages~~
510 ~~using SOAP. One or more SOAP versions can be used. [BWS40001].~~
- 511 • SOAP.1_1
512 ~~When the SOAP.1_1 intent is required, the SCA runtime MUST transmit and receive~~
513 ~~messages using only SOAP 1.1. When the SOAP.1_1 intent is required, the SCA~~
514 ~~runtime MUST transmit and receive messages using only SOAP 1.1~~
515 ~~[BWS40002]. [BWS40002].~~
- 516 • SOAP.1_2
517 ~~When the SOAP.1_2 intent is required, the SCA runtime MUST transmit and receive~~
518 ~~messages using only SOAP 1.2. When the SOAP.1_2 intent is required, the SCA~~
519 ~~runtime MUST transmit and receive messages using only SOAP 1.2~~
520 ~~[BWS40003]. [BWS40003].~~

Formatted: Font color: Red

Formatted: Font color: Red

Formatted: Font color: Red

521 4.2 Default Transport Binding Rules

522 4.2.1 WS-I Basic Profile Alignment

523 To align to WS-I Basic Profile, the resulting WSDL port needs to be all document-literal,
524 or all rpc-literal binding (R2705). This means, for any given portType, for all messages
525 referenced by all operations in that portType, either

- 526 • that every message part references an XML Schema type (rpc-literal pattern)
- 527 • or that every message references exactly zero or one XML Schema elements
528 (document-literal pattern)

529 ~~For an SCA service or reference element, the portType from the service's or reference's~~
530 ~~interface or derived from the that interface MUST follow either the rpc-literal or~~
531 ~~document-literal pattern. [BWS40004]~~~~For an SCA service or reference element, the~~
532 ~~portType from the service's or reference's interface or derived from the that interface~~
533 ~~MUST follow either the rpc-literal or document-literal pattern. [BWS40004].~~

Formatted: Font color: Red

534 The rest of this section assumes the short-hand reference of a "rpc-literal" or
535 "document-literal" pattern, depending on which of the two bullet points above it
536 matches.

537 **4.2.2 Default Transport Binding Rules**

538 The following defines the **default transport binding rules** for the Web Service binding:

- 539
- 540 • HTTP-based transfer protocol;
 - 541 • SOAP 1.1 binding;
 - 542 • "literal" format as described in section 3.5 of [WSDL11][WSDL11];
 - 543 • Either the document literal or rpc literal pattern, depending on the service or reference interface as described in section 4.2.1;
 - 544 o For document literal pattern, each message uses "document" style, as per
 - 545 section 3.5 of [WSDL11];
 - 546 o For rpc-literal pattern, each message uses "rpc" style, as per section 3.5
 - 547 of [WSDL11] and the child elements of the SOAP Body element are
 - 548 namespace qualified with a non-empty namespace name;
 - 549 • For SOAP 1.1 messages, the SOAPAction HTTP header described in section 6.1.1
 - 550 of "Simple Object Access Protocol (SOAP) 1.1" [SOAP11][SOAP11] represents the
 - 551 empty string, in quotes ("");
 - 552 • For SOAP 1.2 messages, the SOAP Action feature described in section 6.5 of
 - 553 [SOAP12Adjuncts] does not appear;
 - 554 • All WSDL message parts are carried in the SOAP body.

Formatted: Font: Not Bold

Formatted: Font: Not Bold

555 **In the event that the transport details are not otherwise determined, an SCA runtime**

556 **MUST enable the default transport binding rules. In the event that the transport details**

557 **are not otherwise determined, an SCA runtime MUST enable the default transport**

558 **binding rules.** [BWS40005][BWS40005].:

Formatted: Font color: Red

Formatted: Font color: Red

Formatted: Indent: Left: 0.5"

559 **When using the default transport binding rules, the SCA runtime MAY provide**

560 **additional WSDL bindings, unless policy is applied that explicitly restricts this. When**

561 **using the default transport binding rules, the SCA runtime MAY provide additional**

562 **WSDL bindings, unless policy is applied that explicitly restricts this**

563 **[BWS40006][BWS40006].**

Formatted: Font color: Red

564 **When using the default transport binding rules with the rpc-literal pattern, the SCA**

565 **runtime SHOULD use the structural URI associated with the binding as the**

566 **namespace of the child elements of the SOAP body element.** [BWS40007]

567 **When using the default transport binding rules with the rpc-literal pattern, the SCA**

568 **runtime SHOULD use the structural URI associated with the binding as the**

569 **namespace of the child elements of the SOAP body element.** [BWS40007].

