



# Service Component Architecture Web Service Binding Specification Version 1.1

## Committee Draft 01

13th August, 2008

### Specification URIs:

#### This 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)

#### Previous Version:

#### Latest Version:

<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-spec.html>  
<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-spec.doc>  
<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-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/200712>

**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® 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 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.....	5
1.1	Terminology .....	5
1.2	Normative References .....	6
2	Web Service Binding Schema.....	7
2.1	Endpoint URI resolution.....	8
2.2	Interface mapping .....	8
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 .....	9
2.8	Intents and binding configuration.....	9
3	Web Service Binding Examples .....	10
3.1	Example Using WSDL documents .....	10
3.2	Examples Without a WSDL Document.....	11
3.3	Example PolicySet Providing The Conversation Intent .....	12
4	WSDL Generation .....	13
4.1	Intents .....	13
4.2	WSDL Service and Ports .....	13
4.3	WSDL Bindings.....	13
4.3.1	SOAP versions .....	14
4.4	WSDL portType .....	14
4.5	WSDL Generation Rules .....	14
5	Conformance .....	16
A.	Web Services Binding Schema .....	17
B.	Acknowledgements .....	18
C.	Non-Normative Text .....	19
D.	Revision History.....	20

# 1 Introduction

The SCA Web Service binding specified in this document applies to the services and references of composites [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 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.

The Web Service binding can point to an existing WSDL [WSDL] document, separately authored, that specifies the details of the WSDL binding and portType schema 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 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.

In most cases it is expected that a binding applied to a composite's reference will point to an existing WSDL document that describes the web service to be invoked. The binding applied to a composite's service may use either approach.

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].

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

<b>Prefix</b>	<b>Namespace</b>	<b>Notes</b>
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

sca	"http://docs.oasis-open.org/ns/opencsa/sca/200712"	Defined by the SCA specifications
-----	--	-----------------------------------

32

## 33 1.2 Normative References

- 34       **[RFC2119]**       S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,  
35                       <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 36       **[SCA-Assembly]** <http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-spec.html>
- 37       **[SCA-JCAA]**       <http://docs.oasis-open.org/opencsa/sca-j/sca-javacaa-1.1-spec.html>
- 38       **[WSDL]**           E. Christensen et al, *Web Service Description Language (WSDL) 1.1*,  
39                       <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>, W3C Note, March 15 2001.  
40                       R. Chinnici et al, *Web Service Description Language (WSDL) Version 2.0 Part 1:*  
41                       *Core Language*, <http://www.w3.org/TR/2007/REC-wsdl20-20070626/>, W3C  
42                       Recommendation, June 26 2007.
- 43       **[WSI-Profiles]** <http://www.ws-i.org/Profiles/BasicProfile-1.1.html>  
44                       <http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>  
45                       <http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0.html>  
46                       <http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html>
- 47       **[JAX-WS]**       <http://jcp.org/en/jsr/detail?id=224>
- 48       **[SOAP]**           <http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>  
49                       <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- 50       **[WS-Addr]**       <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/>

51

52

---

## 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:QNames"
  uri="xs:anyURI"
  wsdlElement="xs:anyURI"?
  wsdl:wsdlLocation="list of xs:anyURI pairs"?
  ...>
  <wsa:EndpointReference>...</wsa: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/@uri*** - the resolution algorithm of Section 2.1 below describes how this attribute is interpreted.
- ***/binding.ws/@wsdlElement*** – optional attribute that specifies the URI of a WSDL element. The use of this attribute indicates that the SCA binding 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>)`  
In this case, all the endpoints in the WSDL Service that have equivalent portTypes with the SCA service or reference must be available to the SCA service or reference.
  - Port (WSDL 1.1):  
`<WSDL-namespace-URI>#wsdl.port(<service-name>/<port-name>)`  
In this case, the identified port in the WSDL 1.1 Service must have an equivalent portType with the SCA service or reference.
  - Endpoint (WSDL 2.0):  
`<WSDL-namespace-URI>#wsdl.endpoint(<service-name>/<endpoint-name>)`  
In this case, the identified endpoint in the WSDL 2.0 Service must have an equivalent portType with the SCA service or reference.
  - Binding:  
`<WSDL-namespace-URI>#wsdl.binding(<binding-name>)`  
In this case, the identified WSDL binding must have an equivalent portType with the SCA service or reference. In this case, the endpoint address URI for an SCA reference MUST be provided by either the *@uri* attribute on the binding or a WS-Addressing *EndpointReference* element, except where the SCA Assembly specification states that the *@uri* attribute is optional. The endpoint address URI for an SCA service or the callback element of an SCA reference is determined as

96 specified in section 2.1. For the *callback* element of an SCA service, the binding  
97 MUST NOT specify an endpoint address URI or a WS-Addressing  
98 EndpointReference..

