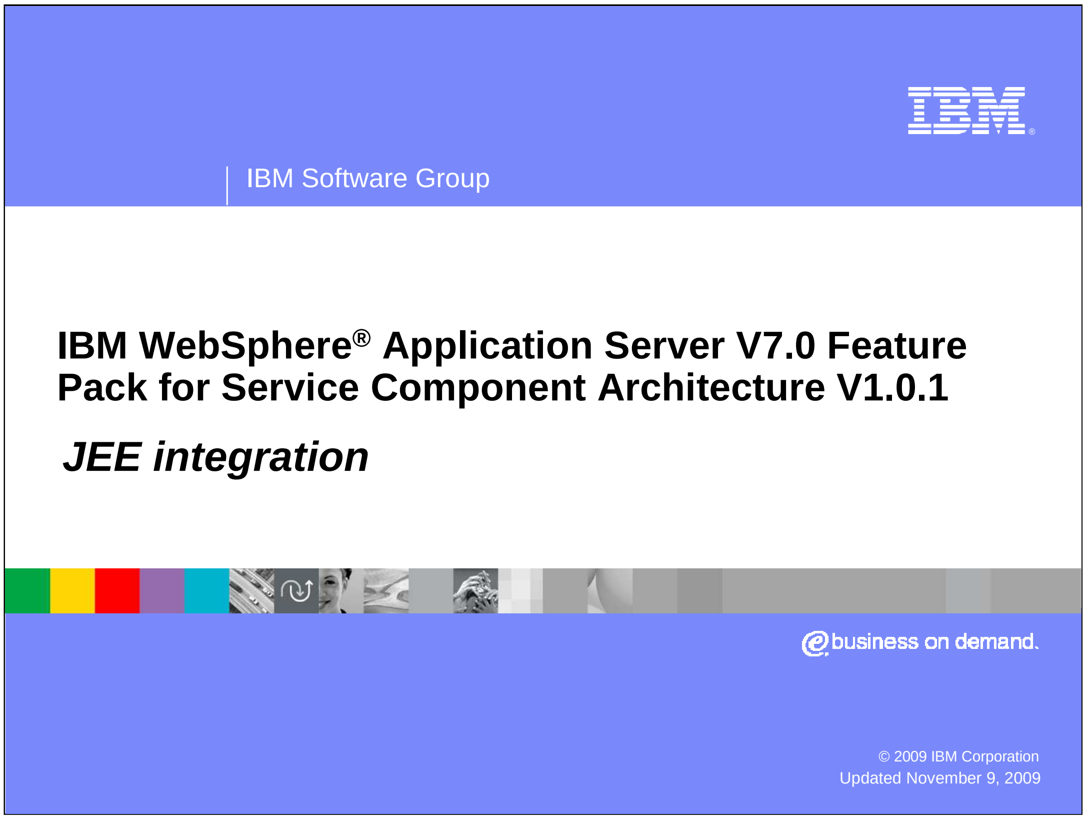


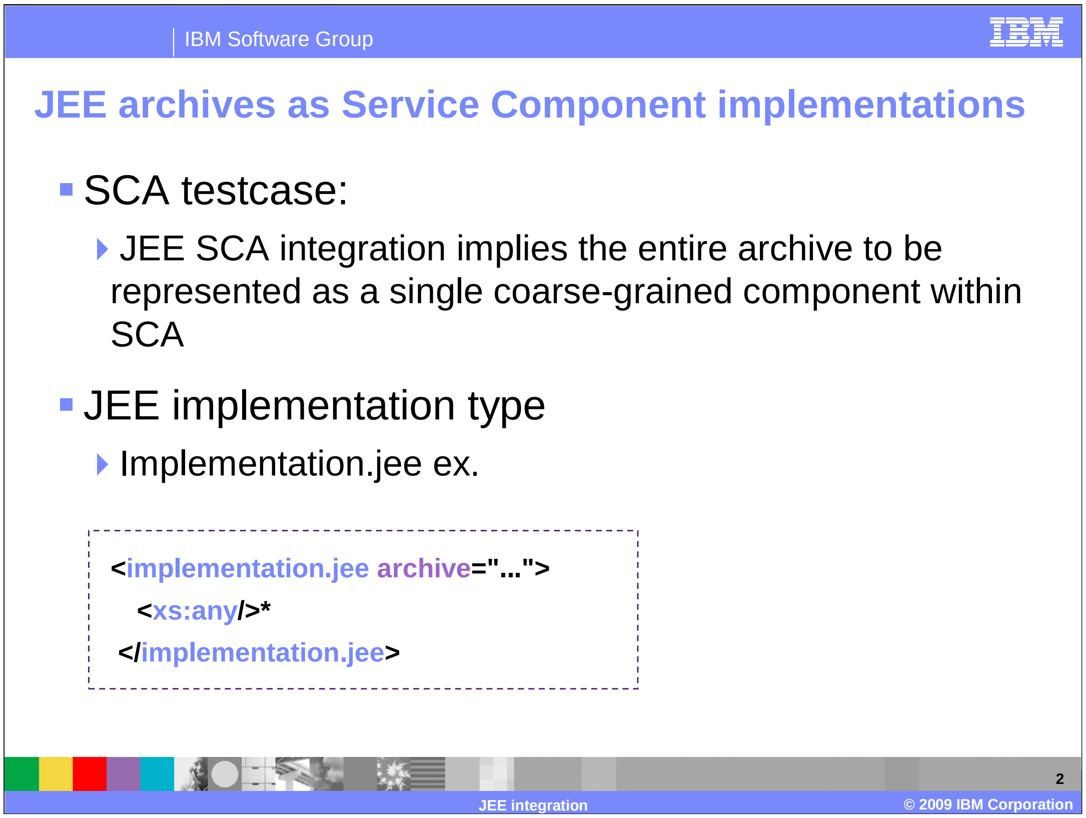
IBM Software Group

IBM WebSphere® Application Server V7.0 Feature Pack for Service Component Architecture V1.0.1

JEE integration

Updated November 9, 2009

This presentation will cover the JEE integration in the SCA feature pack. It will specifically focus on the integration of SCA and Java™ EE within the context of a Java EE application. It will also look at the use of Java EE components as service component implementations and the deployment of Java EE archives either within or as SCA contributions.



IBM Software Group

JEE archives as Service Component implementations

- SCA testcase:
  - JEE SCA integration implies the entire archive to be represented as a single coarse-grained component within SCA
- JEE implementation type
  - Implementation.jee ex.

```
<implementation.jee archive="...">
   <xs:any/>*
 </implementation.jee>
```