Formatted: Font color: Red

570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600

5 Conformance

The XML schema pointed to by the RDDL document at the namespace URI, defined by this specification, are considered to be authoritative and take precedence over the XML schema defined in the appendix of this document.

There are two categories of artifacts for which this specification defines conformance:

a) SCA WS Binding XML Document

b) SCA Runtime

Any SCA runtime that claims to support this binding MUST abide by the requirements of this specification [BWS50001].

The normative web services binding XML Schema can be obtained by dereferencing the XML Schema namespace, and is also included for convenience in Appendix A. The <binding.ws> element MUST be valid according to its XML Schema.

5.1 SCA WS Binding XML Document

An SCA WS Binding XML document is an SCA Composite Document, or an SCA ComponentType Document, as defined by the SCA Assembly specification Section 13.1 [SCA-Assembly], that uses the <binding.ws> element.

An SCA WS Binding XML document MUST be a conformant SCA Composite Document or a SCA ComponentType Document, as defined by the SCA Assembly specification [SCA-ASSEMBLY], and MUST comply with all the applicable requirements specified in this specification.

5.2 SCA Runtime

An implementation that claims to conform to the requirements of an SCA Runtime defined in this specification has to meet the following conditions:

1. The implementation MUST comply with all statements in Appendix XXX: Conformance Items related to an SCA Runtime, notably all "MUST" statements have to be implemented.
2. The implementation MUST conform to the SCA Assembly Model Specification Version 1.1 [SCA-Assembly], and to the SCA Policy Framework Version 1.1 [SCA-Policy].
3. The implementation MUST reject a SCA WS Binding XML Document that is not conformant per Section 5.1.

Formatted: Heading 2,H2, Indent: Left: 0"

Formatted: Indent: Left: 0.25"

Formatted: Heading 2,H2, Indent: Left: 0"

Formatted: Numbered + Level: 1 +
Numbering Style: 1, 2, 3, ... + Start at: 1 +
Alignment: Left + Aligned at: 0.5" + Indent at:
0.75"

A. Web Services XML Binding Schema: sca-binding-webservice.xsd

```
601 <?xml version="1.0" encoding="UTF-8"?>
602 <!-- (c) Copyright (C) OASIS 20056, 20098. All Rights Reserved.
603 OASIS trademark, IPR and other policies apply.-->
604
605 <schema xmlns="http://www.w3.org/2001/XMLSchema"
606 targetNamespace="http://docs.oasis-open.org/ns/opencsa/sca/200712200903"
607 xmlns:sca="http://docs.oasis-open.org/ns/opencsa/sca/200712200903"
608 xmlns:wsdl="http://www.w3.org/ns/wsdl-instance"
609 xmlns:wsa="http://www.w3.org/2005/08/addressing"
610 elementFormDefault="qualified">
611
612 <import namespace="http://www.w3.org/ns/wsdl-instance"
613 schemaLocation="http://www.w3.org/2007/05/wsdl/wsdl20-
614 instance.xsd"
615 />
616 <import namespace="http://www.w3.org/2005/08/addressing"
617 schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd"
618 />
619 <include schemaLocation="sca-core-1.1-cd03.xsd"/>
620
621 <element name="binding.ws" type="sca:WebServiceBinding"
622 substitutionGroup="sca:binding"/>
623 <complexType name="WebServiceBinding">
624 <complexContent>
625 <extension base="sca:Binding">
626 <sequence>
627 <element ref="sca:wireFormat"
628 minOccurs="0" maxOccurs="1" />
629 <element ref="sca:operationSelector"
630 minOccurs="0" maxOccurs="1" />
631 <element name="endpointReference"
632 type="wsa:EndpointReference"
633 minOccurs="0" maxOccurs="unbounded"/>
634 <any namespace="##other" processContents="lax"
635 minOccurs="0" maxOccurs="unbounded"/>
636 </sequence>
637 <attribute name="wsdlElement" type="anyURI" use="optional"/>
638 <attribute ref="wsdl:wsdlLocation" use="optional"/>
639 <anyAttribute namespace="##any" processContents="lax"/>
640 </extension>
641 </complexContent>
642 </complexType>
643 </schema>
```

B. Conformance Items

This section contains a list of conformance items for the SCA Web Service Binding specification.