- 99 • ***/binding.ws/@wsdl:wsdlLocation*** – optional attribute that specifies the  
100 location(s) of the WSDL document(s) associated with specific namespace(s). This  
101 attribute can be specified in the event that the <WSDL-namespace-URI> in the  
102 'endpoint' attribute is not dereferencable, or when the intended WSDL document is  
103 to be found at a different location than the one pointed to by the <WSDL-  
104 namespace-URI>. The use of this attribute indicates that the WSDL binding points  
105 to an existing WSDL document. The semantics of this attribute are specified in  
106 Section 7.1 of WSDL 2.0 [WSDL].
- 107 • ***/binding.ws/wsa:EndpointReference*** – optional WS-Addressing [WS-Addr]  
108 EndpointReference that specifies the endpoint for the service or reference. When this  
109 element is present along with the *@wsdlElement* attribute on the parent element, the  
110 *@wsdlElement* attribute value MUST be of the 'Binding' form as specified above, i.e.  
111 <WSDL-namespace-URI>#wsdl.binding(<binding-name>).
- 112 • ***/binding.ws/@{any}*** - this is an extensibility mechanism to allow extensibility via  
113 attributes.
- 114 • ***/binding.ws/any*** – this is an extensibility mechanism to allow extensibility via  
115 elements.

## 116 2.1 Endpoint URI resolution

117 The rules for resolving the URI at which an SCA service is hosted, or SCA reference  
118 targets, when used with *binding.ws* (in precedence order) are:

- 119 1. The URIs in the endpoint(s) of the referenced WSDL  
120 or  
121 The URI specified by the *wsa:Address* element of the *wsa:EndpointReference*,
- 122 2. The explicitly stated URI in the *@uri* attribute of the *binding.ws* element, which  
123 may be relative,
- 124 3. The implicit URI as defined by the Assembly specification

125 The URI in the WSDL endpoint or in the *wsa:Address* of an EPR may be a relative URI, in  
126 which case it is relative to the URI defined in (2) or (3). The *wsa:Address* element can  
127 be the empty relative URI, in which case it uses the URI defined in (2) or (3) directly.  
128 This allows the EPR writer to specify reference parameters, metadata and other EPR  
129 contents while allowing the URI to be chosen by the deployer.

130 To reference a WSDL document and also specify an EPR, the *@wsdlElement* attribute  
131 must refer to a binding element in the WSDL and not an endpoint or service.

## 132 2.2 Interface mapping

133 When *binding.ws* is used on a service or reference with an interface that is not defined  
134 by *interface.wsdl*, then a WSDL interface for the service or reference is derived from the  
135 interface by the rules defined for that interface type.

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

138 *binding.ws* implementations may use appropriate standards, for example WS-I AP 1.0 or  
139 MTOM, to map interface parameters to binary attachments transparently to the target  
140 component.



141

## 142 **2.3 Production of WSDL description for an SCA service**

143 Any service with one or more web service bindings with HTTP endpoints SHOULD return  
144 a WSDL description of the service in response to an HTTP GET request with the “?wsdl”  
145 suffix to that HTTP endpoint. If none of the web service bindings have HTTP endpoints,  
146 then some other means of obtaining the WSDL description of the service should be  
147 provided. This may include out of band mechanisms, for example publication to a UDDI  
148 registry.

149 Refer to section 4 for a detailed definition of the rules that SHOULD be used for  
150 generating the WSDL description of an SCA service with one or more web service  
151 bindings.

152

## 153 **2.4 Additional binding configuration data**

154 SCA runtime implementations may provide additional metadata that is associated with a  
155 web service binding, for example to enable JAX-WS [JAX-WS] handlers to be executed  
156 as part of the target component dispatch. The specification of such metadata is SCA  
157 runtime-specific and is outside of the scope of this document.

