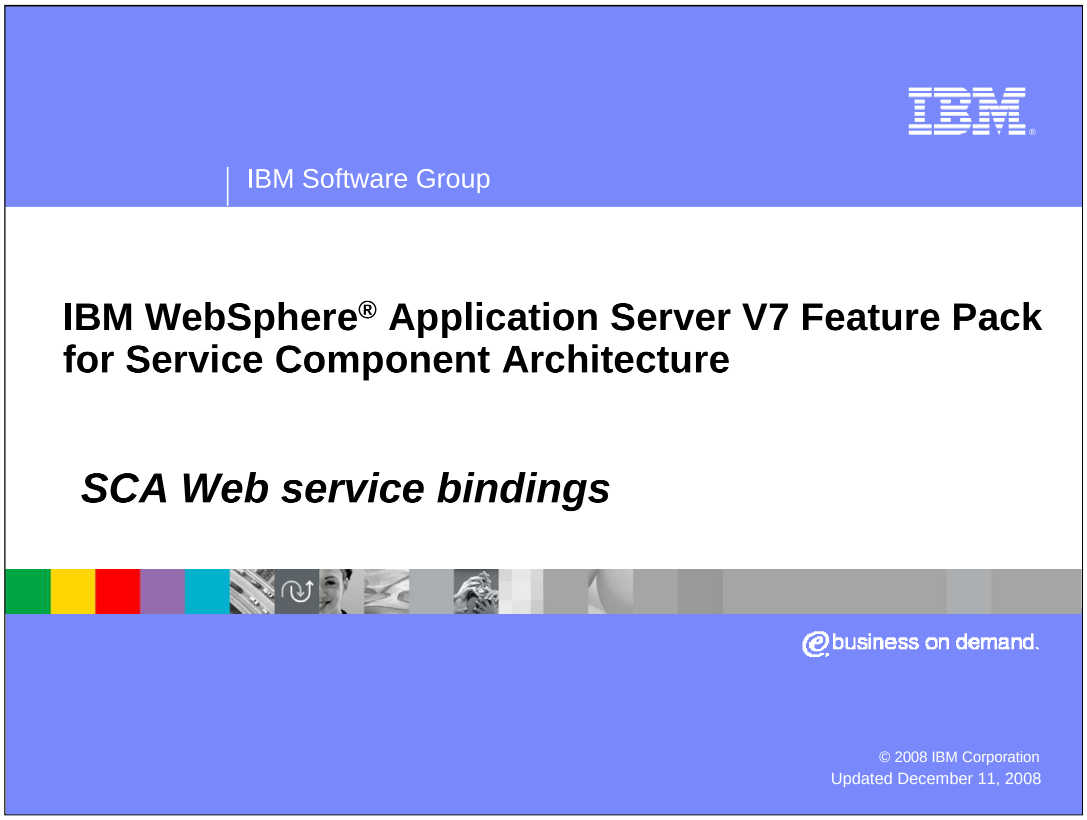


IBM Software Group

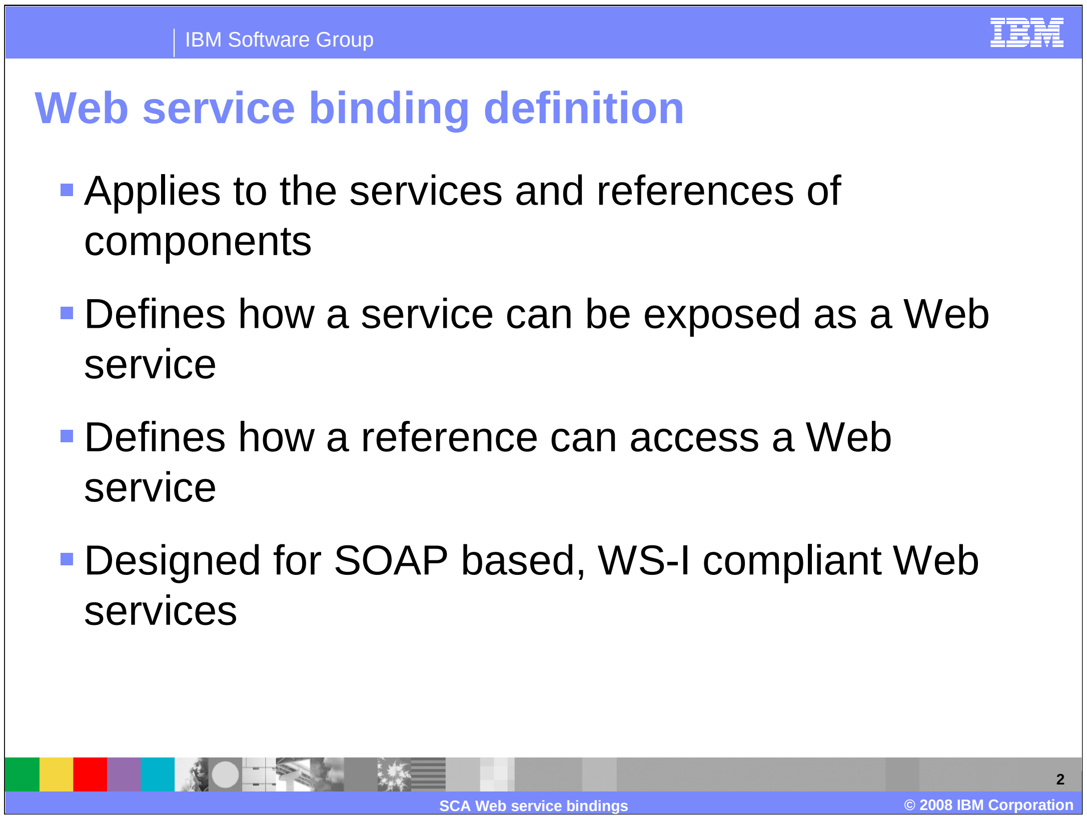
IBM WebSphere® Application Server V7 Feature Pack for Service Component Architecture

SCA Web service bindings

business on demand.

Updated December 11, 2008

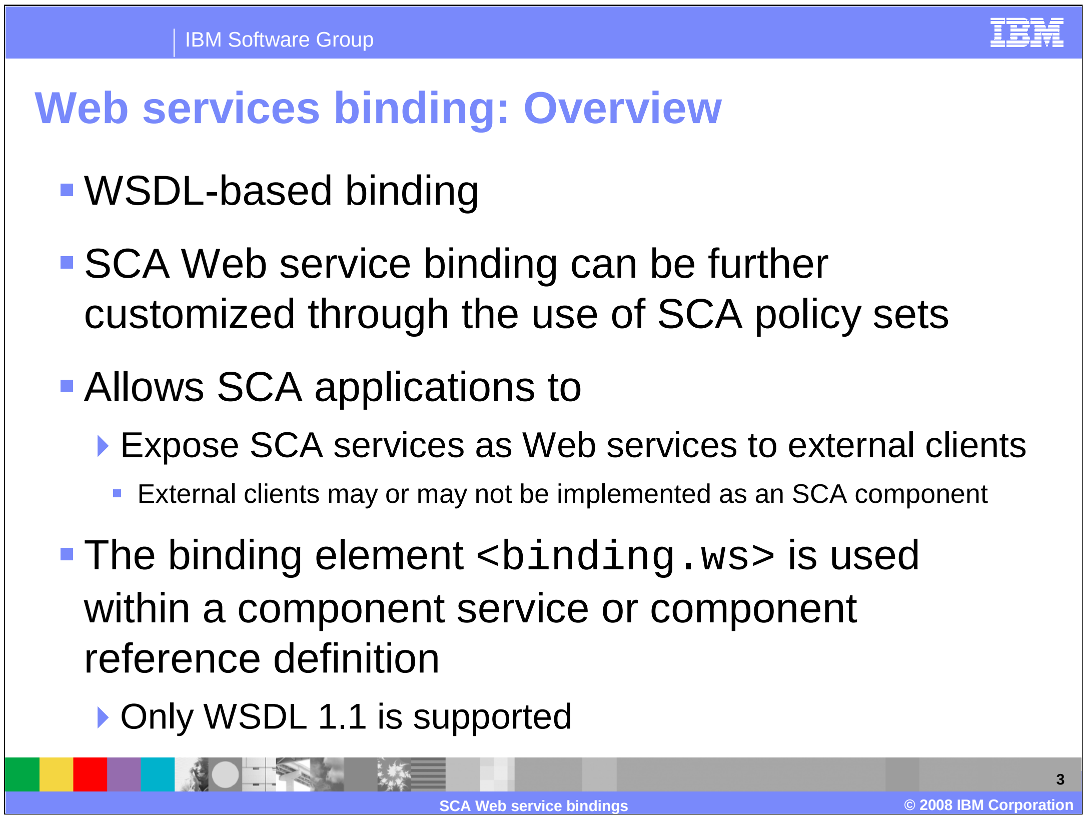
This presentation will cover the SCA Web service binding.



Web service binding definition

- Applies to the services and references of components
- Defines how a service can be exposed as a Web service
- Defines how a reference can access a Web service
- Designed for SOAP based, WS-I compliant Web services

SCA Web service binding 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 applies to the services and references of components. It is also designed for SOAP based, WS-I (interoperable) compliant Web services.