2

JEE integration

© 2009 IBM Corporation

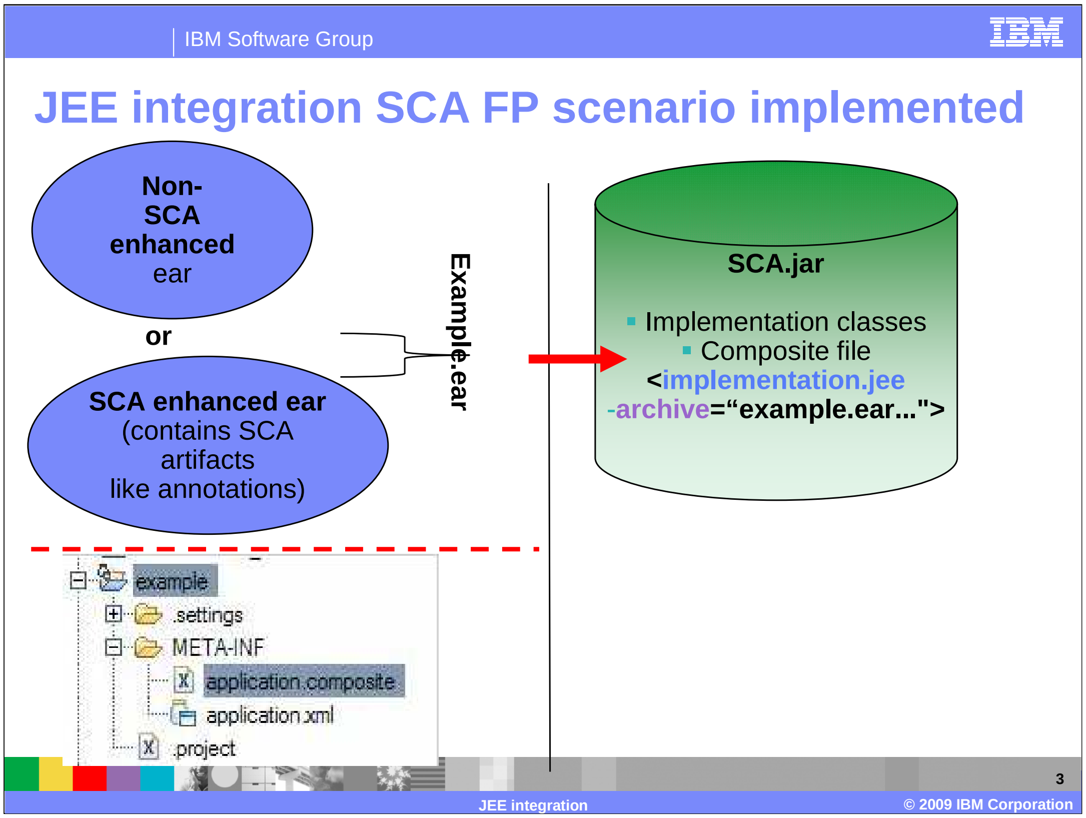
The current SCA JEE integration is the entire archive to be represented as a single coarse-grained component within SCA.

The ***archive*** attribute specifies a relative path to the Java EE archive that serves as implementation artifact. The context of that relative path (the value “.”) is the location of the artifact that contains the ***implementation.jee*** element. All Java EE components contained in the archive are deployed, regardless of any SCA enhancements present.

Every deployed SCA component using the JEE implementation type represents a deployment of the referred Java EE archive.

The archive that is referred to by <implementation.jee> can be an artifact within a larger contribution (that is an EAR inside a larger compressed file), or the archive can itself be a contribution. In the latter case, the @archive attribute can be left unspecified, and the archive is assumed to be the archive of the contribution itself.

The component type derived from a Java EE archive depends on whether it has been enhanced with SCA artifacts or contains an application composite.



IBM Software Group

JEE integration SCA FP scenario implemented

Non-SCA enhanced ear

or

SCA enhanced ear (contains SCA artifacts like annotations)