Conformance ID	Description
[BWS20001]	For an SCA reference, the @uri attribute MUST be an absolute value.
[BWS20002]	The value of the @wsdlElement attribute MUST identify an element in an existing WSDL 1.1 document.
[BWS20003]	If the binding is for an SCA service, the wsdlElement attribute MUST NOT specify the wsdl.service form of URI.
[BWS20004]	If the wsdl.service form of wsdlElement is used on an SCA reference binding, the set of available ports for the reference MUST contain at least one port.
[BWS20005]	If the wsdl.service form of wsdlElement is used on an SCA reference binding, the SCA runtime MUST raise an error if there are no available ports that it supports.
[BWS20006]	When an invocation is made using an SCA reference binding with the wsdl.service form of wsdlElement, the SCA runtime MUST use exactly one port from the set of available ports for the reference (with port selection on a per-invocation basis permitted).
[BWS20007]	If the binding is for an SCA service, the portType associated with the specified WSDL port MUST be compatible with the SCA service interface as defined in section 2.1, and the port MUST satisfy all the policy constraints of the binding.
[BWS20008]	The SCA runtime MUST expose an endpoint for the specified WSDL port, or raise an error if it does not support the WSDL port.
[BWS20009]	If the binding is for an SCA reference, the portType associated with the specified WSDL port MUST be a compatible superset of the SCA reference interface as defined in the SCA Assembly Model specification [SCA-Assembly], and the port MUST satisfy all the policy constraints of the binding.
[BWS20010]	The SCA runtime MUST use the specified WSDL port for invocations made using the SCA reference, or raise an error if it does not support the WSDL port.
[BWS20011]	If the binding is for an SCA service, the portType associated with the specified WSDL binding MUST be compatible with the SCA service interface as defined in section 2.1, and the WSDL binding MUST satisfy all the policy constraints of the binding.
[BWS20012]	The SCA runtime MUST expose an endpoint for the specified WSDL binding, or raise an error if it does not support the WSDL binding.
[BWS20013]	If the binding is for an SCA reference, the portType associated with the specified WSDL binding MUST be a compatible superset of the SCA reference interface as defined in the SCA Assembly Model specification [SCA-Assembly], and the WSDL binding MUST satisfy all the policy constraints of the binding.
[BWS20014]	The SCA runtime MUST use the specified WSDL binding for invocations made using the SCA reference, or raise an error if it does not support the WSDL binding.