IBM Software Group

Web services binding: Overview

- WSDL-based binding
- SCA Web service binding can be further customized through the use of SCA policy sets
- Allows SCA applications to
  - Expose SCA services as Web services to external clients
    - External clients may or may not be implemented as an SCA component
- The binding element `<binding.ws>` is used within a component service or component reference definition
  - Only WSDL 1.1 is supported

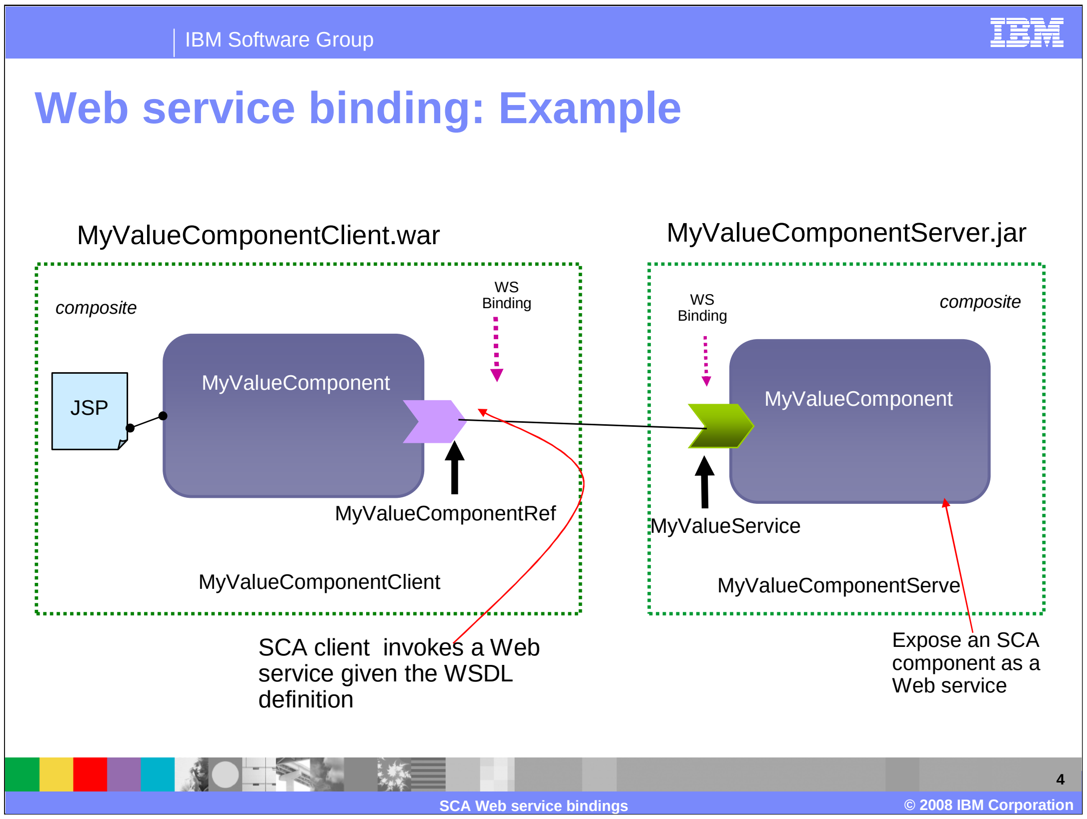
3

SCA Web service bindings © 2008 IBM Corporation

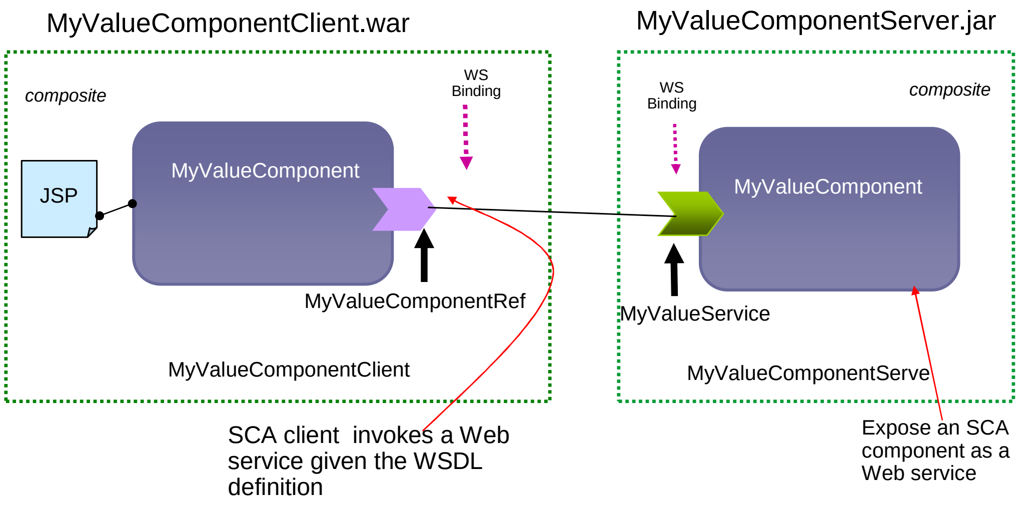
Web services binding is a WSDL-based binding; that means that it either references an existing WSDL binding or allows one to specify enough information to generate one. When an existing WSDL binding is not referenced, WSDL binding can be generated. SCA Web service binding can be further customized through the use of SCA Policy Sets.

Web services technology plays an important role in most SOA solutions relevant today, including SCA. The Web service binding type allows SCA applications to expose services as Web services to external clients in addition to allowing SCA components access to external Web services. External clients that access SCA services exposed as Web services may or may not be implemented as an SCA component. The Web service binding element <**binding.ws**> can be used within either a component service or a component reference definition. When used with a component service, this binding type allows clients to access a service offered by a particular component as a Web service. In the case where the Web service binding is used with a component reference, components in an SCA component can consume an external Web service and access it like it was any other SCA component.

Note that SCA feature pack supports SOAP 1.1 and 1.2 and WSDL 1.1.



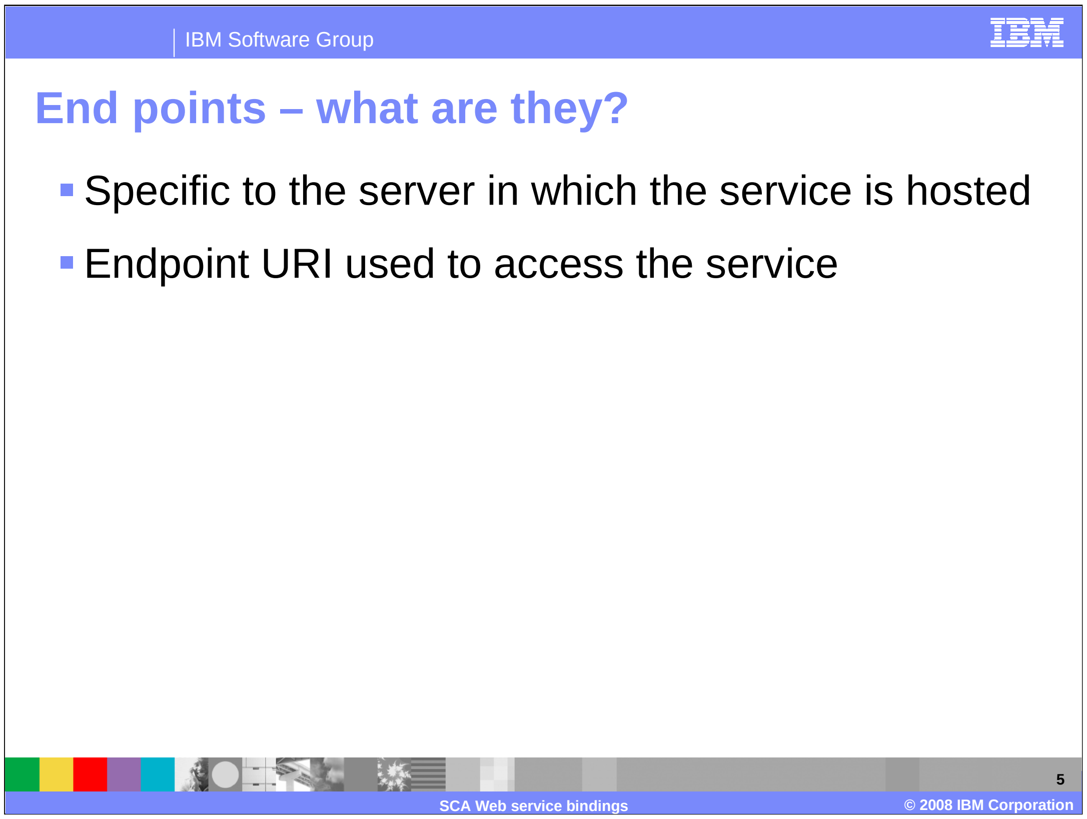
Web service binding: Example


MyValueComponentClient.war
composite
WS Binding
JSP
MyValueComponent
MyValueComponentRef
MyValueComponentClient
SCA client invokes a Web service given the WSDL definition
MyValueComponentServer.jar
WS Binding
composite
MyValueComponent
MyValueService
MyValueComponentServe
Expose an SCA component as a Web service

Here is an example of a Web service binding at work.

This graphic shows two components, one with MyValuecomponent reference and the other with MyValue Service both using a Web service binding to allow communication between the two components.

Note that in typical scenario where a client and a service are both SCA, SCA (default) binding would be preferable over Web service binding.



