Service Component Architecture HTTP Binding Specification Version 1.1

Working Draft 01

22nd October 2008

Specification URIs:
This Version:
http://docs.oasis-open.org/sca-bindings/sca-binding-http-1.1-spec-wd01.html
http://docs.oasis-open.org/sca-bindings/sca-binding-http-1.1-spec-wd01.doc
http://docs.oasis-open.org/sca-bindings/sca-binding-http-1.1-spec-wd01.pdf (Authoritative)

Previous Version:

Latest Version:
http://docs.oasis-open.org/sca-bindings/sca-binding-http-1.1-spec.html
http://docs.oasis-open.org/sca-bindings/sca-binding-http-1.1-spec.doc
http://docs.oasis-open.org/sca-bindings/sca-binding-http-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 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:
This document defines the concept and behavior of an HTTP-based binding.

The binding specified in this document applies to an SCA composite's services and references. The binding is especially well suited for use by services and references of composites that are directly deployed, as opposed to composites that are used as implementations of higher-level components. Services and references of deployed composites become system-level services and references, which are intended to be used by non-SCA clients.

The HTTP binding provides HTTP-specific details of the connection to the required HTTP endpoint and controls formation of HTTP protocol messages. It supports the use of HTTP(s) type destinations.

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

Table of Contents

1 Introduction

This document defines the concept and behavior of the HTTP [HTTP] binding.

The binding specified in this document applies to an SCA composite's services and references. The binding is especially well suited for use by services and references of composites that are directly deployed, as opposed to composites that are used as implementations of higher-level components. Services and references of deployed composites become system-level services and references, which are intended to be used by non-SCA clients.

The HTTP binding provides HTTP-specific details of the connection to the required HTTP endpoint and controls formation of HTTP protocol messages. It supports the use of HTTP(s) type destinations.

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

Prefix	***Namespace***	***Notes***
xs	"http://www.w3.org/2001/XMLSchema"	Defined by XML Schema 1.0 specification
sca	"http://docs.oasis-open.org/ns/opencsa/sca/200712"	Defined by the SCA specifications

1.2 Normative References

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

[HTTP] HTTP specification (1.0, 1.1?) TBC http://www.w3.org/Protocols/rfc2616/rfc2616.html

[WSDL] E. Christensen et al, *Web Service Description Language (WSDL) 1.1*, http://www.w3.org/TR/2001/NOTE-wsdl-20010315, W3C Note, March 15 2001.

R. Chinnici et al, *Web Service Description Language (WSDL) Version 2.0 Part 1: Core Language*, http://www.w3.org/TR/2007/REC-wsdl20-20070626/, W3C Recommendation, June 26 2007.

1.3 Non-Normative References

TBD TBD

2 HTTP Binding Schema

The HTTP binding element is defined by the following schema.

```
<binding.http uri="xs:anyURI"?
              readTimeout="int"?
              numberOfRetries="int"?
              methods="list of string"?
              ...>

    <wireFormat/>?

    <operationSelector/>?

    <response>
        <wireFormat>?
    </response>?

    <pingableMethod name="string" statusCode="int">*

    <proxySettings host="string"?
                   port="int"?
                   type="http or https"?
                   nonProxyHost="string"?>
        <credentials alias="string"/>?
    </proxySettings>*

    <sslSettings alias="string"/>?

    <authentication type="basic or digest or ..."?>
        <credentials alias="string"/>
    </authentication>?

    <headers HTTPVersion="1.0 or 1.1"?
             HTTPContentEncoding="string"?
             HTTPTransferEncoding="string"?
             HTTPMediaType="string"?
             HTTPCharset="string"?
             HTTPMethod="string"?
             ...>
        <property name="NMTOKEN" type="NMTOKEN">*
    </headers>?

    <operationProperties name="string"
                         selectedOperation="string"?
                         methods="list of string"?
                         contextPath="string"?>
        <property name="NMTOKEN" type="NMTOKEN">*
        <headers HTTPVersion="1.0 or 1.1"?
                 HTTPContentEncoding="string"?
                 HTTPTransferEncoding="string"?
                 HTTPMediaType="string"?
                 HTTPCharset="string"?
                 HTTPMethod="string"?
                 ...>
            <property name="NMTOKEN" type="NMTOKEN">*
        </headers>?
    </operationProperties>*
</binding.http>
```

Comment [sajh1]: Given that many of the elements/attributes are specific to use for a service or reference, it may make sense to split this into two elements, although there's no precedent for that in other bindings.

