



# Service Component Architecture Web Service Binding Specification Version 1.1

## Committee Draft 02 Revision 3

11th June, 16th February, 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 supersedes:

- 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/200903>

sca-binding-ws-1.1-spec-cd02-~~rev3~~

Copyright © OASIS® ~~2005, 2009, 2006, 2008~~. All Rights Reserved.

11th June

Page 1 of

<http://docs.oasis-open.org/ns/opencsa/sca/200712>

**Abstract:**

The SCA Web Service binding specified in this document applies to the services and references of an SCA composite. 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® ~~2005, 2009, 2006, 2008~~. 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 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.



~~5.2 SCA Runtime ..... C-  
— Acknowledgements ..... 21  
A. Web Services XML Binding Schema: sca-binding-webservice.xsd ..... D-  
— Non-Normative Text ..... 22  
B. Conformance Items ..... E-  
— Revision History ..... 23  
C. Appendix - WSDL Generation ..... 26  
D. Acknowledgements ..... 27  
E. Non-Normative Text ..... 28  
F. Revision History ..... 29~~

- Field Code Changed

# 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** 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.

**Comment [ask1]:** Mentioning portType here does not seem appropriate.

**Comment [ask2]:** Dave would like to get rid of this.

## 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]**.

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.

**Field Code Changed**

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   |
| wsdl                                 | "http://www.w3.org/ns/wsdl-instance"   | Defined by WSDL 2.0   |
| <del>wsoap11</del><br><del>sca</del> | <del>"http://schemas.xmlsoap.org/wsdl/soap/"</del> <del>"http://docs.oasis-open.org/ns/opencsa/sca/200712"</del> | Defined by <del>WSDL 1.1 [WSDL11]</del> <del>the SCA specifications</del> |
| <del>wsoap12</del>                   | <del>"http://schemas.xmlsoap.org/wsdl/soap12/"</del>   | Defined by <del>[W11-SOAP12]</del>  |
| <del>sca</del>                       | <del>"http://docs.oasis-open.org/ns/opencsa/sca/200903"</del>  | Defined by the <del>SCA specifications</del>                              |

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.pdf)~~  
36 ~~[spec.pdf](http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-spec.html)~~~~[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)~~  
37 ~~[spec.html](http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-spec.html)~~
- 38 [~~SCA-Policy~~] ~~[SCA-JCAA]~~ ~~[http://docs.oasis-open.org/opencsa/sca-policy/sca-policy-1.1-](http://docs.oasis-open.org/opencsa/sca-policy/sca-policy-1.1-spec.pdf)~~  
39 ~~[spec.pdf](http://docs.oasis-open.org/opencsa/sca-j/sca-javacaa-1.1-spec.html)~~~~<http://docs.oasis-open.org/opencsa/sca-j/sca-javacaa-1.1-spec.html>~~
- 40 [~~SCA-JCAA~~] ~~<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-wsdl-20010315>, W3C Note, March 15 2001.
- 43 [~~WSDL20~~] [~~WSDL~~] ~~E. Christensen et al, *Web Service Description Language (WSDL)*~~  
44 ~~*1.1*, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>, W3C Note, March 15~~  
45 ~~2001.~~
- 46 ~~R. Chinnici et al, *Web Service Description Language (WSDL) Version 2.0 Part 1:*~~  
47 ~~*Core Language*, <http://www.w3.org/TR/2007/REC-wsdl20-20070626/>, W3C~~  
48 ~~Recommendation, June 26 2007.~~
- 49 [WSI-Profiles] <http://www.ws-i.org/Profiles/BasicProfile-1.1.html>  
50 <http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>  
51 <http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0.html>  
52 <http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html>
- 53 [JAX-WS] <http://jcp.org/en/jsr/detail?id=224>
- 54 [~~SOAP11~~] ~~<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>~~
- 55 [SOAP] <http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>  
56 <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- 57 [SOAP12Adjuncts] SOAP Version 1.2 Part 2: Adjuncts (Second Edition)  
58 <http://www.w3.org/TR/soap12-part2/>
- 59 [WS-Addr] <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/>
- 60 [~~W11-SOAP12~~] ~~<http://www.w3.org/Submission/wsdl11soap12/>~~

Field Code Changed

Field Code Changed

## 62 1.3 Non-Normative References

- 63 [WSI-AP] <http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>

64 [MTOM] <http://www.w3.org/TR/2005/REC-soap12-mtom-20050125/>

## 65 **1.4 Naming Conventions**

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

- 69 1. Where the names of elements and attributes consist partially or wholly of  
70 acronyms, the letters of the acronyms use the same case. When the acronym  
71 appears at the start of the name of an element or an attribute, or after a period,  
72 it is in lower case. If it appears elsewhere in the name of an element or an  
73 attribute, it is in upper case. For example, an attribute might be named "uri" or  
74 "jndiURL".
- 75 2. Where the names of types consist partially or wholly of acronyms, the letters of  
76 the acronyms are in all upper case. For example, an XML Schema type might be  
77 named "JCABinding" or "MessageID".
- 78 3. Values, including local parts of QName values, follow the rules for names of  
79 elements and attributes as stated above, with the exception that the letters of  
80 acronyms are in all upper case. For example, a value might be "JMSDefault" or  
81 "namespaceURI".