Formatted: Highlight

Formatted: Font: Not Bold

Formatted: Highlight

Formatted: Tab stops: 0.88", Left

Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

[BWS20015]	When the <i>wSDL.binding</i> form of <i>wSDLElement</i> is used, the endpoint address URI for an SCA reference MUST be specified by either the <i>@uri</i> attribute on the binding or a WS-Addressing <i>EndpointReference</i> element, except where the SCA Assembly Model specification [SCA-Assembly] states that the <i>@uri</i> attribute can be omitted.
[BWS20016]	The <i>@wsdl:wsdlLocation</i> attribute MAY be specified by the binding in the event that the <i><WSDL-namespace-URI></i> in the 'endpoint' attribute is not dereferencable, or when the intended WSDL document is to be found at a different location than the one pointed to by the <i><WSDL-namespace-URI></i> .
[BWS20017]	If the <i>@wsdl:wsdlLocation</i> attribute is used the <i>@wsdlElement</i> attribute MUST also be specified.
[BWS20018]	The semantics of this attribute are specified in Section 7.1 of WSDL 2.0 [WSDL20]. The value of the <i>@wsdl:wsdlLocation</i> attribute MUST identify an existing WSDL 1.1 document.
[BWS20019]	A <i>binding.ws</i> element MUST NOT contain more than one of any of the following: <ul style="list-style-type: none"> the <i>@uri</i> attribute the <i>@wsdlElement</i> attribute referring to a WSDL port or to a WSDL service the <i>endpointReference</i> element
[BWS20020]	For the <i>callback</i> element of an SCA service, the binding MUST NOT specify an endpoint address URI or a WS-Addressing <i>EndpointReference</i> .
[BWS20021]	The SCA runtime MUST support all the attributes of the <i><binding.ws></i> element, namely <i>@name</i> , <i>@uri</i> , <i>@requires</i> , <i>@policySets</i> , <i>@wsdlElement</i> , and <i>@wsdl:wsdlLocation</i> .
[BWS20022]	The SCA runtime SHOULD support the element <i><endpointReference></i> .
[BWS20023]	If an SCA runtime does not support the element <i><endpointReference></i> , then it MUST reject an SCA WS Binding XML document (as defined in Section 5.1) that contains the element.
[BWS20024]	The <i><binding.ws></i> element MUST conform to the XML schema defined in <i>sca-binding-webservice.xsd</i> .
[BWS20025]	If there is no target address for a reference binding, the SCA runtime MUST raise an error.
[BWS20026]	For a reference binding, the SCA runtime MUST use the target address.
[BWS20027]	When <i>binding.ws</i> is used on a service or reference with an interface that is not defined by <i>interface.wSDL</i> , the SCA runtime MUST derive a WSDL <i>portType</i> for the service or reference from the interface using the rules defined for that SCA interface type.
[BWS20028]	An SCA runtime MUST raise an error if the interface on a service or reference element with a <i>binding.ws</i> element does not map to a WSDL <i>portType</i> .
[BWS20029]	Any service hosted by an SCA runtime with one or more web service bindings with HTTP endpoints SHOULD return a WSDL description of the service in response to an HTTP GET request with the "?wsdl" suffix to that HTTP

Formatted: Highlight

Formatted: Tab stops: 1.19", Left

Formatted: Highlight

Formatted: Font: Not Bold, Highlight

Formatted: Highlight

Formatted: Highlight

Formatted: Bulleted + Level: 1 + Aligned at: 0.5" + Indent at: 0.75"

Formatted: Tab stops: 0.99", Left

Formatted: Highlight

Formatted: Font: Verdana, Font color: Auto, Highlight

Formatted: Highlight

Formatted: Highlight

	<u>endpoint</u> .
[BWS20030]	If none of the web service bindings for an SCA service have HTTP endpoints, then the SCA runtime SHOULD provide some other means of obtaining the WSDL description of the service.
[BWS20031]	SCA runtime implementations MAY provide additional metadata that is associated with a web service binding.
[BWS20032]	An SCA runtime MUST support the WSDL extensions defined in the namespace associated with the prefix "sca" (as defined in section 1.1).
[BWS20033]	The SCA runtime MUST support the WSDL 1.1 binding extension for SOAP 1.1 over HTTP [WSDL11], as identified by the WSDL element <code>wsoap11:binding</code> that has the <code>@transport</code> attribute with a value of "http://schemas.xmlsoap.org/soap/http".
[BWS20034]	The SCA runtime SHOULD support the WSDL 1.1 binding extension for SOAP 1.2 over HTTP [WSDL11-SOAP12], as identified by the WSDL element <code>wsoap12:binding</code> that has the <code>@transport</code> attribute with a value of "http://schemas.xmlsoap.org/soap/http".
[BWS20035]	The <code><bindingType></code> element associated with this binding MUST include the SOAP.1.1 intent in its <code>@mayProvides</code> or <code>@alwaysProvides</code> attributes.
[BWS20036]	The <code><bindingType></code> element associated with this binding SHOULD include the SOAP.1.2 intent in its <code>@mayProvides</code> attribute.
[BWS20037]	The SCA runtime MUST raise an error if a web service binding is configured with a policy intent(s) that conflicts with the binding instance's configuration.
[BWS40001]	When the SOAP intent is required, the SCA runtime MUST transmit and receive messages using SOAP. One or more SOAP versions can be used.
[BWS40002]	When the SOAP.1.1 intent is required, the SCA runtime MUST transmit and receive messages using only SOAP 1.1.
[BWS40003]	When the SOAP.1.2 intent is required, the SCA runtime MUST transmit and receive messages using only SOAP 1.2.
[BWS40004]	For an SCA service or reference element, the portType from the service's or reference's interface or derived from the that interface MUST follow either the rpc-literal or document-literal pattern.
[BWS40005]	In the event that the transport details are not otherwise determined, an SCA runtime MUST enable the default transport binding rules.
[BWS40006]	When using the default transport binding rules, the SCA runtime MAY provide additional WSDL bindings, unless policy is applied that explicitly restricts this.
[BWS40007]	When using the default transport binding rules with the rpc-literal pattern, the SCA runtime SHOULD use the structural URI associated with the binding as the namespace of the child elements of the SOAP body element.
[BWS20001]	The <code>@wsdl:wsdlLocation</code> attribute MAY be specified by the binding in the event that the <code><WSDL-namespace-URI></code> in the <code>'endpoint'</code> attribute is not dereferencable, or when the intended WSDL document is to be found at a different location than the one pointed to by the <code><WSDL-namespace-URI></code> .
[BWS20002]	When the <code>endpointReference</code> element is present along with the <code>@wsdlElement</code> attribute on the parent element, the <code>@wsdlElement</code> attribute value MUST be of

Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

	the 'Binding' form <WSDL-namespace-URI>#wsdl:binding(<binding-name>)
[BWS20003]	An SCA runtime MUST raise an error if the interface on a service or reference element with a binding:ws element does not map to a WSDL portType
[BWS20004]	Any service hosted by an SCA runtime with one or more web service bindings with HTTP endpoints SHOULD return a WSDL description of the service in response to an HTTP GET request with the "?wsdl" suffix to that HTTP endpoint
[BWS20005]	If none of the web service bindings for an SCA service have HTTP endpoints, then the SCA runtime SHOULD provide some other means of obtaining the WSDL description of the service
[BWS20006]	SCA runtime implementations MAY provide additional metadata that is associated with a web service binding
[BWS20007]	An SCA runtime MUST support the WSDL extensions defined in the namespace associated with the prefix "sca"
[BWS20008]	The SCA runtime MUST raise an error if a web service binding is configured with a policy intent(s) that conflicts with the binding instance's configuration
[BWS20009]	If the @wsdl:wsdlLocation attribute is used the @wsdl:Element attribute MUST also be specified
[BWS20010]	When binding:ws is used on a service or reference with an interface that is not defined by interface:wsdl, the SCA runtime MUST derive a WSDL portType for the service or reference from the interface using the rules defined for that SCA interface type
[BWS40001]	When the SOAP intent is required, the SCA runtime MUST transmit and receive messages using SOAP. One or more SOAP versions can be used
[BWS40002]	When the SOAP_1_1 intent is required, the SCA runtime MUST transmit and receive messages using only SOAP 1.1
[BWS40003]	When the SOAP_1_2 intent is required, the SCA runtime MUST transmit and receive messages using only SOAP 1.2
[BWS40004]	For an SCA service or reference element, the portType from the service's or reference's interface or derived from the that interface MUST follow either the rpc-literal or document-literal pattern
[BWS40005]	In the event that the transport details are not otherwise determined, an SCA runtime MUST enable the default transport binding rules
[BWS40006]	When using the default transport binding rules, the SCA runtime MAY provide additional WSDL bindings, unless policy is applied that explicitly restricts this
[BWS40007]	When using the default transport binding rules with the rpc-literal pattern, the SCA runtime SHOULD use the structural URI associated with the binding as the namespace of the child elements of the SOAP body element
[BWS50001]	Any SCA runtime that claims to support this binding MUST abide by the requirements of this specification