The ***binding.http*** element has the following attributes:

- ***/binding.http*** – This is the generic HTTP binding type. The type is extensible so that HTTP binding implementers can add additional attributes and elements although such extensions are not guaranteed to be portable across runtimes.
- ***/binding.http/@uri*** – (from binding) for a reference, identifies the URL of the target service. For a service, identifies the base context path.
- ***/binding.http/@readTimeout*** – The amount of time that should be allowed for a response to be received after sending a request.
- ***/binding.http/@numberOfRetries*** – the number of times the request should be retried.
- ***/binding.http/@methods*** – for a service, the list of methods supported for all operations, unless overridden for a specific operation.
- ***/binding.http/wireFormat*** - identifies the wire format used by requests and responses sent or received by this binding.
- ***/binding.http/operationSelector*** – identifies the operation selector used when receiving requests for a service. If specified for a reference this provides the default operation selector for callbacks if not specified via a callback service element.
- ***/binding.http/response*** – defines the resources used for handling response messages (receiving responses for a reference, and sending responses from a service).
- ***/binding.http/response/wireFormat*** – identifies the wire format used by responses sent or received by this binding. This value overrides the ***wireFormat*** specified at the binding level.
- ***/binding.http/pingableMethod*** – for a service, identifies a method that is pingable.
- ***/binding.http/pingableMethod/@name*** – the name of the pingable method.
- ***/binding.http/pingableMethod/@statusCode*** – the HTTP status code that is returned.
- ***/binding.http/proxySettings*** – for a reference, identifies the settings of an HTTP proxy to be used.
- ***/binding.http/proxySettings/@host*** – the host name of the proxy.
- ***/binding.http/proxySettings/@port*** – the port number of the proxy.
- ***/binding.http/proxySettings/@type*** – the type of the proxy, http or https.
- ***/binding.http/proxySettings/@nonProxyHost*** – Host name for which the proxy should not be used.
- ***/binding.http/proxySettings/credentials*** – Credentials to be used with the proxy.
- ***/binding.http/proxySettings/credentials/@alias*** – The alias of the credential settings to be used.
- ***/binding.http/sslSettings*** – for a reference, identifies the SSL settings to use to connect to the target service.
- ***/binding.http/sslSettings/@alias*** – The alias of the SSL settings to be used.
- ***/binding.http/authentication*** – for a reference, identifies the authentication to use to connect to the target service.
- ***/binding.http/authentication/@type*** – The type of authentication to use.
- ***/binding.http/authentication/credentials*** – Credentials to be used to authenticate to the target service.
- ***/binding.http/authentication/credentials/@alias*** – The alias of the credential settings to be used.
- ***/binding.http/headers*** – this element allows HTTP headers to be set to the given values for all operations. These values apply to requests from a reference and responses from a service.
- ***/binding.http/headers/@HTTPVersion, @HTTPMethod, @HTTPContentEncoding, @HTTPTransferEncoding, @HTTPMediaType, @HTTPCharSet*** – specifies the value to use for the given HTTP header property. ***@HTTPVersion*** and ***@HTTPMethod*** only apply to requests sent from references.
- ***/binding.http/headers/property*** – specifies the value to use for the named HTTP header.

Comment [sajh2]: Do we want to say anything at all about the format of the URI wrt. parameterization?

Comment [SAJH3]: This is somewhat JEE-centric; perhaps these should be removed and replaced by intents?

- ***/binding.http/operationProperties*** – specifies various properties that are specific to the processing of a particular operation.
- ***/binding.http/operationProperties/@name*** – The name of the operation in the interface.
- ***/binding.http/operationProperties/@selectedOperation*** – The value generated by the ***operationSelector*** that corresponds to the operation in the service or reference interface identified by the ***operationProperties/@name*** attribute. If this attribute is omitted then the value defaults to the value of the ***operationProperties/@name*** attribute. The value of this attribute MUST be unique across the containing ***binding.http*** element.
- ***/binding.http/operationProperties/@methods*** – for a service, the list of methods supported for this operation. Overrides the value specified for the binding.
- ***/binding.http/operationProperties/@contextPath*** – for a service, context path for this specific operation.
- ***/binding.http/operationProperties/property*** – specifies properties specific to this operation. These properties are intended to be used to parameterize the ***wireFormat*** identified for the binding for a particular operation. The SCA runtime SHOULD make the ***operationProperties*** element corresponding to the ***selectedOperation*** available to the ***wireFormat*** implementation.
- ***/binding.http/operationProperties/headers*** – this element allows HTTP headers to be set to the given values for all operations. These values apply to requests from a reference and responses from a service.
- ***/binding.http/operationProperties/headers/@HTTPVersion, @HTTPMethod, @HTTPContentEncoding, @HTTPTransferEncoding, @HTTPMediaType, @HTTPCharSet*** – specifies the value to use for the given HTTP header property. ***@HTTPVersion*** and ***@HTTPMethod*** only apply to requests sent from references.
- ***/binding.http/operationProperties/headers/property*** – specifies the value to use for the named HTTP header.
- ***/binding.http/@{any}*** - this is an extensibility mechanism to allow extensibility via attributes.
- ***/binding.http/any*** – this is an extensibility mechanism to allow extensibility via elements.