82 [WS-RM] <http://docs.oasis-open.org/ws-rx/wsrn/200702/wsrn-1.2-spec-cd-01.html>

## 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][SCA-Assembly].
- **/binding.ws/@requires** - as defined in the SCA Assembly Specification [SCA-Assembly][SCA-Assembly].
- **/binding.ws/@policySets** - as defined in the SCA Assembly Specification [SCA-Assembly][SCA-Assembly].
- **/binding.ws/@uri** - the resolution algorithm of Section 2.22-1 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 value of the @wsdlElement attribute MUST identify an element in an existing WSDL 1.1 document. This attribute points to the specified element in an existing WSDL document. 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 In this case, the SCA runtime MUST make all the ports in the WSDL service that have equivalent 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 SCA service or reference available 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

129 binding with the *wSDL.service* form of *wSDLElement*, the SCA runtime MUST use  
 130 exactly one port from the set of available ports for the reference (with port  
 131 selection on a per-invocation basis permitted). [BWS20006] SCA service or  
 132 reference.

- 133 o Port: (WSDL 1.1):

134 <WSDL-namespace-URI>#wSDL.port(<service-name>/<port-name>)

135 If the binding is for an SCA service, the portType associated with the specified  
 136 WSDL port MUST be compatible with the SCA service interface as defined in  
 137 section 2.1, and the port MUST satisfy all the policy constraints of the  
 138 binding. [BWS20007] The SCA runtime MUST expose an endpoint for the specified  
 139 WSDL port, or raise an error if it does not support the WSDL port. [BWS20008] If  
 140 the binding is for an SCA reference, the portType associated with the specified  
 141 WSDL port MUST be a compatible superset of the SCA reference interface as  
 142 defined in the SCA Assembly Model specification [SCA-Assembly], and the port  
 143 MUST satisfy all the policy constraints of the binding. [BWS20009] The SCA  
 144 runtime MUST use the specified WSDL port for invocations made using the SCA  
 145 reference, or raise an error if it does not support the WSDL port. [BWS20010]

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

- 150 o Endpoint (WSDL 2.0):

151 <WSDL-namespace-URI>#wSDL.endpoint(<service-name>/<endpoint-name>)

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

- 156
- 157 o Binding:

158 <WSDL-namespace-URI>#wSDL.binding(<binding-name>)

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

165 If the binding is for an SCA reference, the portType associated with the specified  
 166 WSDL binding MUST be a compatible superset of the SCA reference interface as  
 167 defined in the SCA Assembly Model specification [SCA-Assembly], and the  
 168 WSDL binding MUST satisfy all the policy constraints of the binding.  
 169 [BWS20013] The SCA runtime MUST use the specified WSDL binding for  
 170 invocations made using the SCA reference, or raise an error if it does not support  
 171 the WSDL binding. [BWS20014]

172 When the *wSDL.binding* form of *wSDLElement* is used, the endpoint address URI  
 173 for an SCA reference MUST be specified by either the *@uri* attribute on the  
 174 binding or a WS-Addressing *EndpointReference* element, except where the SCA  
 175 Assembly Model specification [SCA-Assembly] states that the *@uri* attribute can  
 176 be omitted. [BWS20015]

177 In this case, the WSDL binding identified by the `<binding.ws>` element MUST  
178 implement a portType that is equivalent to the one specified for the SCA service  
179 or reference. The SCA runtime MUST make the service or reference available via  
180 the specified WSDL binding. In this case, the endpoint address URI for an SCA  
181 reference MUST be specified by either the `@uri` attribute on the binding or a WS-  
182 Addressing `EndpointReference` element, except where the SCA Assembly  
183 specification states that the `@uri` attribute can be omitted. The endpoint address  
184 URI for an SCA service or the callback element of an SCA reference is determined  
185 as specified in section 2.1. For the `callback` element of an SCA service, the  
186 binding MUST NOT specify an endpoint address URI or a WS-Addressing  
187 `EndpointReference`.