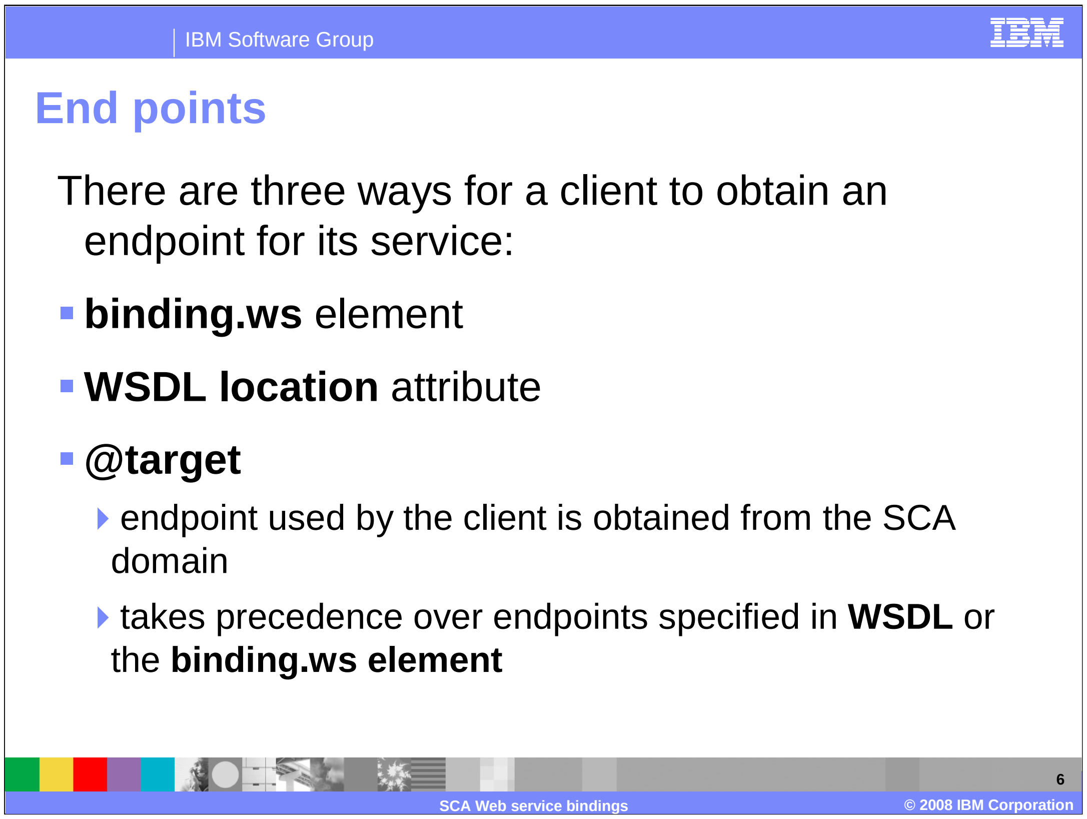
End points – what are they?

- Specific to the server in which the service is hosted
- Endpoint URI used to access the service

When a service is exposed over the SCA Web service binding, the service endpoint is specific to the server in which the service is hosted. Clients use this endpoint URI to access the service. In some cases, you may want clients to indirectly reference the service by using a proxy server as the service endpoint. For example, a proxy server is required to implement clustered Web service binding endpoints. To enable clients to use a proxied endpoints, there are two ways to do this:

If your endpoints are specified in the SCA contributions SCDL or WSDL document location attribute, you must specify the proxy server endpoint instead of the WebSphere server specific endpoint.

If your client resolves the endpoint by using the <reference target=""> attribute in your client SCDL document, use the administrative console to configure the custom endpoints for SCA composites that are accessed by the HTTP protocol. This approach is the most flexible for SCA clients within the same domain as their service providers. When using <reference target=""> attribute, SCA references can resolve the service endpoints without the client specifying endpoints in the SCDL or WSDL document.

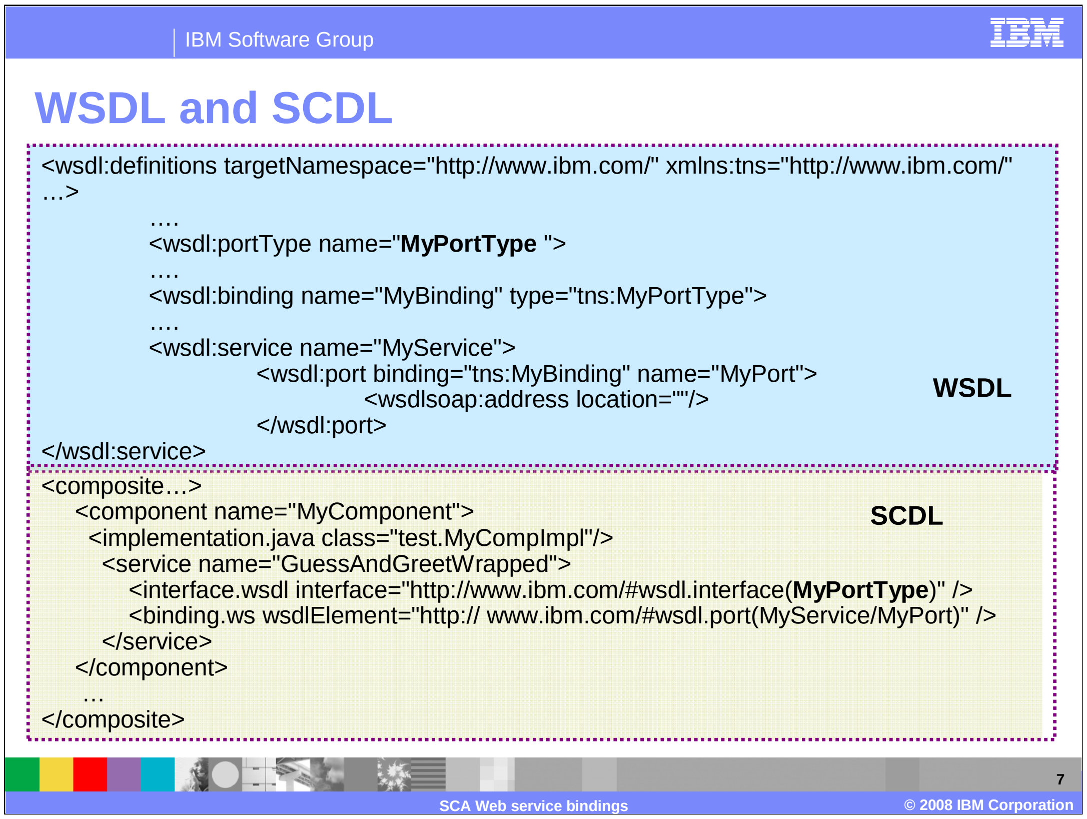


End points

There are three ways for a client to obtain an endpoint for its service:

- **binding.ws** element
- **WSDL location** attribute
- **@target**
  - endpoint used by the client is obtained from the SCA domain
  - takes precedence over endpoints specified in **WSDL** or the **binding.ws element**

There are three ways for a client to obtain an endpoint for its service: through **binding.ws** element, through WSDL location attribute, and through target annotation. When target annotation (@target) is used, the endpoint used by the client is obtained from the SCA domain. It takes precedence over endpoints specified in WSDL or the binding.ws element. Therefore when using @target annotation you do not need to specify any reference endpoints.


IBM Software Group

WSDL and SCDL

WSDL

```
<wsdl:definitions targetNamespace="http://www.ibm.com/" xmlns:tns="http://www.ibm.com/"
…>
        ….
        <wsdl:portType name="MyPortType ">
        ….
        <wsdl:binding name="MyBinding" type="tns:MyPortType">
        ….
        <wsdl:service name="MyService">
                <wsdl:port binding="tns:MyBinding" name="MyPort">
                        <wsdlsoap:address location=""/>
                </wsdl:port>
</wsdl:service>
```

SCDL

```
<composite…>
   <component name="MyComponent">
    <implementation.java class="test.MyCompImpl"/>
     <service name="GuessAndGreetWrapped">
       <interface.wsdl interface="http://www.ibm.com/#wsdl.interface(MyPortType)" />
       <binding.ws wsdlElement="http:// www.ibm.com/#wsdl.port(MyService/MyPort)" />
     </service>
   </component>
    …
</composite>
```

7

SCA Web service bindings © 2008 IBM Corporation

The following example shows the relationship between the WSDL (Web Services Description Language) file and the Service Component Definition Language (SCDL). Note the reference to the WSDL portType name in both files.

The SCA Web service binding specification defines several options of specifying a WSDL element in the SCDL. Depending on which form you chose, a different element within the WSDL is used. It also describes deploying SCA services and references without a WSDL document, this means without a wsdl element.