2.1 Intents listed in the bindingType

This specification places no requirements on the intents that must be listed as either ***@alwaysProvides*** or ***@mayProvides*** in the ***bindingType*** for ***binding.http***.

2.2 Intents and binding configuration

The SCA runtime MUST raise an error if the HTTP binding is configured with a policy intent(s) that conflicts with a binding instance's configuration.

3 Default Operation Selection and Data Binding behavior

In general HTTP-based applications deal with URL paths and HTTP methods. There is not always a direct relationship between these and an operation defined in a WSDL portType **[WSDL]**. Messages have a content type which corresponds in some way to the schema of an input or output message of an operation in the interface of a service or reference, however additional information is required in order for an SCA runtime to know how to identify the operation and understand the format of messages.

The process of identifying the operation to be invoked is operation selection; the information that describes the contents of messages is a wire format. The binding element as described in the SCA Assembly specification [ref needed] provides the means to identify specific operation selection via the ***operationSelector*** element and the format of messages received and to be sent using the ***wireFormat*** element. The HTTP binding defines a default operation selector and wire format and the corresponding ***operationSelector*** and ***wireFormat*** elements; SCA providers may provide additional such elements.

No standard means is provided for linking the ***wireFormat*** or ***operationSelector*** elements with the runtime components that implement their behavior.

This section describes the default operation selector and wire format for an HTTP binding.

3.1 Default Operation Selector

When receiving a request at a service the selected operation name is determined as follows:

- If there is only one operation on the service's interface, then that operation is assumed as the selected operation name.
- Otherwise, if the HTTP header "***scaOperationName***" is present, then its value is used as the selected operation name.
- Otherwise, a combination of the URL's context path and HTTP method is used as the selected operation name, i.e. HTTP Method + "@" + URL context path.

The selected operation name is then be mapped to an operation in the service's interface via a matching ***operationProperties*** element in the HTTP binding. If there is no matching element, the operation name is assumed to be the same as the selected operation name.

The use of this operation selector can be explicitly specified in a ***binding.http*** using the ***<operationSelector.httpdefault>*** element; if no ***operationSelector*** element is specified then SCA runtime MUST use this as the default.

An SCA runtime **SHOULD** provide the means for identifying alternative operation selectors via additional ***<operationSelector>*** elements.

3.2 Default Wire Format

Comment [sajh4]: Should we support MIME messages in the default wire format?

Also can the default wire format make use of additional content types?

Should we say anything about URL parameterization here?

The default wire format behavior maps between a message on the wire and the object(s) expected by the component implementation.

The message body is mapped to the parameters or return value of the target operation as follows:

- If the operation has a single parameter or return value that is a byte array, then the whole message is passed as is, and MUST have a content type of "application/octet-stream".
- Otherwise, the message content type MUST be "text/xml".
- If there is a single parameter, or for the return value, the text XML payload is the XML serialization of that parameter according to the WSDL schema for the message.
- If there are multiple parameters, then they are encoded in XML using the document wrapped style, according to the WSDL schema for the message.

- When sending request messages, if there is a single parameter and the interface includes more than one operation, the SCA runtime MUST set the HTTP header "***scaOperationName***" to the name of the operation being invoked.

The use of this wire format can be explicitly specified in a ***binding.http*** using the ***<wireFormat.httpdefault>*** element; if no ***wireFormat*** element is specified then SCA runtimes MUST use this as the default.

An SCA runtime SHOULD provide the means for identifying alternative wire formats to support any other type of HTTP message via additional ***<wireFormat>*** elements.