- **`/binding.ws/@wsdl:wsdlLocation`** – when present this attribute specifies the location(s) of the WSDL document(s) associated with specific namespace(s). This attribute MAY be specified by the binding in the event that the `<WSDL-namespace-URI>` 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 `<WSDL-namespace-URI>`. The use of this attribute indicates that the WSDL binding points to an existing WSDL document. The semantics of this attribute are specified in Section 7.1 of WSDL 2.0 [WSDL].

The `@wsdl:wsdlLocation` attribute MAY be specified by the binding in the event that the `<WSDL-namespace-URI>` 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 `<WSDL-namespace-URI>`. [BWS20016]

If the `@wsdl:wsdlLocation` attribute is used the `@wsdlElement` attribute MUST also be specified. [BWS20017] The semantics of this attribute are specified in Section 7.1 of WSDL 2.0 [WSDL20]. The value of the `@wsdl:wsdlLocation` attribute MUST identify an existing WSDL 1.1 document. [BWS20018]

- **`/binding.ws/wireFormat`** – as defined in the SCA Assembly Specification [SCA-Assembly]. This specification does not define any new `wireFormat` elements.
- **`/binding.ws/operationSelector`** – as defined in the SCA Assembly Specification [SCA-Assembly]. This specification does not define any new `operationSelector` elements.
- **`/binding.ws/endpointReference`** – when present this element provides the WS-Addressing [WS-Addr][WS-Addr] `EndpointReference` that specifies the endpoint for the service or reference. When this element is present along with the `@wsdlElement` attribute on the parent element, the `@wsdlElement` attribute value MUST be of the `'Binding'` form as specified above, i.e. `<WSDL-namespace-URI>#wsdl.binding(<binding-name>)`.
- **`/binding.ws/@{any}`** – this is an extensibility mechanism to allow extensibility via attributes.
- **`/binding.ws/any`** – this is an extensibility mechanism to allow extensibility via elements.

A `binding.ws` element MUST NOT contain more than one of any of the following: the `@uri` attribute; the `@wsdlElement` attribute referring to a WSDL port or to a WSDL service; the `endpointReference` element. [BWS20019]

The endpoint address URI for an SCA service or the callback element of an SCA reference is determined as specified in section 2.2. For the `callback` element of an SCA service, the binding MUST NOT specify an endpoint address URI or a WS-Addressing `EndpointReference`. [BWS20020]

226 The SCA runtime MUST support all the attributes of the <binding.ws> element, namely  
227 @name, @uri, @requires, @policySets, @wsdlElement, and  
228 @wsdl:wsdlLocation. [BWS20021]  
229 The SCA runtime SHOULD support the element <endpointReference>. [BWS20022] If an  
230 SCA runtime does not support the element <endpointReference>, then it MUST reject  
231 an SCA WS Binding XML document (as defined in Section 5.1) that contains the element.  
232 [BWS20023]  
233 The <binding.ws> element MUST conform to the XML schema defined in sca-binding-  
234 webservice.xsd. [BWS20024]

## 235 **2.1 Compatibility of SCA Service Interfaces and WSDL portTypes**

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

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

## 255 **2.1.2 Endpoint URI resolution**

256 This specification does not mandate any particular way to determine  
257 the rules for resolving the URI for a web services binding on at which an SCA service.  
258 An absolute is hosted, or SCA reference targets, when used with binding.ws (in precedence order) are:

- 259 1. The URIs in the endpoint(s) of the referenced WSDL  
260 or  
261 The URI can be indicated specified by the @uri attribute, by the URI in a  
262 wsa:Address element within an of the endpointReference,
- 263 2. The explicitly stated URI in the @uri attribute of the binding.ws element, or  
264 which can be relative;
- 265 3. The structural URI as defined by the URI indicated in a Assembly specification

266 An SCA runtime MUST follow rules listed above in determining the URI at which an SCA  
267 service is hosted or an SCA reference is targeted.

268 The URI in the WSDL Port via endpoint or in the wsa:Address of an EPR MAY be a relative  
269 URI, in which case it is relative to the URI defined in (2) or (3). The wsa:Address  
270 element MAY be the empty relative URI, in which case it uses the URI defined in (2) or

271 (3) directly. This enables the EPR writer to specify reference parameters, metadata and  
272 other EPR contents while letting the deployer choose the URI.

273 To reference a WSDL document and also specify an EPR, the `@wsdlElement` attribute.  
274 Implementations can use the specified URI as the service endpoint URI or they can use a  
275 different URI which might include portions of the specified URI. For example, the service  
276 endpoint URI might be produced by modifying any or all of the host name, the port  
277 number, and a portion of the path.