Example.ear

SCA.jar

- Implementation classes
- Composite file

<implementation.jee -archive=“example.ear...">

3

JEE integration

© 2009 IBM Corporation

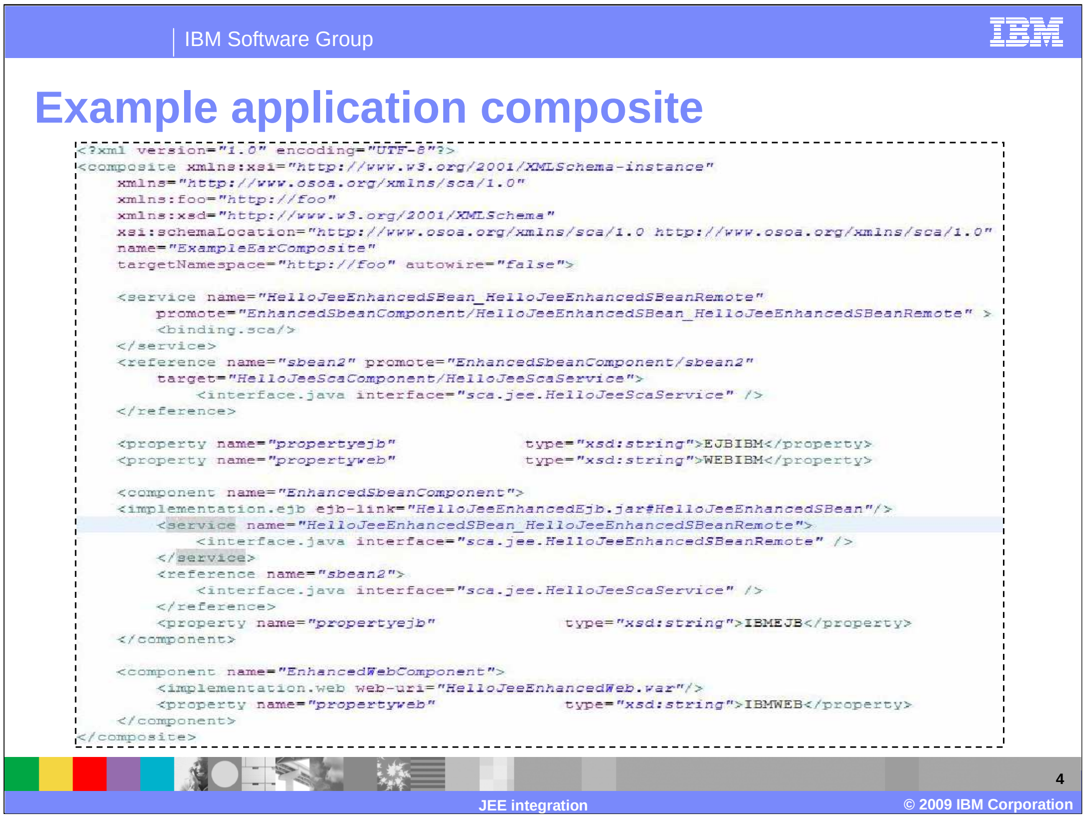
Let’s look at the scenario supported for SCA feature pack as far as JEE integration is concerned. As mentioned in the previous slide, implementation.jee is what is supported in this feature pack.

Let’s look at an example:

Suppose there is an EAR containing one or more EJB jars and one or more Web modules . An SCA jar can use such an EAR as its component implementation using the <implementation.jee> element, specifying the EAR in the "archive" attribute as shown. Direct use of EJB jar module or Web module as component implementation in SCA composites **is not supported**.

The EAR itself can contain no SCA artifacts like annotations, in which case it is called a **non-SCA enhanced EAR** (when it contains annotations, its called an SCA enhanced EAR). The user can choose to provide an application composite file for the EAR (located within the META-INF directory of the EAR). This application composite can internally use <implementation.ejb> and <implementation.web> as component implementations pointing to EJB jars and Web wars within the EAR. In such cases the application composite will "promote" relevant services, references and properties to make them available to an SCA component using the EAR as implementation by way of <implementation.jee>.

What this means is that direct use of <implementation.ejb> and <implementation.web> as component implementation from SCA composites are not supported. Rather users must always use <implementation.jee> in such cases.