4 Examples

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

4.1 URI Binding Example

The following example shows the HTTP binding using the URI attribute for a service to specify the context path, and for a reference to specify the location of the target service:

```
<?xml version="1.0" encoding="ASCII"?>
<composite xmlns="http://www.osoa.org/xmlns/sca/1.0"
           name="MyValueComposite">

    <service name="MyValueService">
        <interface.java interface="services.myvalue.MyValueService"/>
        <binding.http uri="/myValueService"/>
    </service>

    <reference name="StockQuoteService">
        <interface.java interface="services.stockquote.StockQuoteService"/>
        <binding.http uri="http://www.example.org/StockQuoteService"/>
    </reference>
</composite>
```

TBD: Need more examples of using various binding elements/attributes

5 Conformance

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

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

Within this specification, the following conformance targets are used:

- XML document elements and attributes, including binding.http and its children, and bindingType
- The SCA runtime – this refers to the implementation that provides the functionality to support the SCA specifications, including that specific to the HTTP binding as well as other SCA capabilities
- WSDL documents

A. HTTP Binding Schema

Note: needs to be validated.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- (c) Copyright OASIS 2008 -->
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        targetNamespace="http://docs.oasis-open.org/ns/opencsa/sca/200712"
        xmlns:sca="http://docs.oasis-open.org/ns/opencsa/sca/200712"
        elementFormDefault="qualified">

   <include schemaLocation="sca-core.xsd"/>

   <complexType name="HTTPBinding">
      <complexContent>
         <extension base="sca:Binding">
            <sequence>
                <element name="response" type="sca:HTTPResponse" minOccurs="0"/>
                <element name="pingableMethod" type="sca:HTTPPingableMethod"
                         minOccurs="0"/>
                <element name="proxySettings" type="sca:HTTPProxySettings"
                         minOccurs="0"/>
                <element name="sslSettings" type="sca:HTTPSSLSettings"
                         minOccurs="0"/>
                <element name="authentication" type="sca:HTTPAuthentication"
                         minOccurs="0"/>
                <element name="headers" type="sca:HTTPHeaders" minOccurs="0"/>
                <element name="operationProperties"
                         type="sca:HTTPOperationProperties"
                         minOccurs="0" maxOccurs="unbounded"/>
                <any namespace="##other" processContents="lax"
                     minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
            <attribute name="readTimeout" type="int"/>
            <attribute name="numberOfRetries" type="int"/>
            <attribute name="methods" type="string"/>
            <anyAttribute/>
         </extension>
      </complexContent>
   </complexType>

   <complexType name="HTTPResponse">
      <sequence>
         <element name="wireFormat" type="sca:WireFormat" minOccurs="0"/>
      </sequence>
   </complexType>

   <complexType name="HTTPProxySettings">
      <sequence>
         <element name="credentials" minOccurs="0">
            <attribute name="alias" type="string"/>
         </element>
         <attribute name="host" type="string"/>
         <attribute name="port" type="int"/>
         <attribute name="type">
            <simpleType>
               <restriction base="string">
                  <enumeration value="http"/>
                  <enumeration value="https"/>
               </restriction>
            </simpleType>
         </attribute>
```

```
         <attribute name="nonProxyHost" type="string"/>
      </sequence>
   </complexType>

   <complexType name="HTTPSSLSettings">
      <attribute name="alias" type=”string”/>
   </complexType>

   <complexType name="HTTPAuthentication">
      <sequence>
         <element name="credentials" minOccurs="0">
            <attribute name="alias" type="string"/>
         </element>
         <attribute name="type" type=”string”/>
      </sequence>
   </complexType>

   <simpleType name="HTTPVersion">
      <restriction base="string">
         <enumeration value="1.1"/>
         <enumeration value="1.0"/>
      </restriction>
   </simpleType>

   <complexType name="HTTPHeaders">
      <sequence>
         <element name="property" type="string"
                  minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="HTTPVersion" type="HTTPVersion" default="1.1"/>
      <attribute name="HTTPContentEncoding" type="string"/>
      <attribute name="HTTPTransferEncoding" type="string"/>
      <attribute name="HTTPMediaType" type="int"/>
      <attribute name="HTTPCharset" type="string"/>
      <attribute name="HTTPMethod" type="string"/>
   </complexType>

   <complexType name="HTTPOperationProperties">
      <sequence>
         <element name="property" type="string"
                  minOccurs="0" maxOccurs="unbounded"/>
         <element name="headers" type="sca:HTTPHeaders"/>
      </sequence>
      <attribute name="name" type="string" use="required"/>
      <attribute name="selectedOperation" type="string"/>
   </complexType>

   <element name="binding.http" type="sca:HTTPBinding"
            substitutionGroup="sca:binding"/>
</schema>
```

B. Acknowledgements

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

Participants:

[Participant Name, Affiliation | Individual Member]

[Participant Name, Affiliation | Individual Member]

C. Non-Normative Text

D. Revision History

[optional; should not be included in OASIS Standards]

Revision	Date	Editor	Changes Made
wd01	2008-10-22	Simon Holdsworth	Initial draft