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

283 The target address for a reference ~~must~~ refer to a binding is defined as one of the  
284 following:

- 285 A. The value of the `@uri` attribute
- 286 B. The value of the `wsa:Address` element of the `endpointReference` element
- 287 C. The value of the `address` element of the WSDL port referenced by the  
288 `@wsdlElement` attribute

289 The value of the `address` element of one of the set of available WSDL ports as specified  
290 under the definition of the `@wsdlElement` attribute when it references a ~~in the~~ WSDL  
291 service element.

292 If there is no target address for a reference binding, the SCA runtime MUST raise an  
293 error. [BWS20025]

294 For a reference binding, the SCA runtime MUST use the target address. [BWS20026]

### 295 2.22.3 Interface mapping

296 When `binding.ws` is used on a service or reference with an interface that is not defined  
297 by `interface.wsdl`, the SCA runtime MUST derive a WSDL portType for the service or  
298 reference from the interface using the rules defined for that SCA interface type.  
299 [BWS20027]

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. [BWS20028]

302 When `binding.ws` is used on a service or reference with an interface that is not defined  
303 by `interface.wsdl`, then a WSDL portType for the service or reference is derived from the  
304 interface by the rules defined for that SCA interface type. An SCA runtime MUST raise  
305 an error if the interface does not map to a WSDL portType.

306 For example, for `interface.java`, the mapping to a WSDL portType is as defined in the  
307 SCA Java Common Annotations and API Specification [SCA-Policy] [SCA-JCAA].

308 `binding.ws` implementations can use appropriate standards, for example WS-I AP 1.0  
309 [WSI-AP] [WSI-AP] or MTOM [MTOM] [MTOM], to map interface parameters to binary  
310 attachments transparently to the target component.

311

312 **2.32.4 Production of WSDL description for an SCA service**

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

316 ~~If none of the web service bindings for an SCA service have HTTP endpoints, then the~~  
317 ~~SCA runtime SHOULD provide some other means of obtaining the WSDL description of~~  
318 ~~the service. [BWS20030]Any service hosted by an SCA runtime with one or more web~~  
319 ~~service bindings with HTTP endpoints SHOULD return a WSDL description of the service~~  
320 ~~in response to an HTTP GET request with the "?wsdl" suffix to that HTTP endpoint. If~~  
321 ~~none of the web service bindings have HTTP endpoints, then some other means of~~  
322 ~~obtaining the WSDL description of the service SHOULD be provided by the SCA runtime.~~  
323 This can include out of band mechanisms, for example publication to a UDDI registry.

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

327

328 **2.42.5 Additional binding configuration data**

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

331 ~~This can be used~~SCA runtime implementations ~~MAY provide additional metadata that is~~  
332 ~~associated with a web service binding,~~ for example to enable JAX-WS ~~[JAX-WS]~~**[JAX-**  
333 ~~WS]~~ handlers to be executed as part of the target component dispatch. The  
334 specification of such metadata is SCA runtime-specific and is outside of the scope of this  
335 document.

336

337 **2.52.6 Web Service Binding and SOAP Intermediaries**

338 The Web Service binding does not provide any direct or explicit support for SOAP intermediaries  
339 intermediaries ~~[SOAP11]~~ ~~[http://www.w3.org/TR/2000/NOTE-SOAP-](http://www.w3.org/TR/2000/NOTE-SOAP-20000508/)~~  
340 ~~[20000508/](http://www.w3.org/TR/2000/NOTE-SOAP-20000508/)~~  
341 ~~[SOAP]~~**[SOAP]**.

342

343 **2.62.7 Support for WSDL extensibility**

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

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

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

355 | ~~The SCA runtime SHOULD support the WSDL 1.1 binding extension for SOAP 1.2 over~~  
356 | ~~HTTP [W11-SOAP12], as identified by the WSDL element wsoap12:binding that has~~  
357 | ~~the @transport attribute with a value of "http://schemas.xmlsoap.org/soap/http".~~  
358 | ~~[BWS20034]~~

359 | ~~This specification requires that an SCA runtime MUST support the WSDL extensions~~  
360 | ~~defined in the namespace associated with the prefix "sca" (as defined in section 1.1).~~

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

## 365 | **2.72.8 Intents listed in the bindingType**

366 | This specification places no requirements on the intents that are listed as either  
367 | @alwaysProvides or @mayProvides in the bindingType for *binding.ws*.