The different options are:

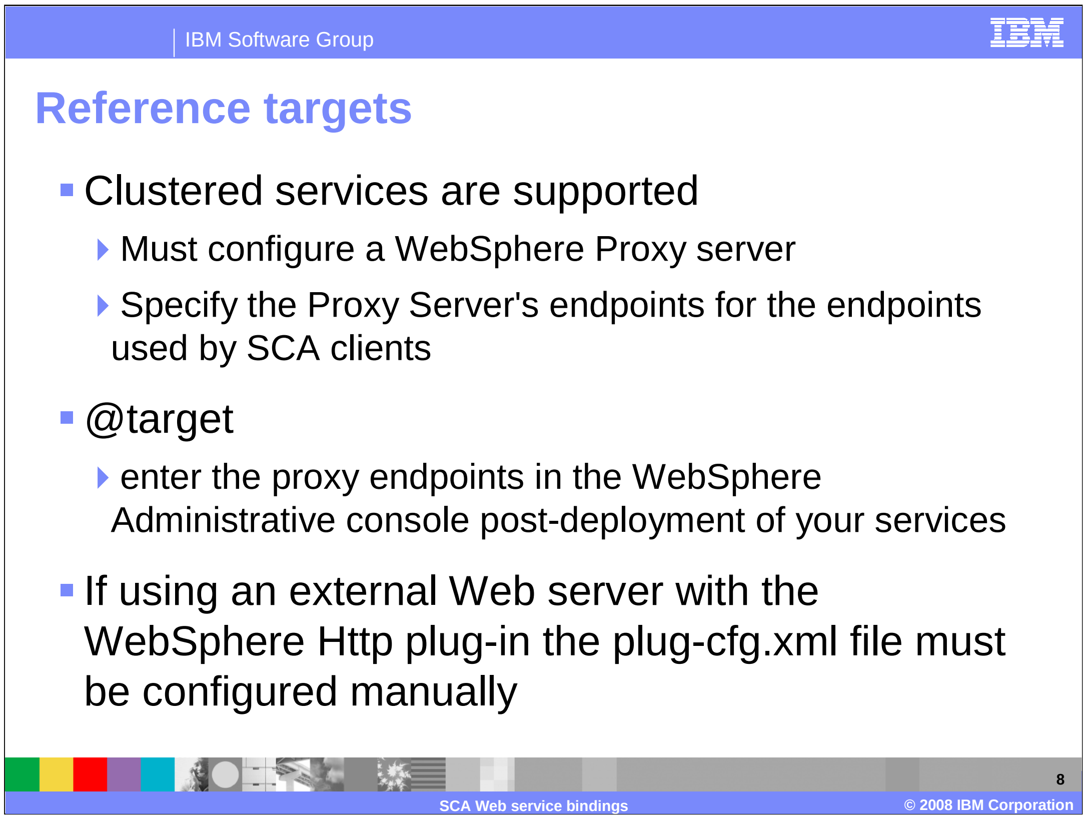
wsdl.port

wsdl.binding

wsdl.service

no wsdl element

The wsdl.port and wsdl.service both require port definitions. For wsdl.binding and "no wsdl document" an appropriate service/port entry will be generated by the runtime.

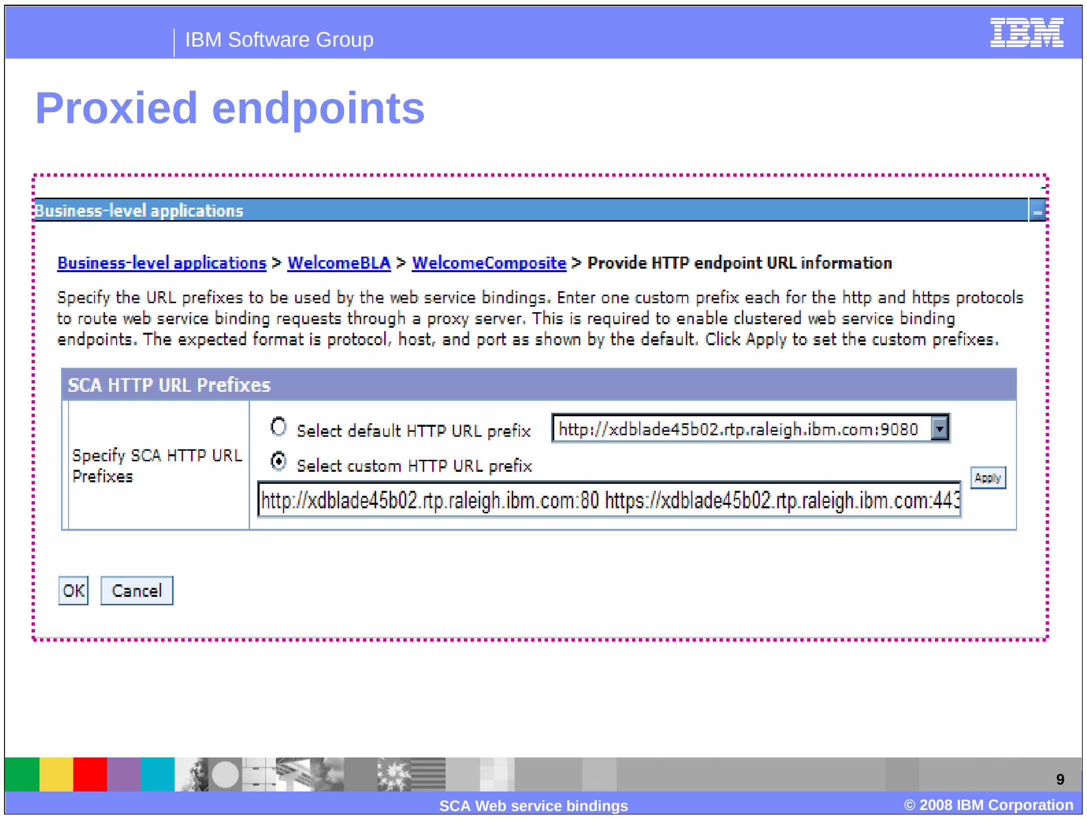


Reference targets

- Clustered services are supported
  - Must configure a WebSphere Proxy server
  - Specify the Proxy Server's endpoints for the endpoints used by SCA clients
- @target
  - enter the proxy endpoints in the WebSphere Administrative console post-deployment of your services
- If using an external Web server with the WebSphere Http plug-in the plug-cfg.xml file must be configured manually

For clustered services (supported), a WebSphere proxy server and proxy server’s endpoints needs to be configured for the endpoints used by SCA clients. This condition applies whether you specify client endpoints using @target or as endpoints specified on binding.ws or in the WSDL location attribute. For example, for @target annotation you need to enter the proxy endpoints in the WebSphere administrative console post-deployment of your services. While still in the administrative console, in the custom HTTP endpoints URL's textbox you need to enter one secure and one non-secure HTTP endpoint that corresponds to the endpoints your proxy server is listening on. The proxy server will then route the request to a specific application server that hosts your service.

Note: ** if using an external Web server with the WebSphere HTTP plug-in the plugin-cfg.xml file must be configured manually.


IBM Software Group

Proxied endpoints

Business-level applications

Business-level applications > WelcomeBLA > WelcomeComposite > **Provide HTTP endpoint URL information**

Specify the URL prefixes to be used by the web service bindings. Enter one custom prefix each for the http and https protocols to route web service binding requests through a proxy server. This is required to enable clustered web service binding endpoints. The expected format is protocol, host, and port as shown by the default. Click Apply to set the custom prefixes.

SCA HTTP URL Prefixes

Specify SCA HTTP URL Prefixes

Select default HTTP URL prefix: http://xdblade45b02.rtp.raleigh.ibm.com:9080

[✓ Select custom HTTP URL prefix]

http://xdblade45b02.rtp.raleigh.ibm.com:80 https://xdblade45b02.rtp.raleigh.ibm.com:443

9

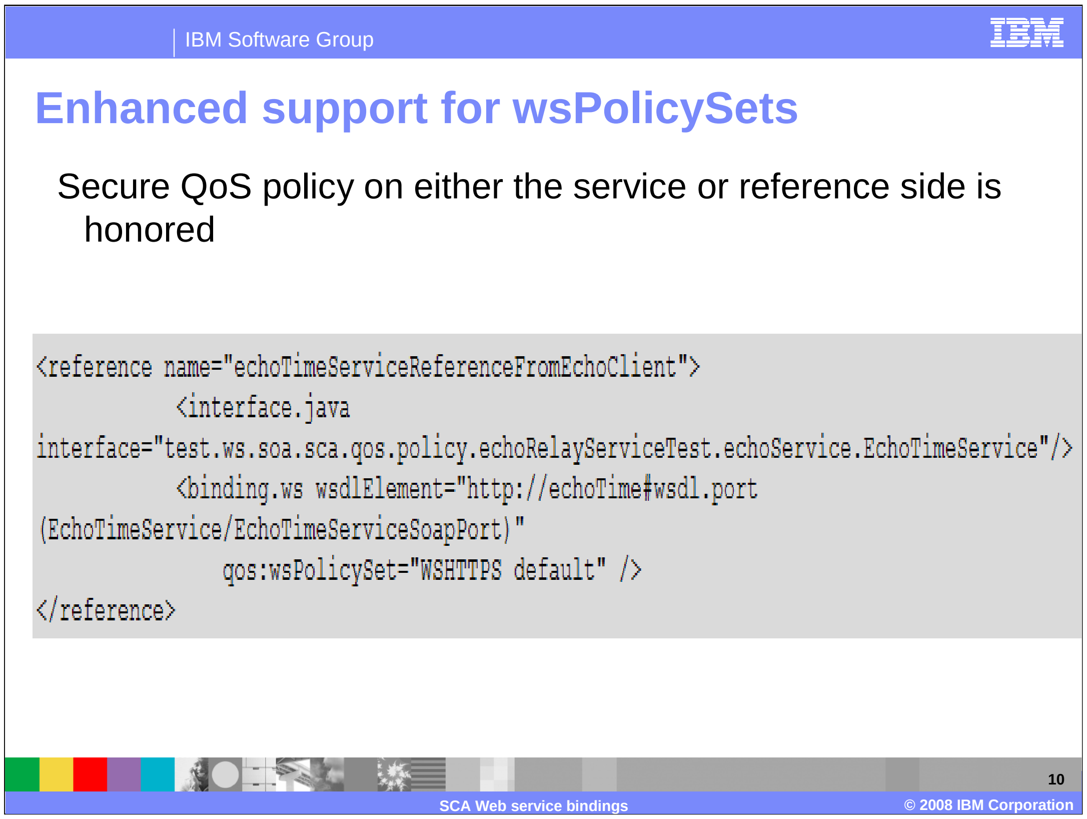
SCA Web service bindings © 2008 IBM Corporation

Here is an example of proxied endpoints from the administrative console.