158

## 159 **2.5 Web Service Binding and SOAP Intermediaries**

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

162

## 163 **2.6 Support for WSDL extensibility**

164 When a Web Service binding is specified using the `wsdlElement` attribute, the details of  
165 the binding are specified by the WSDL element referenced by the value of the attribute.  
166 WSDL elements allow for extensibility via elements as well as attribute. The Web Service  
167 binding allows the use of such extensibility in WSDL. Note that as a consequence of this,  
168 when using this form of Web Service binding, it is not possible to determine whether the  
169 binding is supported by the SCA runtime without parsing the referenced WSDL element  
170 and its dependent elements.

## 171 **2.7 Intents listed in the bindingType**

172 This specification places no requirements on the intents that must be listed as either `@alwaysProvides` or  
173 `@mayProvides` in the `bindingType` for `binding.ws`.

## 174 **2.8 Intents and binding configuration**

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

---

## 3 Web Service Binding Examples

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

### 3.1 Example Using WSDL documents

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

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

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

```
<?xml version="1.0" encoding="ASCII"?>
<composite xmlns="http://docs.oasis-open.org/ns/opencsa/sca/200712"
  name="MyValueComposite">
  <service name="MyValueService">
    <interface.java interface="services.myvalue.MyValueService"/>
    <binding.ws wsdlElement="http://www.example.org/MyValueService#
wsdl.endpoint(MyValueService/MyValueServiceSOAP)"/>
    ...
  </service>
  ...
  <reference name="StockQuoteReference1">
    <interface.java interface="services.stockquote.StockQuoteService"/>
    <binding.ws wsdlElement="http://www.example.org/StockQuoteService#
      wsdl.service(StockQuoteService) "
      wsdl:wsdlLocation="http://www.example.org/StockQuoteService
      http://www.example.org/StockQuoteService.wsdl"/>
  </reference>
  <reference name="StockQuoteReference2">
    <interface.java interface="services.stockquote.StockQuoteService"/>
    <binding.ws wsdlElement="http://www.example.org/StockQuoteService#
      wsdl.binding(StockQuoteBinding) "
      wsdl:wsdlLocation="http://www.example.org/StockQuoteService
      http://www.example.org/StockQuoteService.wsdl "
      uri="http://www.example.org/StockQuoteService5"/>
  </reference>
</composite>
```

## 225 3.2 Examples Without a WSDL Document

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

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

234 The reference indicates a service to be invoked which must have a SOAP binding and  
235 portType that matches the default options for binding synthesis and interface mapping.  
236 One particular use of this case would be where the reference is to an SCA service with a  
237 web service binding which itself uses all the defaults.

238

```
239 <?xml version="1.0" encoding="ASCII"?>
240 <composite xmlns="http://docs.oasis-open.org/ns/opencsa/sca/200712"
241           name="MyValueComposite">
242
243     <service name="MyValueService">
244       <interface.java interface="services.myvalue.MyValueService"/>
245       <binding.ws/>
246       ...
247     </service>
248
249     ...
250
251     <reference name="StockQuoteService">
252       <interface.java interface="services.stockquote.StockQuoteService"/>
253       <binding.ws uri="http://www.example.org/StockQuoteService"/>
254     </reference>
255 </composite>
```

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

262

```
263 <?xml version="1.0" encoding="ASCII"?>
264 <composite xmlns="http://docs.oasis-open.org/ns/opencsa/sca/200712"
265           name="MyValueComposite">
266
267     <service name="MyValueService">
268       <interface.java interface="services.myvalue.MyValueService"/>
269       <binding.ws name="MyValueServiceSOAP11" requires="soap.1_1"/>
270       <binding.ws requires="soap.1_2"/>
271       ...
272     </service>
273
274     ...
275
276     <reference name="StockQuoteService">
277       <interface.java interface="services.stockquote.StockQuoteService"/>
278       <binding.ws uri="http://www.example.org/StockQuoteService"
279                 requires="soap.1_2"/>
280     </reference>
```

281 </composite>

282

### 283 3.3 Example PolicySet Providing The Conversation Intent

284 This policy set applies to *binding.ws* and provides the conversation intent. The  
285 conversation intent is provided by using WS-ReliableMessaging protocol which has a  
286 concept of a Sequence. This Sequence (which appears as a `wsm:Sequence` SOAP  
287 header in the message) is used as a correlation mechanism, on the wire, to implement  
288 conversational semantics.

```
289 <policySet name="WSRM-Sequence-based-conversation"  
290           provides="sca:conversation"  
291           appliesTo="sca:binding.ws">  
292   <wsp:Policy>  
293     <wsrmp:RMAssertion  
294       xmlns:wsrmp="http://docs.oasis-open.org/ws-rx/wsrmp/200608"/>  
295   </wsp:Policy>  
296 </policySet>
```

297

---

## 298 4 WSDL Generation

299 This section defines the rules that SHOULD be used for generation of a WSDL document  
300 that describes an SCA service with one or more web service bindings that require a  
301 SOAP binding.