## 368 | **2.82.9 Intents and binding configuration**

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

375 | ~~The SCA runtime MUST raise an error if a web service binding is configured with a policy~~  
376 | ~~intent(s) that conflicts with the binding instance's configuration. [BWS20037]~~

377 | ~~The SCA runtime MUST raise an error if the web service binding is configured with a~~  
378 | ~~policy intent(s) that conflicts with a binding instance's configuration. For example, it is~~  
379 | ~~an error to use the SOAP policy intent in combination with a WSDL binding that does not~~  
380 | ~~use SOAP.~~

### 381 3 Web Service Binding Examples

382 The following snippets show the sca.composite file for the MyValueComposite file  
383 containing the service element for the MyValueService and reference element for the  
384 StockQuoteService. Both the service and the reference use a Web Service binding.  
385

#### 386 3.1 Example Using WSDL documents

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

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

392 The **first** reference's **first** binding is defined by the specified WSDL service in the WSDL  
393 document at the given location. The reference can use any of the WSDL service's  
394 ports/**endpoints** to invoke the target service. The **second** reference's **second** binding is  
395 defined by the specified WSDL binding. The specific endpoint URI to be invoked is  
396 provided via the `@uri` attribute.  
397

```
398 <?xml version="1.0" encoding="ASCII"?>  
399 <composite xmlns="http://docs.oasis-  
400 open.org/ns/opencsa/sca/200903" xmlns="http://docs.oasis-  
401 open.org/ns/opencsa/sca/200712"  
402     name="MyValueComposite">  
403     <service name="MyValueService">  
404         <interface.java interface="services.myvalue.MyValueService"/>  
405         <binding.ws wsdlElement="http://www.example.org/MyValueService#  
406             wsdl.binding(MyValueService/MyValueServiceSOAP)"/>wsdl.endpoint(MyValueService  
407             /MyValueServiceSOAP)"/>  
408         ...  
409     </service>  
410     ...  
411     <reference name="StockQuoteReference1">  
412         <interface.java interface="services.stockquote.StockQuoteService"/>  
413         <binding.ws wsdlElement="http://www.example.org/StockQuoteService#  
414             wsdl.service(StockQuoteService) "  
415             wsdlLocation="http://www.example.org/StockQuoteService  
416             http://www.example.org/StockQuoteService.wsdl"/>  
417         </reference>  
418     <reference name="StockQuoteReference2">  
419         <interface.java interface="services.stockquote.StockQuoteService"/>  
420         <binding.ws wsdlElement="http://www.example.org/StockQuoteService#  
421             wsdl.binding(StockQuoteBinding) "  
422             wsdlLocation="http://www.example.org/StockQuoteService  
423             http://www.example.org/StockQuoteService.wsdl"  
424             uri="http://www.example.org/StockQuoteService5"/>  
425         </reference>  
426     </composite>
```

### 431 3.2 Examples Without a WSDL Document

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

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

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

```
444  
445 <?xml version="1.0" encoding="ASCII"?>  
446 <composite xmlns="http://docs.oasis-  
447 open.org/ns/opencsa/sca/200903" xmlns="http://docs.oasis-  
448 open.org/ns/opencsa/sca/200712"  
449     name="MyValueComposite">  
450  
451     <service name="MyValueService">  
452         <interface.java interface="services.myvalue.MyValueService"/>  
453         <binding.ws/>  
454         ...  
455     </service>  
456  
457     ...  
458  
459     <reference name="StockQuoteService">  
460         <interface.java interface="services.stockquote.StockQuoteService"/>  
461         <binding.ws uri="http://www.example.org/StockQuoteService"/>  
462     </reference>  
463 </composite>
```

464 The next example shows the use of the binding element without a WSDL document, with  
465 multiple SOAP bindings with non-default values. The SOAP 1.2 binding name defaults to  
466 the service name, the SOAP 1.1 binding is given an explicit name. The reference has a  
467 web service binding which uses SOAP 1.2, but otherwise uses all the defaults for SOAP  
468 binding. The reference binding name defaults to the reference name.

```
470  
471 <?xml version="1.0" encoding="ASCII"?>  
472 <composite xmlns="http://docs.oasis-  
473 open.org/ns/opencsa/sca/200903" xmlns="http://docs.oasis-  
474 open.org/ns/opencsa/sca/200712"  
475     name="MyValueComposite">  
476  
477     <service name="MyValueService">  
478         <interface.java interface="services.myvalue.MyValueService"/>  
479         <binding.ws name="MyValueServiceSOAP11" requires="SOAP.1_1"/>  
480         <binding.ws requires="SOAP.1_2"/>  
481         ...  
482     </service>  
483  
484     ...  
485  
486     <reference name="StockQuoteService">
```

```
487 <interface.java interface="services.stockquote.StockQuoteService"/>
488 <binding.ws uri="http://www.example.org/StockQuoteService"
489           requires="SOAP_1_2"/>
490 </reference>
491 </composite>
```

492

### 493 3.3 Example PolicySet Providing The Conversation Intent

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

```
499 <policySet name="WSRM-Sequence-based-conversation"
500           provides="sca:conversation"
501           appliesTo="sca:binding.ws">
502   <wsp:Policy>
503     <wsrmp:RMAssertion
504       xmlns:wsrmp="http://docs.oasis-open.org/ws-rx/wsrmp/200608"/>
505   </wsp:Policy>
506 </policySet>
```

507

## 508 4 Transport Binding

509 The binding.ws element provides numerous ways to specify exactly how messages ought  
510 to be transmitted from or to the reference or service. Those ways include references to  
511 WSDL binding elements from the @wsdlElement attribute, policy intents, and even  
512 vendor extensions within the binding.ws element. ~~However, all of those ways to indicate~~  
513 ~~how messages get carried happen to be optional.~~ This section describes the defaults to  
514 be used if the specific transport details are not otherwise specified.