651 **C. Appendix - WSDL Generation**

652 Due to the number of factors that determine how a WSDL might be generated, including
653 compatibility with existing WSDL uses, precise details cannot be specified. For example,
654 implementation decisions can affect the way WSDL might be generated. For reference,
655 and consistency, this section suggests non-normative choices for some of the various
656 details involved in generating WSDL. For brevity, the following definitions apply:

- 657 • component name = the value of the @name attribute of the component element
658 containing the binding.ws element
- 659 • service name = the value of the @name attribute of the service element
660 containing the binding.ws element
- 661 • binding name = the value of @name attribute of the binding.ws element, or the
662 default if no @name attribute is present
- 663 • SOAP version = either "SOAP11" or "SOAP12" as appropriate

664 With those definitions in place, here are the suggested choices:

- 665 • wsdl:definitions/@name = <component name> + "." + <service name>
- 666 • wsdl:definitions/@targetNamespace = <structural URI for the service>
- 667 • import each WSDL 1.1 portType, rather than putting them inline
- 668 • wsdl:binding/@name = <binding name> + <SOAP version> + "Binding"
- 669 • wsdl:service/@name = <service name>
- 670 • wsdl:port/@name = <binding name> + <SOAP version> + "Port"

671

D. Acknowledgements

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

674 **Participants:**

675 [Participant Name, Affiliation | Individual Member]

676 [Participant Name, Affiliation | Individual Member]

677

E. Non-Normative Text

679

F. Revision History

680 [optional; should not be included in OASIS Standards]

Revision	Date	Editor	Changes Made
1	2007-09-25	Anish Karmarkar	Applied the OASIS template + related changes to the Submission
2	2008-04-02	Anish Karmarkar	<ul style="list-style-type: none"> * Partially applied the resolution of issue 14 in the conformance section. * Applied resolution to issue 9. * Applied resolution to issue 15. * Applied resolution to issue 16. * Applied resolution to issue 10. * Applied resolution to issue 8. * Applied resolution to issue 3.
3	2008-06-12	Simon Holdsworth	<ul style="list-style-type: none"> * Completed application of resolution to issue 10 * Applied most of the editorial changes from Eric Johnson's review
4	2008-08-13	Anish Karmarkar	<ul style="list-style-type: none"> * Applied rest of Eric Johnson's ed review comments. * Applied resolution of issue 13. * Reapplied resolution of issue 15 (it was not applied correctly before) * Applied resolution of issue 19. * Applied resolution of issue 30. * Applied resolution of issue 32. * Applied resolution of issue 36. * Applied resolution of issue 38.
cd01-rev1	2008-10-16	Simon Holdsworth	Applied resolution of issue 41.
cd01-rev2	2008-10-20	Anish Karmarkar	Added rfc2119 statements.
cd01-rev3	2008-11-19	Anish Karmarkar	Incorporated feedback from Bryan, Eric & Dave
cd01-rev3	2008-12-02	Anish Karmarkar	Removed 'required' word associated with description of pseudo-schema + changed section 2.6 (wsdl extensibility) per the TC decision. Both of these were associated with issue 51 (2119 strmts)
cd01-rev5	2009-02-06	Simon Holdsworth	<ul style="list-style-type: none"> Applied resolution of issue 11 Applied resolution of issue 49 Applied action item 20080904-1
cd02	2009-02-16	Simon Holdsworth	Renamed, applied editorial issues
cd02-rev1	2009-06-02	Anish Karmarkar	* Applied resolution of issue 61 by using the

			<p>document at http://www.oasis-open.org/apps/org/workgroup/sca-bindings/download.php/32160/sca-binding-ws-1.1-spec-cd02-issue61-rev3.doc as the base document.</p> <p>* Updated NS URI (Applied action item 20090311-2).</p> <p>* Updated Copyright statement in various places.</p> <p>* Updated schema per http://lists.oasis-open.org/archives/sca-bindings/200903/msg00057.html (Applied action item 20090312-1).</p> <p>* Applied resolution of issue 23, 25, 43, 54, 55, 64.</p> <p>* Replaced 3 occurrences of 'required' with 'specified'.</p> <p>* Recreated all bookmarks, cross-references, and conformance item table.</p>
--	--	--	---

Field Code Changed

681