302 A WSDL document may be generated for an SCA service with non-SOAP web service  
303 bindings, or other bindings. For non-SOAP web service bindings that do not refer to an  
304 existing WSDL document, or non-web service bindings, the generation rules below may  
305 be considered a template, and a similar approach taken.

306

### 307 4.1 Intents

308 The following intents affect WSDL generation:

- 309 • soap  
310 This indicates that a SOAP binding is required. The SOAP binding may be of any  
311 SOAP version, including multiple versions.
- 312 • soap.1\_1  
313 A SOAP 1.1 binding only is required.
- 314 • soap.1\_2  
315 A SOAP 1.2 binding only is required.

316

### 317 4.2 WSDL Service and Ports

318 A separate WSDL document is generated for each SCA service. Each has its own unique  
319 target namespace. This is to ensure that bindings on different services of the same  
320 component do not clash. The WSDL service has one or more ports for each web service  
321 binding on the SCA service that has a SOAP requirement, or that refers to an existing  
322 WSDL binding, depending on the requirements of the web service binding. Each of  
323 those ports has a single binding.

324 Additional ports and bindings may be generated in this WSDL document for non-web  
325 service bindings, or web service bindings with non-SOAP requirements. The manner in  
326 which that is done is undefined.

327 The binding elements themselves may be generated as defined below, or may be  
328 imported from existing WSDL documents in the case that the web service binding refers  
329 to the binding element of such a document.

330 The target namespace of the WSDL document, and of the service, ports and generated  
331 binding elements is:

332 Base System URI for HTTP / Component Name / Service Name

333

### 334 4.3 WSDL Bindings

335 The binding elements in the generated WSDL document are either defined within the  
336 document, derived from the requirements of the binding, or are imported from existing  
337 WSDL documents.

- 338 Generated bindings have the following fixed assumptions:
- 339 • use="literal" for input and output messages
  - 340 • style="document" for the binding
  - 341 • All faults map to soap: faults
  - 342 • No header or headerFault elements are generated
  - 343 • The transport is "http://schemas.xmlsoap.org/soap/http", unless the system
  - 344 provides intents for alternative transports
  - 345 • The soap version is determined from the soap intents as defined above

346

### 347 4.3.1 SOAP versions

348 Where a web service binding requires a specific SOAP version, then a single WSDL port  
349 and SOAP binding of the appropriate version is generated.

350 Where no specific SOAP version is required, then one or more WSDL ports with  
351 associated SOAP bindings may be generated, depending on the level(s) supported in the  
352 target runtime.

353

### 354 4.4 WSDL portType

355 An SCA service has a single interface. This interface is always imported into the  
356 generated WSDL document. This may be done directly for WSDL-defined interfaces, or  
357 indirectly via a WSDL generated from the interface type for the service.

358

### 359 4.5 WSDL Generation Rules

360 The following is the formal definition of the generation of a WSDL document from an  
361 SCA service with one or more web service bindings, with either a SOAP requirement or  
362 existing WSDL document:

```
363 <?xml version="1.0" encoding="UTF-8"?>
364 <definitions name="componentName.serviceName"
365             targetNamespace="HTTP Base URI/componentName/serviceName"
366             {(if any bindings require SOAP 1.1)
367             xmlns:soap11="http://schemas.xmlsoap.org/wsdl/soap/"
368             }
369             {(if any bindings require SOAP 1.2)
370             [xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"]
371             }
372             xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
373             xmlns="http://schemas.xmlsoap.org/wsdl/">
374
375     <import namespace="SCA service interface namespace"
376            location="SCA service interface location"/>
377
378     {(for each binding.ws element in the service with a WSDL, do the following:)
379     <import namespace="existing WSDL binding namespace"
380            location="existing WSDL binding location"/>
381     }
382
```

```

383     {(for each binding.ws element in the service without a WSDL, do the
384 following
385     for each SOAP version required:)
386     <binding name="/service/binding.ws[n]/@name+[.soapVersionPrefix]+'Binding' "
387         type="SCA service interface portType name">
388         <soapVersionPrefix:binding
389 transport="http://schemas.xmlsoap.org/soap/http"/>
390         {(for each operation in the interface do the following:)
391         <operation name="name-of-the-operation">
392             <soapVersionPrefix:operation/>
393             <input>
394                 <soapVersionPrefix:body use="literal"/>
395             </input>
396             {(if there is an output)
397             <output>
398                 <soapVersionPrefix:body use="literal"/>
399             </output>
400             }
401             {(if there is a fault)
402             <fault>
403                 <soapVersionPrefix:fault name="name-of-the-fault"/>
404             </fault>
405             }
406         </operation>
407         }
408     </binding>
409     }
410
411     <service name="/service/@name">
412         {(for each binding.ws element in the service do the following for each
413 SOAP
414         version required:)
415         <port name="/service/binding.ws[n]/@name+[.soapVersionPrefix]+'Port' "
416
417 binding="/service/binding.ws[n]/@name+[.soapVersionPrefix]+'Binding' ">
418             <soapVersionPrefix:address location="/service/binding.ws[n]/@uri"/>
419             </port>
420         }
421     </service>
422 </definitions>