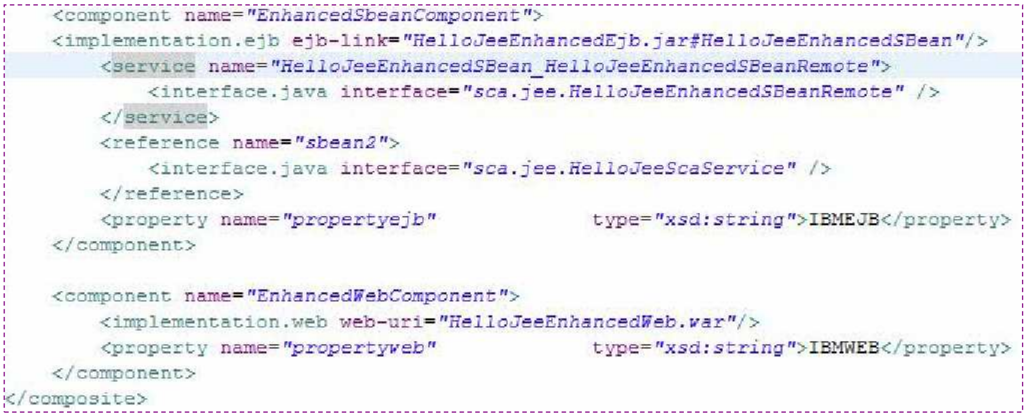
Example application composite

```
<?xml version="1.0" encoding="UTF-8"?>
<composite xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.osoa.org/xmlns/sca/1.0"
    xmlns:foo="http://foo"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xsi:schemaLocation="http://www.osoa.org/xmlns/sca/1.0 http://www.osoa.org/xmlns/sca/1.0"
    name="ExampleEarComposite"
    targetNamespace="http://foo" autowire="false">

    <service name="HelloJeeEnhancedSBean_HelloJeeEnhancedSBeanRemote"
        promote="EnhancedSbeanComponent/HelloJeeEnhancedSBean_HelloJeeEnhancedSBeanRemote" >
        <binding.sca/>
    </service>
    <reference name="sbean2" promote="EnhancedSbeanComponent/sbean2"
        target="HelloJeeScaComponent/HelloJeeScaService">
            <interface.java interface="sca.jee.HelloJeeScaService" />
    </reference>

    <property name="propertyejb"            type="xsd:string">EJBIBM</property>
    <property name="propertyweb"            type="xsd:string">WEBIBM</property>

    <component name="EnhancedSbeanComponent">
    <implementation.ejb ejb-link="HelloJeeEnhancedEjb.jar#HelloJeeEnhancedSBean"/>
        <service name="HelloJeeEnhancedSBean_HelloJeeEnhancedSBeanRemote">
            <interface.java interface="sca.jee.HelloJeeEnhancedSBeanRemote" />
        </service>
        <reference name="sbean2">
            <interface.java interface="sca.jee.HelloJeeScaService" />
        </reference>
        <property name="propertyejb"            type="xsd:string">IBMEJB</property>
    </component>

    <component name="EnhancedWebComponent">
        <implementation.web web-uri="HelloJeeEnhancedWeb.war"/>
        <property name="propertyweb"            type="xsd:string">IBMWEB</property>
    </component>
</composite>
```

This is an example application.composite with implementation.ejb and implementation.web. This example is taken from the HelloJEE sample provided with the product.

The next slide is a blown up version of some of the implementation.ejb and implementation.web lines of code shown here.

Implementation (application composite)

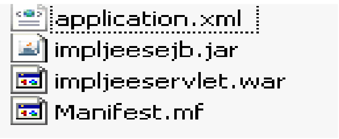


```
    <component name="EnhancedSbeanComponent">
    <implementation.ejb ejb-link="HelloJeeEnhancedEjb.jar#HelloJeeEnhancedSBean"/>
        <service name="HelloJeeEnhancedSBean_HelloJeeEnhancedSBeanRemote">
            <interface.java interface="sca.jee.HelloJeeEnhancedSBeanRemote" />
        </service>
        <reference name="sbean2">
            <interface.java interface="sca.jee.HelloJeeScaService" />
        </reference>
        <property name="propertyejb"             type="xsd:string">IBMEJB</property>
    </component>

    <component name="EnhancedWebComponent">
        <implementation.web web-uri="HelloJeeEnhancedWeb.war"/>
        <property name="propertyweb"             type="xsd:string">IBMWEB</property>
    </component>
</composite>
```

Note the implementation.ejb and implementation.web.