(1) for **@target** you need to enter the proxy endpoints in the WebSphere administrative console post-deployment of your services. As you can see from the screen capture in the **custom HTTP endpoints URL's** textbox you need to enter one secure and one non-secure HTTP endpoint that corresponds to the endpoints your proxy server is listening on. The proxy server will then route the request to a specific application server that hosts your service.

(2) for **binding.ws** or **WSDL specified endpoints** you would enter the same base URI information (example: Protocol, host, and port) that you enter in the custom HTTP endpoints textbox. The key point is the endpoints need to be those of the proxy server.

The slide should also show the https endpoint. You need to enter both on the same line separated by a space. Panel is not very good in showing this but the text does say it.



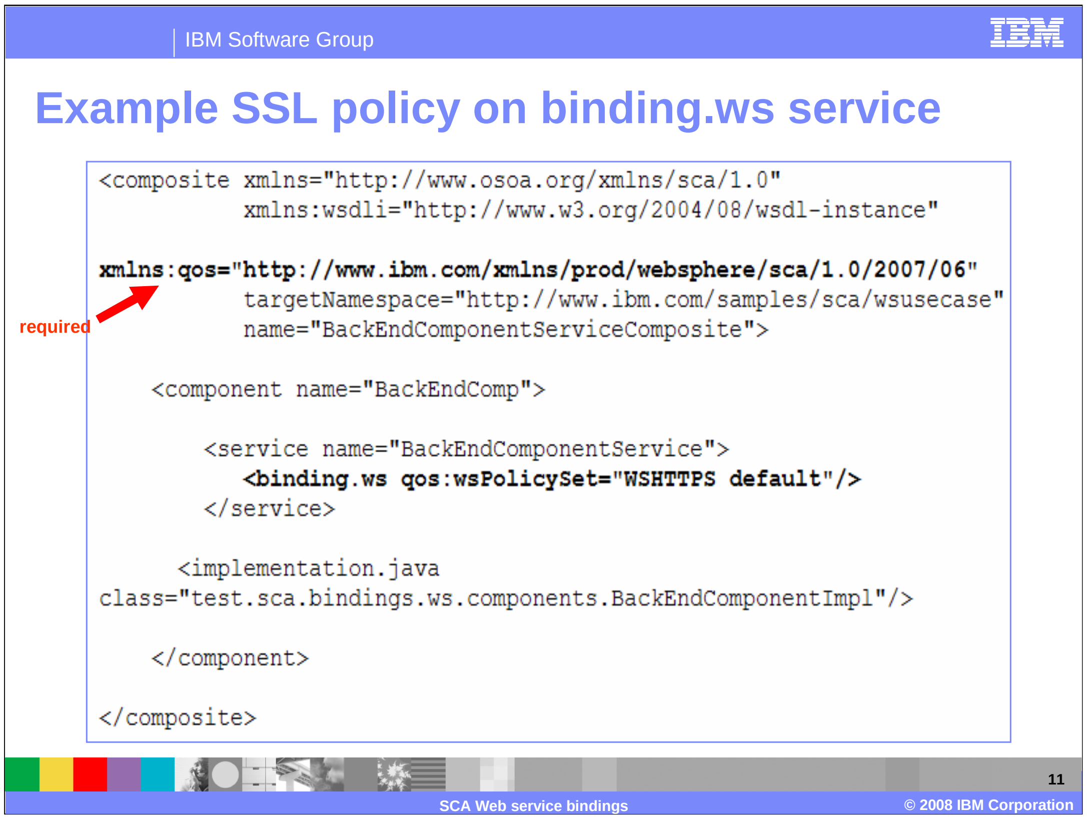
Enhanced support for wsPolicySets

Secure QoS policy on either the service or reference side is honored

```
<reference name="echoTimeServiceReferenceFromEchoClient">
            <interface.java
interface="test.ws.soa.sca.qos.policy.echoRelayServiceTest.echoService.EchoTimeService"/>
            <binding.ws wsdlElement="http://echoTime#wsdl.port
(EchoTimeService/EchoTimeServiceSoapPort)"
                qos:wsPolicySet="WSHTTPS default" />
</reference>
```

In the past, services exposed to the Web services binding were always available over both secure and non-secure endpoints. Now, if a service is described with a QoS policy that requires SSL, the service will not be available on a non-secure endpoint.

Additionally, when using <reference target="">, the system-determined endpoints will honor a secure QoS policy on either the service or reference side as shown in the example. The next slide will address how **qos:namespace** is resolved..



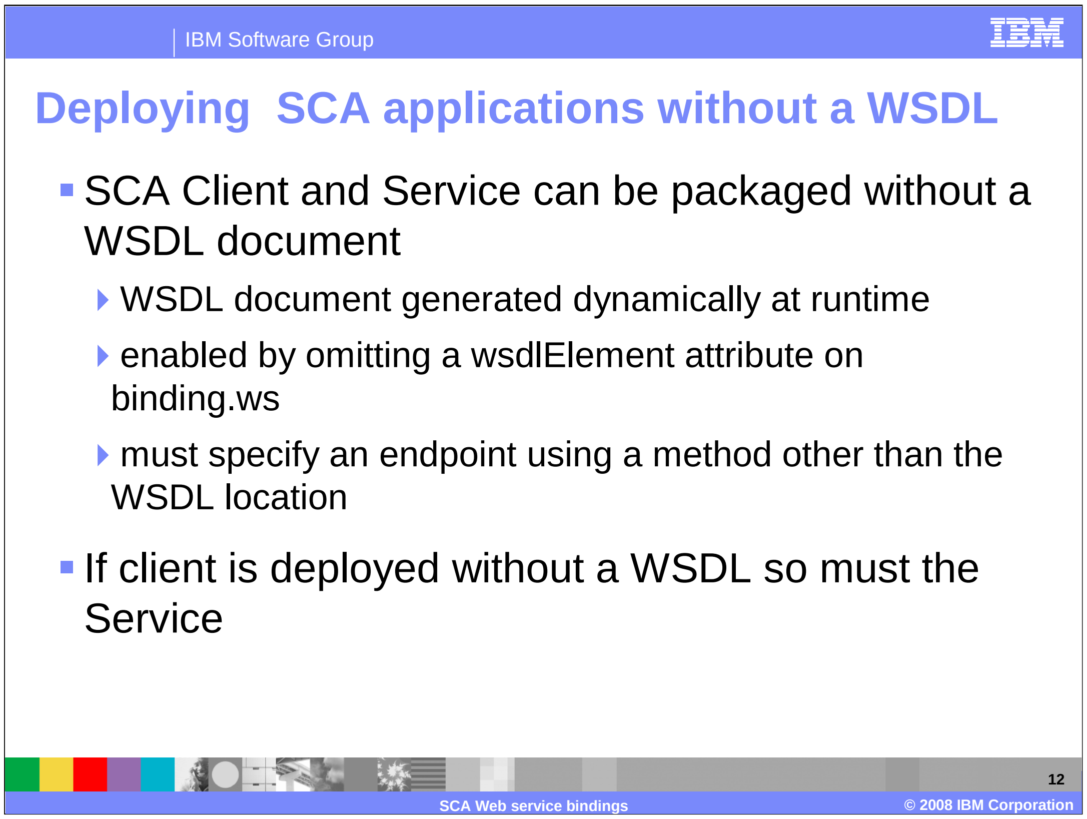
Example SSL policy on binding.ws service

```
<composite xmlns="http://www.osoa.org/xmlns/sca/1.0"
           xmlns:wsdli="http://www.w3.org/2004/08/wsdl-instance"

xmlns:qos="http://www.ibm.com/xmlns/prod/websphere/sca/1.0/2007/06"
           targetNamespace="http://www.ibm.com/samples/sca/wsusecase"
           name="BackEndComponentServiceComposite">

    <component name="BackEndComp">

        <service name="BackEndComponentService">
           <binding.ws qos:wsPolicySet="WSHTTPS default"/>
        </service>

      <implementation.java
class="test.sca.bindings.ws.components.BackEndComponentImpl"/>

    </component>

</composite>
```

required

This example shows how SSL policy is applied to **binding.ws** service. Reference side is similar. The key point here is that the **xmlns** for qos: is required.

Note that not ALL types policy can be applied. Only the policy that SCA FP supports.