### 515 4.1 Intents

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

- 518 • ~~SOAP~~  
519 ~~When the SOAP intent is required, the SCA runtime MUST transmit and receive~~  
520 ~~messages using SOAP. One or more SOAP versions can be used. [BWS40001]~~
- 521 • ~~SOAP.1.1~~  
522 ~~When the SOAP.1.1 intent is required, the SCA runtime MUST transmit and receive~~  
523 ~~messages using only SOAP 1.1. [BWS40002]~~
- 524 • ~~SOAP.1.2~~  
525 ~~When the SOAP.1.2 intent is required, the SCA runtime MUST transmit and receive~~  
526 ~~messages using only SOAP 1.2. [BWS40003]~~
- 527 • ~~SOAP~~  
528 ~~This indicates that messages MUST be transmitted using SOAP. One or more SOAP~~  
529 ~~versions can be used.~~
- 530 • ~~SOAP.1.1~~  
531 ~~Messages MUST be transmitted using only SOAP 1.1.~~
- 532 • ~~SOAP.1.2~~  
533 ~~Messages MUST be transmitted using only SOAP 1.2.~~

### 534 4.2 Default Transport Binding Rules

#### 535 4.2.1 WS-I Basic Profile Alignment

536 To align to WS-I Basic Profile, the resulting WSDL port needs to be all document-literal,  
537 or all rpc-literal binding (~~per WS-I Basic Profile 1.1 R2705 [WSI-Profiles], R2705~~).  
538 This means, for any given portType, for all messages referenced by all operations in that  
539 portType, either

- 540 • that every message part references an XML Schema type (rpc-literal pattern)
- 541 • or that every message references exactly zero or one XML Schema elements  
542 (document-literal pattern)

543 ~~For an SCA service or reference element, the portType from the service's or reference's~~  
544 ~~interface or derived from that interface MUST follow either the rpc-literal or document-~~  
545 ~~literal pattern. [BWS40004]~~

546 ~~For a service element, the portType from the service's interface or derived from the~~  
547 ~~service's interface MUST fit one of these two patterns.~~ The rest of this section assumes

548 | the short-hand reference of an "rpc-literal" or "document-literal" pattern, depending on  
549 | which of the two bullet points above it matches.

#### 550 | 4.2.2 Default Transport Binding Rules

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

552 | In the event that the transport details are not otherwise determined, an SCA runtime  
553 | MUST enable the following configuration:

- 554 | • HTTP-based transfer protocol;
- 555 | • SOAP 1.1 binding;
- 556 | • ~~Bindings for SOAP 1.1 MUST be provided and additional bindings MAY be~~  
557 | ~~provided, unless policy is applied that explicitly restricts this.~~
- 558 | • "literal" format as described in section 3.5 of **WSDL11**; ~~{WSDL11}~~
- 559 | • Either the document literal or rpc literal pattern, depending on the service or  
560 | reference interface as described in section 4.2.1;
- 561 | • For document literal pattern, each message uses "document" style, as per section  
562 | 3.5 of **WSDL11**; ~~{WSDL11}~~
- 563 | • For rpc-literal pattern, each message uses "rpc" style, as per section 3.5 of  
564 | **WSDL11** ~~and {WSDL11}. In this case,~~ the child elements of the SOAP Body  
565 | element ~~are~~ MUST be namespace qualified with a non-empty namespace  
566 | name; name. ~~This namespace SHOULD be the structural URI associated with the~~  
567 | ~~binding.~~
- 568 | • For SOAP 1.1 messages, the SOAPAction HTTP header described in section 6.1.1  
569 | of "Simple Object Access Protocol (SOAP) 1.1" [SOAP11] represents the empty  
570 | string, in quotes (""); ~~{""}~~.
- 571 | • For SOAP 1.2 messages, the SOAP Action feature described in section 6.5 of  
572 | **SOAP12Adjuncts** ~~{SOAP12Adjuncts}~~ does not appear; ~~;~~
- 573 | • All WSDL message parts are carried in the SOAP body.