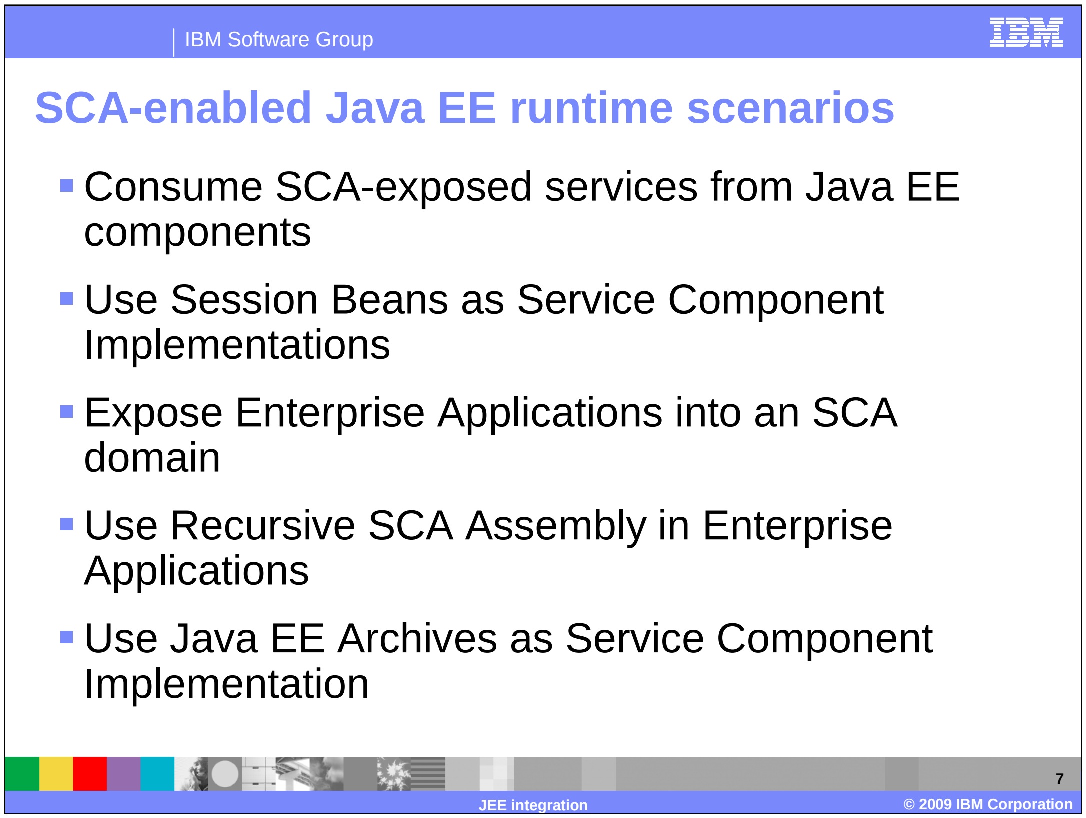
Basic steps for integrating a JEE application in SCA

1. Create a JEE ear file
2. Create an SCA jar file with the EAR file as it’s component implementation
3. Import both the EAR file and the SCA jar file as assets
4. Create a business level application
5. Add both the EAR and the SCA jar into the BLA as deployed assets
6. Start the BLA



application.xml
impljeesejb.jar
impljeeservlet.war
Manifest.mf

Here are simple steps for integrating a JEE application in SCA. First, create a JEE ear file. Then create an SCA jar file with the EAR file as it’s component implementation. Import both the EAR file and the SCA jar file as assets. You can then create a business level application.Add both the EAR and the SCA jar into the BLA as deployed assets. You can now start the BLA.