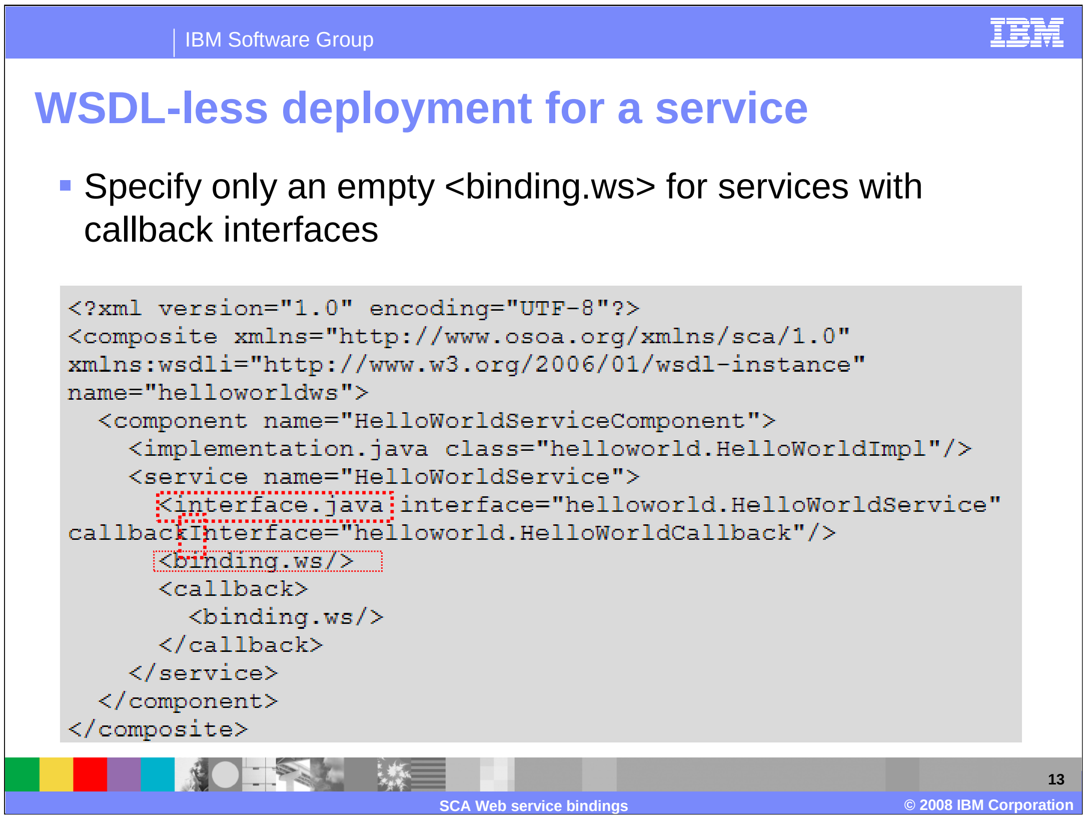
Deploying SCA applications without a WSDL

- SCA Client and Service can be packaged without a WSDL document
  - WSDL document generated dynamically at runtime
  - enabled by omitting a wsdlElement attribute on binding.ws
  - must specify an endpoint using a method other than the WSDL location
- If client is deployed without a WSDL so must the Service

SCA Client and Service can be package without a WSDL document. This is enabled by omitting a wsdlElement attribute on binding.ws and not packing a WSDL document with your contribution. In this scenario, the WSDL document will then be generated dynamically at runtime. Clients must specify an endpoint using a method other than the WSDL location attribute since WSDL does not exist.

Normally, both the client and service need to use the same WSDL, therefore if client is deployed without a WSDL so must the Service.

The JAXB and JAX-WS specifications are used to define the mapping between the Java™ service interface, (including method arguments, return types, and exceptions), and the WSDL definition. See the JAXB presentation


IBM Software Group

WSDL-less deployment for a service

- Specify only an empty <binding.ws> for services with callback interfaces

```
<?xml version="1.0" encoding="UTF-8"?>
<composite xmlns="http://www.osoa.org/xmlns/sca/1.0"
xmlns:wsdli="http://www.w3.org/2006/01/wsdl-instance"
name="helloworldws">
  <component name="HelloWorldServiceComponent">
    <implementation.java class="helloworld.HelloWorldImpl"/>
    <service name="HelloWorldService">
      <interface.java interface="helloworld.HelloWorldService"
callbackInterface="helloworld.HelloWorldCallback"/>
      <binding.ws/>
      <callback>
        <binding.ws/>
      </callback>
    </service>
  </component>
</composite>
```

13

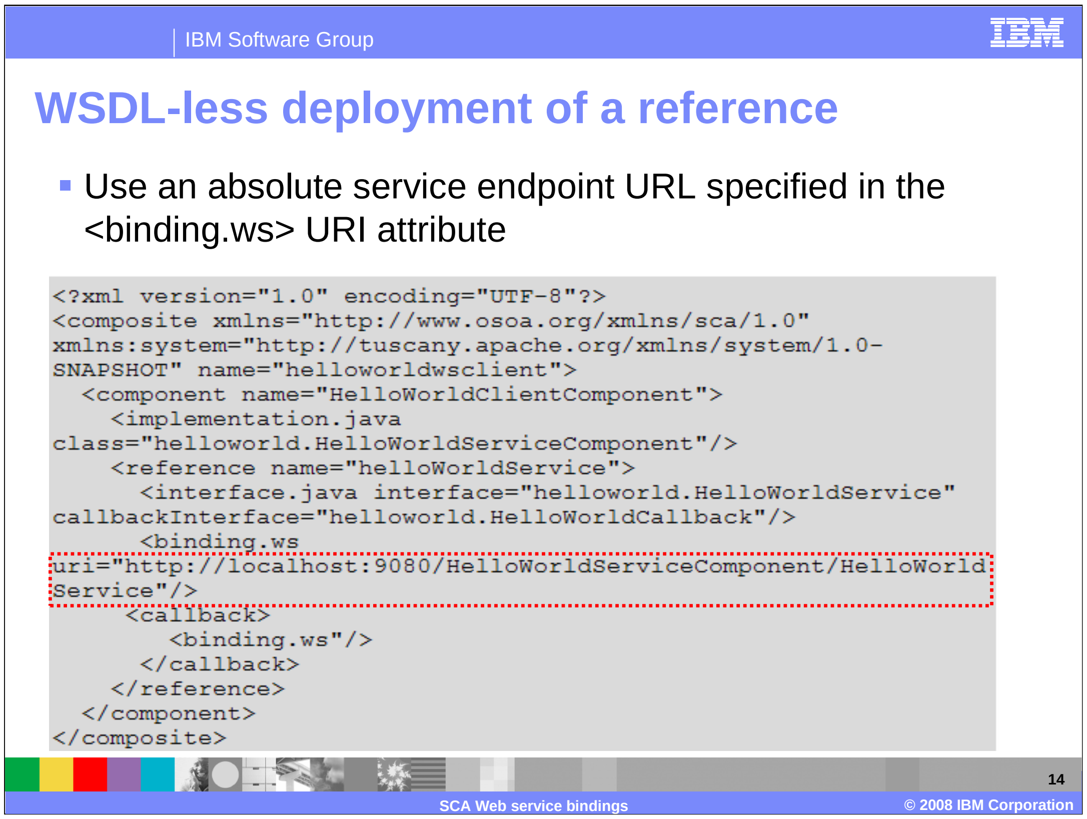
SCA Web service bindings © 2008 IBM Corporation

This example is showing how to use WSDL-less deployment **for a Service**. For services with callback interfaces specify only an empty <binding.ws>. This is because The callbackinterface requires its own binding.ws for the callback since a callback is implemented as a bi-directional service. Therefore it has its own service and reference defined in SCDL. Additionally, the callback also must define its interface.

The interface.java and an appropriate callbackInterface are required service elements.

The ?wsdl function on a WSDL-less deployed service with a <binding.ws> will return a WSDL document that can be used for client-side development, if needed.

Note that for the SCA feature pack the top-down approach (WSDL-less) is the best practice approach