574 | In the event that the transport details are not otherwise determined, an SCA runtime  
575 | MUST enable the default transport binding rules. [BWS40005]

576 | When using the default transport binding rules, the SCA runtime MAY provide additional  
577 | WSDL bindings, unless policy is applied that explicitly restricts this. [BWS40006]

578 | When using the default transport binding rules with the rpc-literal pattern, the SCA  
579 | runtime SHOULD use the structural URI associated with the binding as the namespace of  
580 | the child elements of the SOAP body element. [BWS40007]

581 **5 Conformance**

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

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

586 a) ~~Any~~ SCA WS Binding XML Document

587 b) SCA Runtime

588 **5.1 SCA WS Binding XML Document**

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

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

596 **5.2 SCA Runtime**

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

599 The implementation ~~support this binding~~ MUST comply with all statements in Appendix B:  
600 Conformance Items related to an SCA Runtime, notably all "MUST" statements have to  
601 be implemented. ~~abide by the requirements of this specification.~~

602 1. The implementation MUST conform to the SCA Assembly Model Specification  
603 Version 1.1 [SCA-Assembly], and to the SCA Policy Framework Version 1.1  
604 [SCA-Policy].

605 2. The implementation MUST reject a SCA WS Binding XML Document that is not  
606 conformant per Section 5.1.

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

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

```
610 <?xml version="1.0" encoding="UTF-8"?>
611 <!-- (c) Copyright(C) OASIS 2005, 2009. All Rights Reserved. 2006, 2008 -->
612     OASIS trademark, IPR and other policies apply.-->
613
614 <schema xmlns="http://www.w3.org/2001/XMLSchema"
615     targetNamespace="http://docs.oasis-
616     open.org/ns/opencsa/sca/200903" targetNamespace="http://docs.oasis-
617     open.org/ns/opencsa/sca/200712"
618     xmlns:sca="http://docs.oasis-open.org/ns/opencsa/sca/200903"
619     xmlns:sca="http://docs.oasis-open.org/ns/opencsa/sca/200712"
620     xmlns:wsdli="http://www.w3.org/ns/wsdl-instance"
621     xmlns:wsa="http://www.w3.org/2005/08/addressing"
622     elementFormDefault="qualified">
623
624     <import namespace="http://www.w3.org/ns/wsdl-instance"
625     schemaLocation="http://www.w3.org/2007/05/wsdl/wsdl20-
626     instance.xsd"
627     />
628
629     <import namespace="http://www.w3.org/2005/08/addressing"
630     schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd"
631     />
632
633     <include schemaLocation="sca-core-1.1-cd03.xsd"/>
634     <include schemaLocation="sca-
635     core.xsd"/>
636
637     <element name="binding.ws" type="sca:WebServiceBinding"
638     substitutionGroup="sca:binding"/>
639     <complexType name="WebServiceBinding">
640     <complexContent>
641     <extension base="sca:Binding">
642     <sequence>
643     <element ref="sca:wireFormat"
644     minOccurs="0" maxOccurs="1" />
645     <element ref="sca:operationSelector"
646     minOccurs="0" maxOccurs="1" />
647
648     <element name="endpointReference"
649     type="wsa:EndpointReference" type="wsa:EndpointReference"
650     minOccurs="0" maxOccurs="unbounded"/>
651     <any namespace="##other" processContents="lax"
652     minOccurs="0" maxOccurs="unbounded"/>
653     </sequence>
654     <attribute name="wsdlElement" type="anyURI" use="optional"/>
655     <attribute ref="wsdli:wsdlLocation" use="optional"/>
656     <anyAttribute namespace="##any" processContents="lax"/>
657     </extension>
658     </complexContent>
659     </complexType>
660
661 </schema>
```

663

## **B. Conformance Items**

664

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

| <b>Conformance ID</b> | <b>Description</b>   |
|-----------------------|--|
| [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  |