```

---

423 **5 Conformance**

424 Any SCA runtime that claims to support this binding must abide by the requirements of this specification.

425 The XML schema available at the namespace URI, defined by this specification, is considered to be  
426 authoritative and takes precedence over the XML Schema defined in the appendix of this document.



## A. Web Services Binding Schema

```
428 <?xml version="1.0" encoding="UTF-8"?>
429 <!-- (c) Copyright OASIS 2006, 2008 -->
430 <schema xmlns="http://www.w3.org/2001/XMLSchema"
431         targetNamespace="http://docs.oasis-open.org/ns/opencsa/sca/200712"
432         xmlns:sca="http://docs.oasis-open.org/ns/opencsa/sca/200712"
433         xmlns:wsdli="http://www.w3.org/ns/wsdl-instance"
434         xmlns:wsa="http://www.w3.org/2005/08/addressing"
435         elementFormDefault="qualified">
436
437     <import namespace="http://www.w3.org/ns/wsdl-instance"
438             schemaLocation="http://www.w3.org/2007/05/wsdl/wsdl20-
439 instance.xsd"
440     />
441     <import namespace="http://www.w3.org/2005/08/addressing"
442             schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd"
443     />
444     <include schemaLocation="sca-core.xsd"/>
445
446     <element name="binding.ws" type="sca:WebServiceBinding"
447             substitutionGroup="sca:binding"/>
448     <complexType name="WebServiceBinding">
449         <complexContent>
450             <extension base="sca:Binding">
451                 <sequence>
452                     <element ref="wsa:EndpointReference" minOccurs="0"
453                             maxOccurs="unbounded"/>
454                     <any namespace="##other" processContents="lax"
455 minOccurs="0"
456                             maxOccurs="unbounded"/>
457                 </sequence>
458                 <attribute name="wsdlElement" type="anyURI" use="optional"/>
459                 <attribute ref="wsdli:wsdlLocation" use="optional"/>
460                 <anyAttribute namespace="##any" processContents="lax"/>
461             </extension>
462         </complexContent>
463     </complexType>
464 </schema>
```

465

---

466 **B. Acknowledgements**

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

469 **Participants:**

470 [Participant Name, Affiliation | Individual Member]

471 [Participant Name, Affiliation | Individual Member]

472



474

## D. Revision History

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

476

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"> <li>* Partially applied the resolution of issue 14 in the conformance section.</li> <li>* Applied resolution to issue 9.</li> <li>* Applied resolution to issue 15.</li> <li>* Applied resolution to issue 16.</li> <li>* Applied resolution to issue 10.</li> <li>* Applied resolution to issue 8.</li> <li>* Applied resolution to issue 3.</li> </ul>
3	2008-06-12	Simon Holdsworth	<ul style="list-style-type: none"> <li>* Completed application of resolution to issue 10</li> <li>* Applied most of the editorial changes from Eric Johnson's review</li> </ul>
4	2008-08-13	Anish Karmarkar	<ul style="list-style-type: none"> <li>* Applied rest of Eric Johnson's ed review comments.</li> <li>* Applied resolution of issue 13.</li> <li>* Reapplied resolution of issue 15 (it was not applied correctly before)</li> <li>* Applied resolution of issue 19.</li> <li>* Applied resolution of issue 30.</li> <li>* Applied resolution of issue 32.</li> <li>* Applied resolution of issue 36.</li> <li>* Applied resolution of issue 38.</li> </ul>

477

478