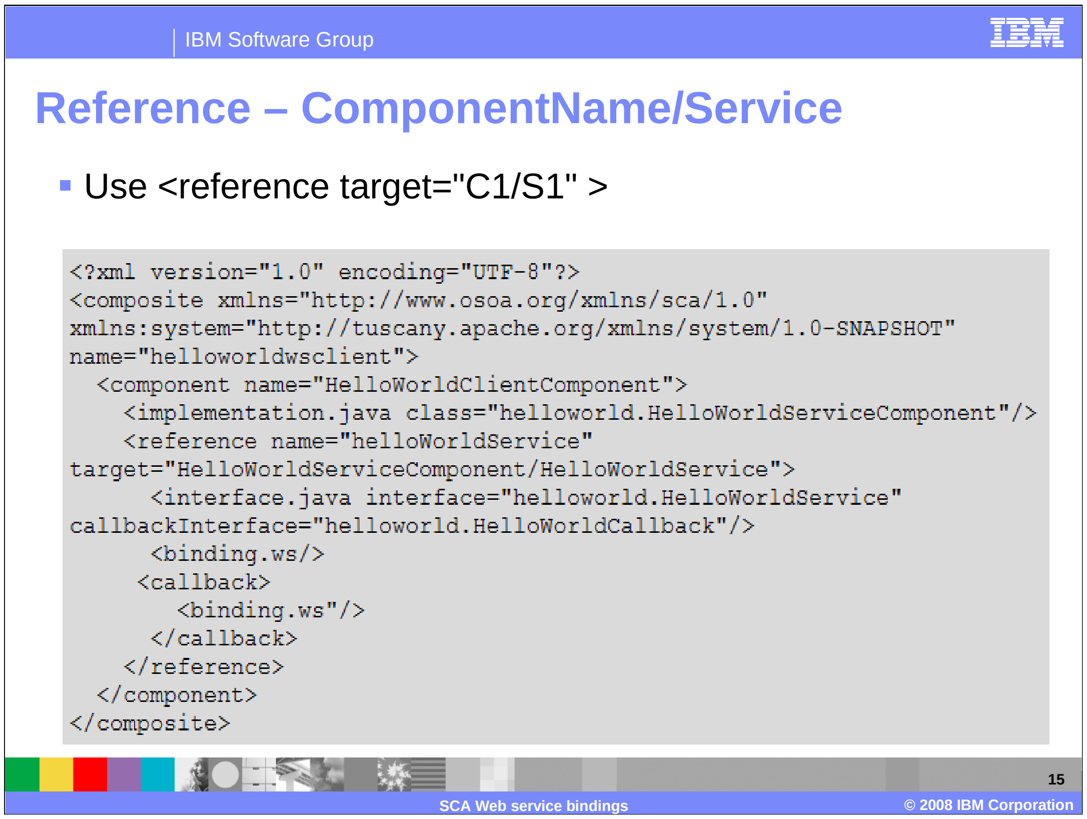
WSDL-less deployment of a reference

- Use an absolute service endpoint URL specified in the <binding.ws> URI attribute

```
<?xml version="1.0" encoding="UTF-8"?>
<composite xmlns="http://www.osoa.org/xmlns/sca/1.0"
xmlns:system="http://tuscany.apache.org/xmlns/system/1.0-
SNAPSHOT" name="helloworldwsclient">
  <component name="HelloWorldClientComponent">
    <implementation.java
class="helloworld.HelloWorldServiceComponent"/>
    <reference name="helloWorldService">
      <interface.java interface="helloworld.HelloWorldService"
callbackInterface="helloworld.HelloWorldCallback"/>
      <binding.ws
uri="http://localhost:9080/HelloWorldServiceComponent/HelloWorld
Service"/>
     <callback>
        <binding.ws"/>
      </callback>
    </reference>
  </component>
</composite>
```

This example is showing how to use WSDL-less deployment **for a Reference**.

For references with callback interfaces specify only an empty <binding.ws> just like in the service scenario. The interface.java and an appropriate callbackInterface are required reference elements. What’s different for a reference is the endpoint. Determine how the endpoint of the referenced service is specified. Use an absolute service endpoint URL specified in the <binding.ws> URI attribute.

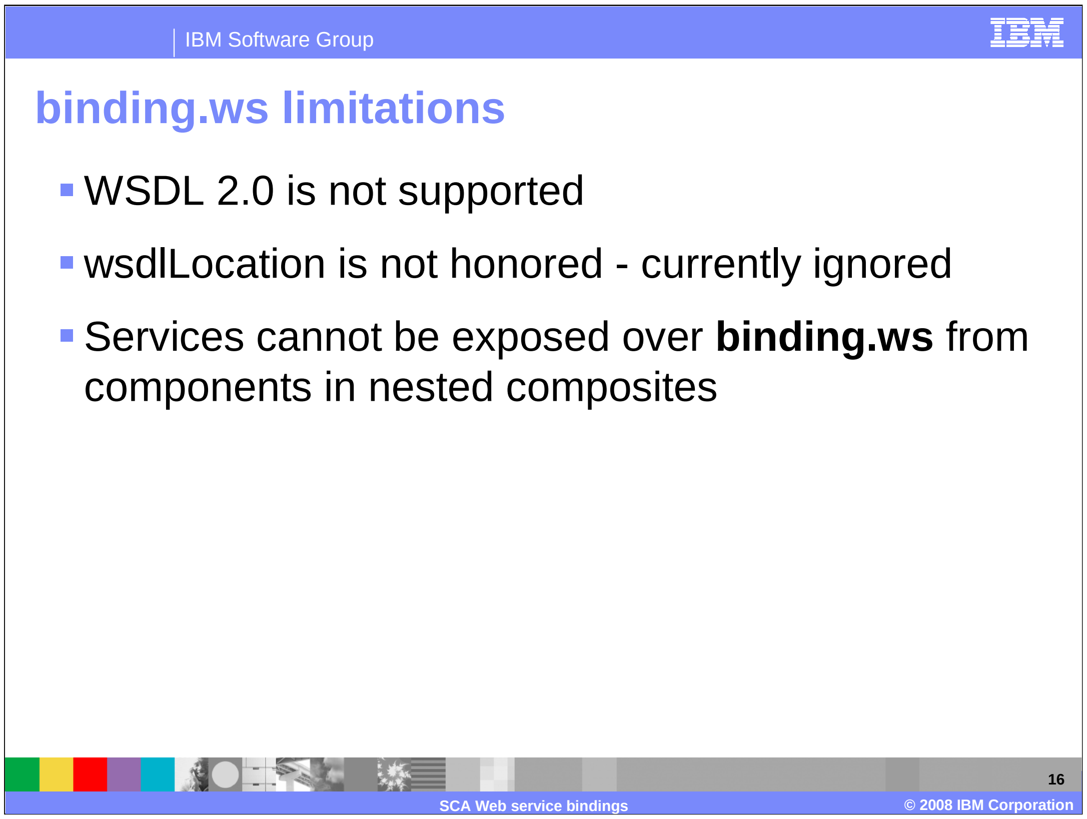


Reference – ComponentName/Service

- Use <reference target="C1/S1" >

```
<?xml version="1.0" encoding="UTF-8"?>
<composite xmlns="http://www.osoa.org/xmlns/sca/1.0"
xmlns:system="http://tuscany.apache.org/xmlns/system/1.0-SNAPSHOT"
name="helloworldwsclient">
  <component name="HelloWorldClientComponent">
    <implementation.java class="helloworld.HelloWorldServiceComponent"/>
    <reference name="helloWorldService"
target="HelloWorldServiceComponent/HelloWorldService">
      <interface.java interface="helloworld.HelloWorldService"
callbackInterface="helloworld.HelloWorldCallback"/>
      <binding.ws/>
     <callback>
        <binding.ws"/>
      </callback>
    </reference>
  </component>
</composite>
```

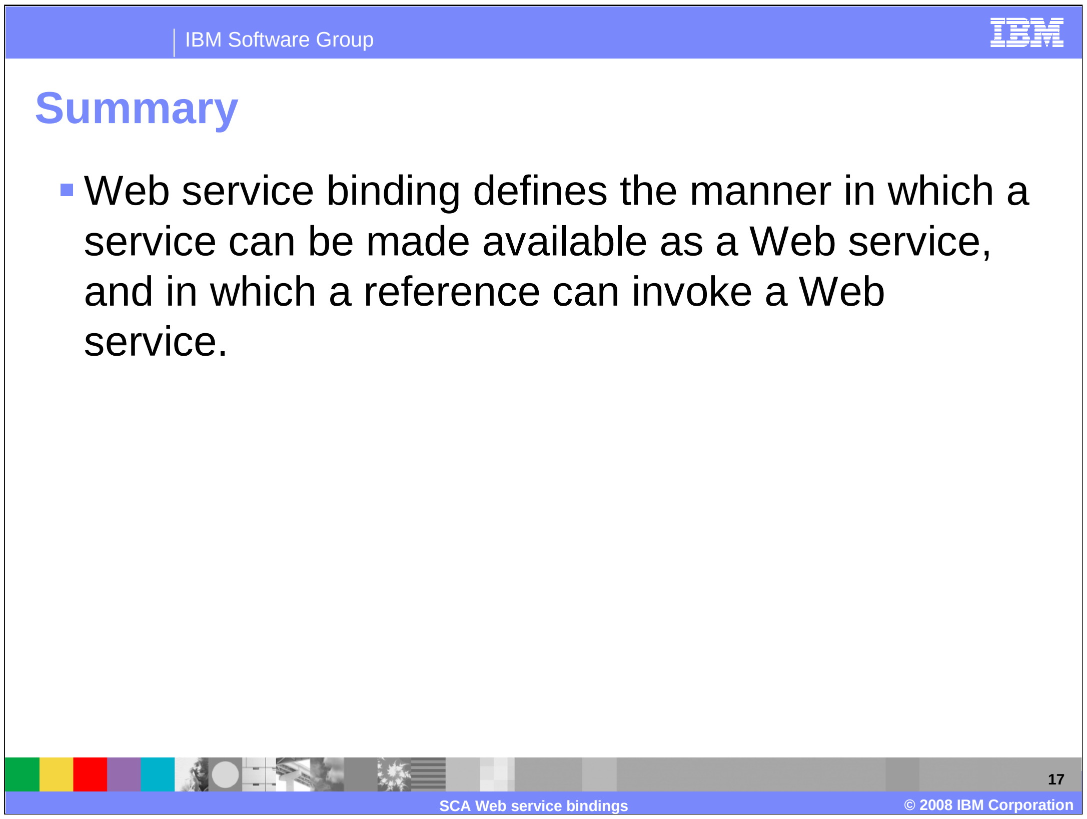
This example shows the use of a reference specifying the target attribute to resolve the service endpoint (Example: <reference target="HellowWorldServiceComponent/HelloWorldService">). This is only applicable if the reference targets a service in the same SCA domain (WebSphere cell) as the reference.



binding.ws limitations

- WSDL 2.0 is not supported
- wsdlLocation is not honored - currently ignored
- Services cannot be exposed over **binding.ws** from components in nested composites