SCA-enabled Java EE runtime scenarios

- Consume SCA-exposed services from Java EE components
- Use Session Beans as Service Component Implementations
- Expose Enterprise Applications into an SCA domain
- Use Recursive SCA Assembly in Enterprise Applications
- Use Java EE Archives as Service Component Implementation

The following scenarios apply when enabling SCA on Java EE runtimes:

Consume SCA-exposed services from Java EE components

For example, a Web component should be able to easily consume a service implemented by a service component. This can be either by using SCA constructs in the implementation of a Java EE component implementation or by way of an EJB reference in combination with an EJB binding over an SCA service.

Use Session Beans as Service Component Implementations

The recursive assembly model of SCA provides rich means of configuration and re-use of service components that can be implemented as SCA composites or by some other implementation type. Session beans are the Java EE component implementation model and serve also as service component implementations.

Expose Enterprise Applications into an SCA domain

The SCA Assembly specification describes a deployment model for SCA contributions that provides cross-enterprise application assembly capabilities when layered over Java EE.

Use Recursive SCA Assembly in Enterprise Applications

SCA Assembly provides means to define sophisticated application assembly for enterprise applications.

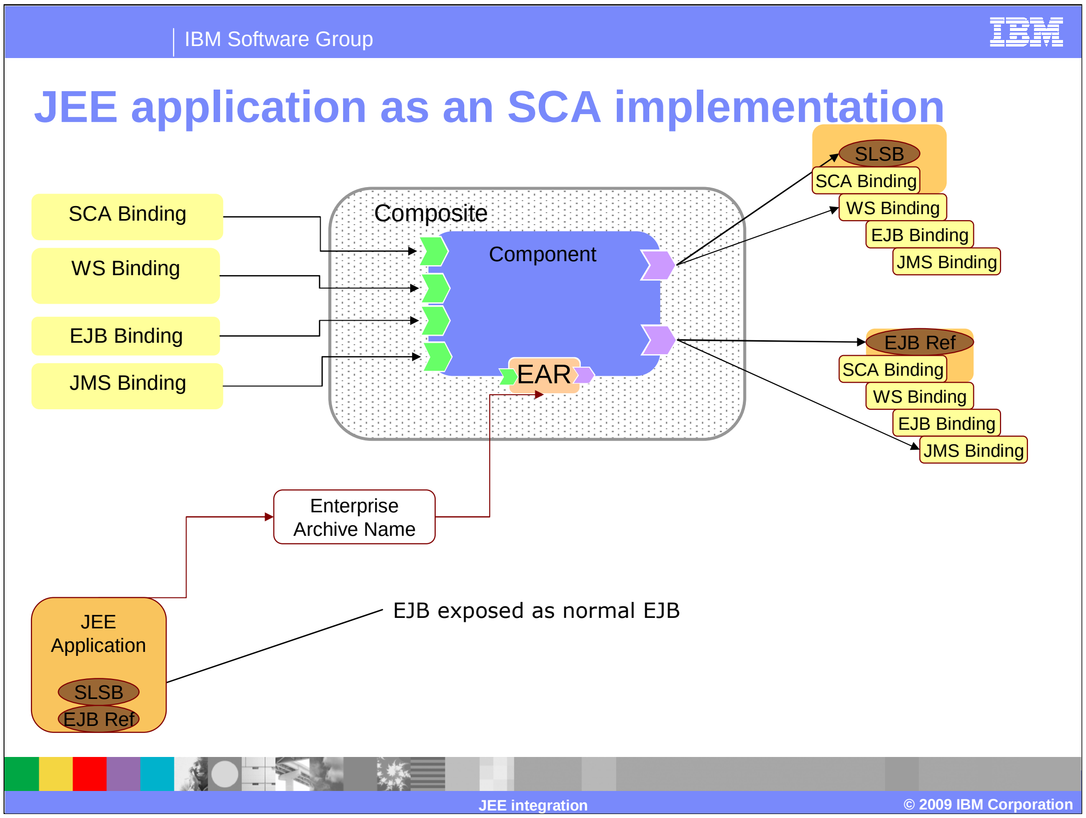
Deploy SCA Components as a Part of a Java EE application

SCA applications will typically combine Java EE components with components using other

implementation technologies, such as BPEL. The Java EE SCA specification enables the deployment of components implemented in these “foreign” technologies as part of a Java EE application, taking advantage of whatever tools and infrastructure support exists for the deployment and life cycle management of Java EE applications. Such components are treated as running in unmanaged environment and should not rely on Java EE features (access to java:comp/env, and so on.)