|            |   |
|------------|---|
|            | <b>binding.</b>   |
| [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 @uri attribute on the binding or a WS-Addressing <i>EndpointReference</i> element, except where the SCA Assembly Model specification [SCA-Assembly] states that the @uri attribute can be omitted. |
| [BWS20016] | The @wsdl:wsdlLocation attribute MAY be specified by the binding in the event that the <WSDL-namespace-URI> 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 <WSDL-namespace-URI>.   |
| [BWS20017] | If the @wsdl:wsdlLocation attribute is used the @wSDLElement attribute MUST also be specified.  |
| [BWS20018] | The value of the @wsdl:wsdlLocation attribute MUST identify an existing WSDL 1.1 document.  |
| [BWS20019] | A binding.ws element MUST NOT contain more than one of any of the following: the @uri attribute; the @wSDLElement attribute referring to a WSDL port or to a WSDL service; the endpointReference element.   |
| [BWS20020] | For the callback element of an SCA service, the binding MUST NOT specify an endpoint address URI or a WS-Addressing EndpointReference.  |
| [BWS20021] | The SCA runtime MUST support all the attributes of the <binding.ws> element, namely @name, @uri, @requires, @policySets, @wSDLElement, and @wsdl:wsdlLocation.  |
| [BWS20022] | The SCA runtime SHOULD support the element <endpointReference>.   |
| [BWS20023] | If an SCA runtime does not support the element <endpointReference>, then it MUST reject an SCA WS Binding XML document (as defined in Section 5.1) that contains the element.   |
| [BWS20024] | The <binding.ws> element MUST conform to the XML schema defined in sca-binding-webservice.xsd.  |
| [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 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.   |
| [BWS20028] | 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.  |
| [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 endpoint.  |
| [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 <b>WSDL11</b> , 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 <b>[W11-SOAP12]</b> , 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>&lt;bindingType&gt;</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>&lt;bindingType&gt;</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 that interface MUST follow either the <code>rpc-literal</code> or <code>document-literal</code> 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 <code>rpc-literal</code> 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.   |

665

## B.C. Appendix - WSDL Generation

666

667

668

669

670

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

671

672

- component name = the value of the @name attribute of the component element containing the binding.ws element

673

674

- service name = the value of the @name attribute of the service element containing the binding.ws element

675

676

- binding name = the value of @name attribute of the binding.ws element, or the default if no @name attribute is present

677

- SOAP version = either "SOAP11" or "SOAP12" as appropriate

678

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

679

- wsdl:definitions/@name = <component name> + "." + <service name>

680

- wsdl:definitions/@targetNamespace = <structural URI for the service>

681

- import each WSDL 1.1 portType, rather than putting them inline

682

- wsdl:binding/@name = <binding name> + <SOAP version> + "Binding"

683

- wsdl:service/@name = <service name>

684

- wsdl:port/@name = <binding name> + <SOAP version> + "Port"

685

---

## **C.D. Acknowledgements**

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

688 **Participants:**

689 [Participant Name, Affiliation | Individual Member]

690 [Participant Name, Affiliation | Individual Member]

691

---

**D.E. Non-Normative Text**

**E.F. Revision History**

694 [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                 | * Partially applied the resolution of issue 14 in the conformance section.<br>* Applied resolution to issue 9.<br>* Applied resolution to issue 15.<br>* Applied resolution to issue 16.<br>* Applied resolution to issue 10.<br>* Applied resolution to issue 8.<br>* Applied resolution to issue 3.  |
| 3                         | 2008-06-12                 | Simon Holdsworth                | * Completed application of resolution to issue 10<br>* Applied most of the editorial changes from Eric Johnson's review  |
| 4                         | 2008-08-13                 | Anish Karmarkar                 | * Applied rest of Eric Johnson's ed review comments.<br>* Applied resolution of issue 13.<br>* Reapplied resolution of issue 15 (it was not applied correctly before)<br>* Applied resolution of issue 19.<br>* Applied resolution of issue 30.<br>* Applied resolution of issue 32.<br>* Applied resolution of issue 36.<br>* 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                | Applied resolution of issue 11<br>Applied resolution of issue 49<br>Applied action item 20080904-1   |
| cd02                      | 2009-02-16                 | Simon Holdsworth                | Renamed, applied editorial issues  |
| <a href="#">cd02-rev1</a> | <a href="#">2009-06-02</a> | <a href="#">Anish Karmarkar</a> | <a href="#">* Applied resolution of issue 61 by using the</a>  |

|                           |                            |                                 |   |
|---------------------------|----------------------------|---------------------------------|---|
|                           |                            |                                 | <p><a href="http://www.oasis-open.org/apps/org/workgroup/sca-bindings/download.php/32160/sca-binding-ws-1.1-spec-cd02-issue61-rev3.doc">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</a> 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 <a href="http://lists.oasis-open.org/archives/sca-bindings/200903/msg00057.html">http://lists.oasis-open.org/archives/sca-bindings/200903/msg00057.html</a> (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> |
| <a href="#">cd02-rev2</a> | <a href="#">2009-06-09</a> | <a href="#">Anish Karmarkar</a> | <p>Ed. fixes. Changed the way the crossrefs/bookmarks for RFC2119 keywords work. Fixed a few references.</p>  |
| <a href="#">cd02-rev3</a> | <a href="#">2009-06-11</a> | <a href="#">Anish Karmarkar</a> | <p>* Removed ':' from 40005, reformatted 40006/40007.</p> <p>* minor ed changes pointed out by SimonN.</p> <p>* minor formatting changes.</p> <p>* modified BWS20018 to remove the first sentence.</p>  |

695