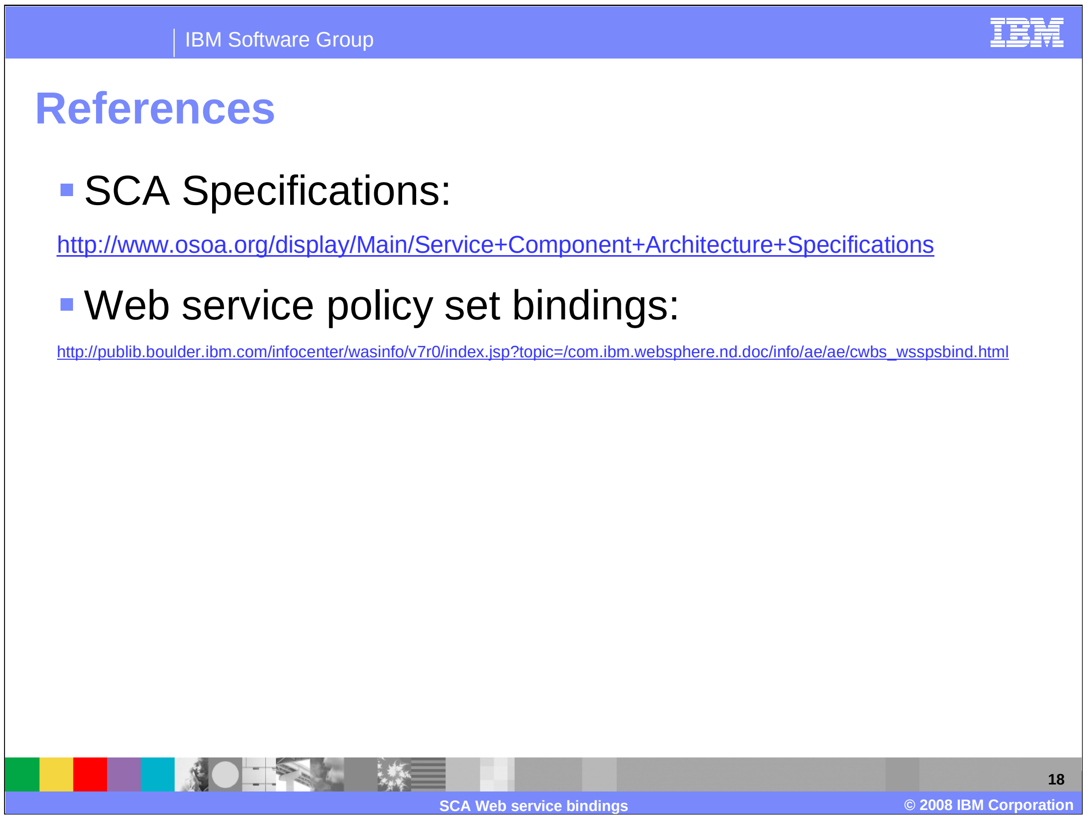
Here are some Web service binding limitations for SCA feature pack; WSDL 2.0 is not supported, which implies wsdl.endpoint is not supported, wsdlLocation is not honored, this is currently ignored and services cannot be exposed over **binding.ws** from components in nested composites.



Summary

- Web service binding 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.

Web service binding 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.


IBM Software Group

References

- SCA Specifications:

http://www.osoa.org/display/Main/Service+Component+Architecture+Specifications

- Web service policy set bindings:

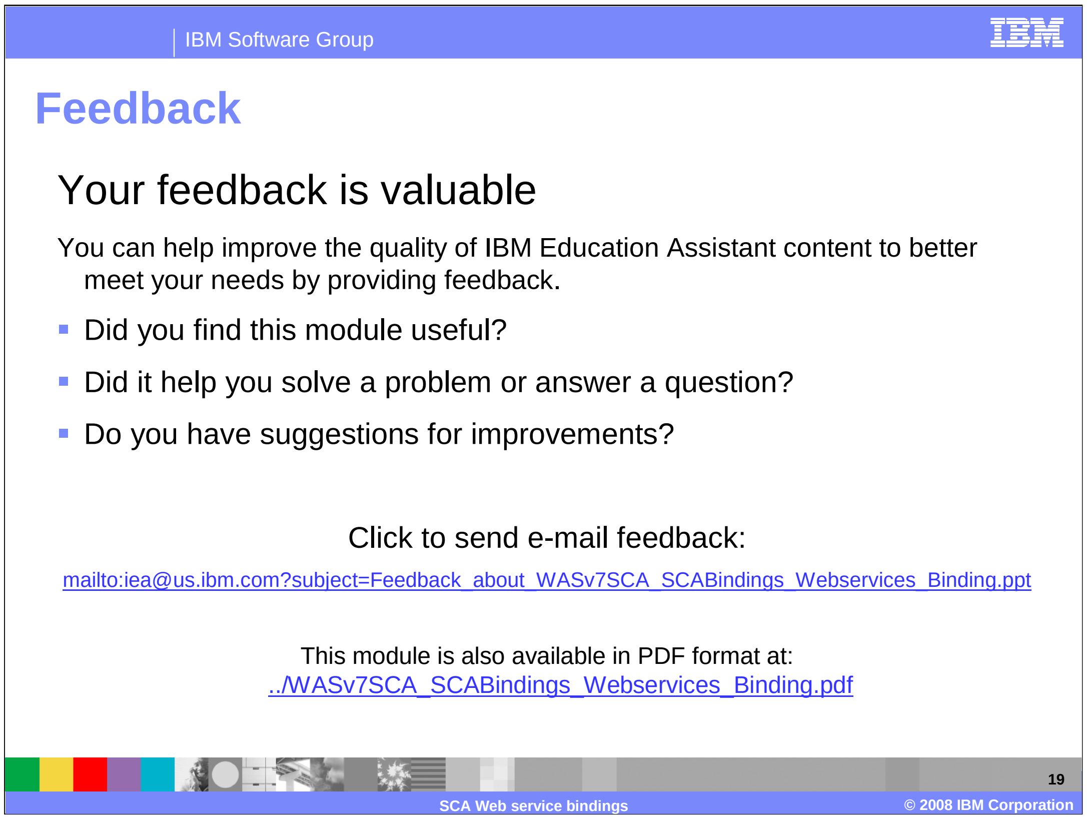
http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/cwbs_wsspsbind.html

18

SCA Web service bindings

© 2008 IBM Corporation

The Service Component Architecture specifications and the information center article on Web service bindings are available at the addresses shown here.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

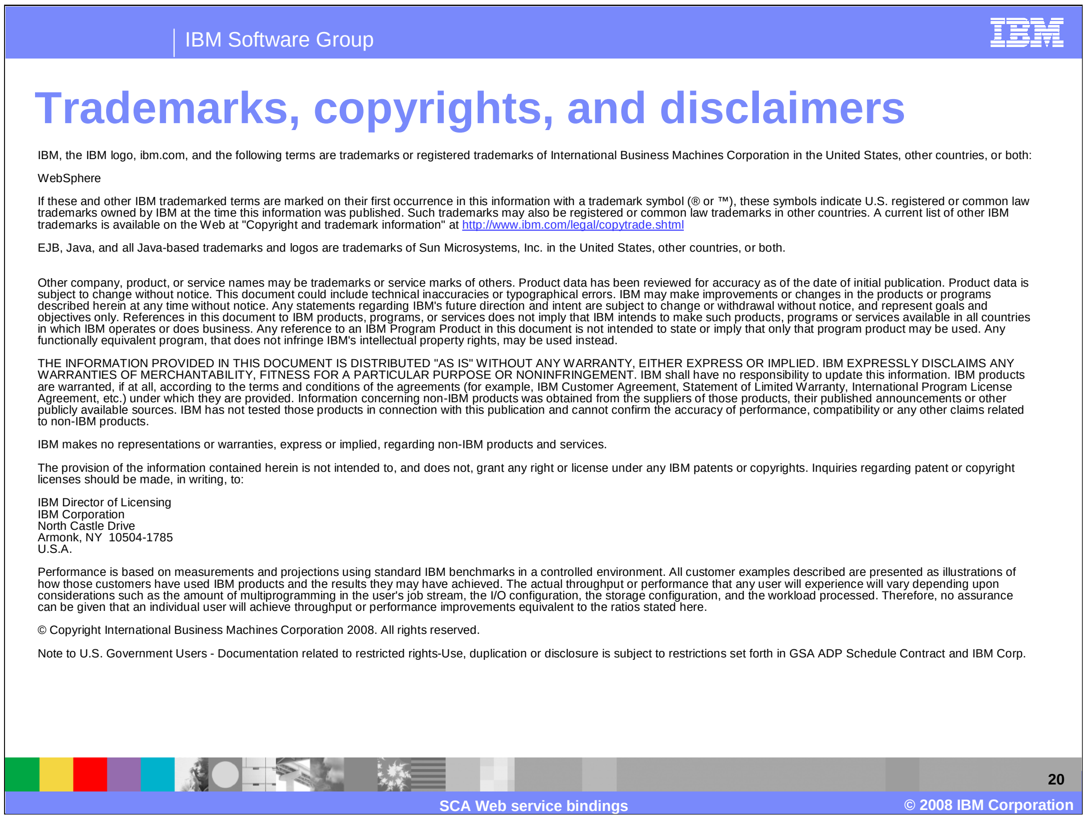
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WASv7SCA_SCABindings_Webservices_Binding.ppt

This module is also available in PDF format at:
../WASv7SCA_SCABindings_Webservices_Binding.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, copyrights, and disclaimers