Use Java EE Archives as Service Component Implementation

SCA Java EE integration specification enables the creation of SCA applications whose components are implemented by Java JEE archives, so that they can be wired to each other and to components implemented using other technologies. This use-case requires a high-level view of the Java EE application as a single SCA component implementation, providing services and consuming references as a single component.



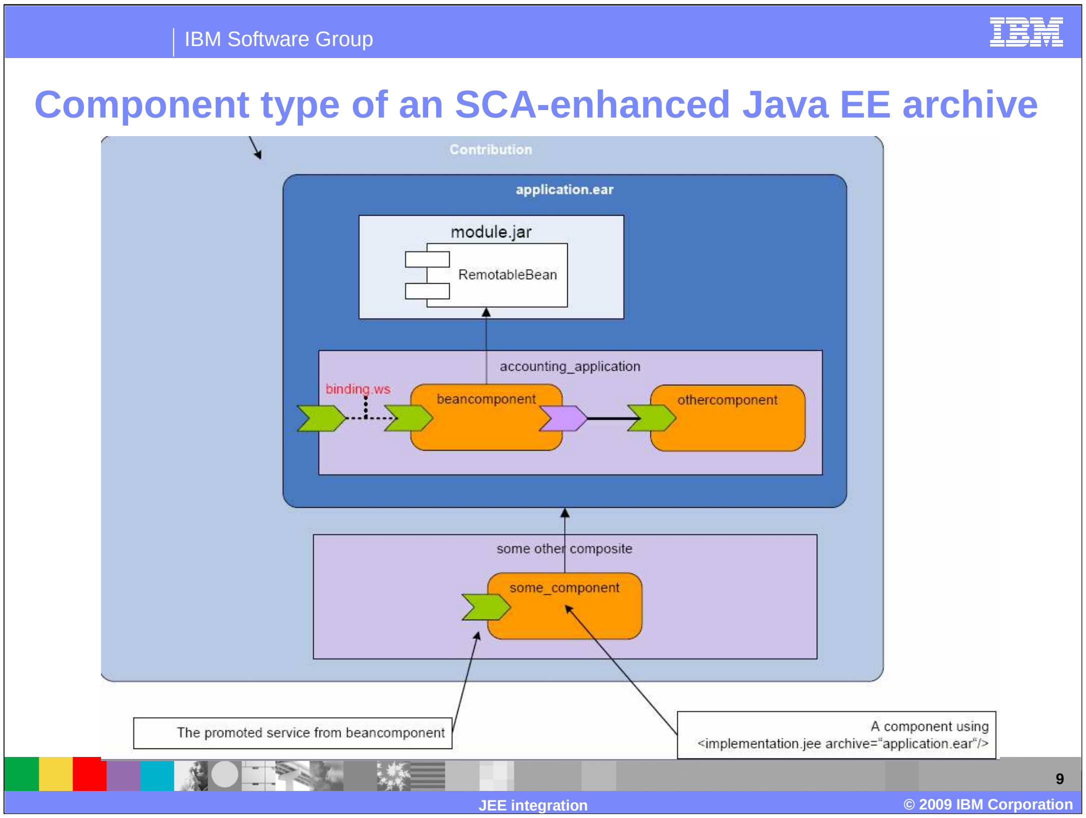
JEE application as a SCA implementation:

Here is a picture showing a JEE application as an SCA implementation. You can rewire an Enterprise JavaBeans (EJB) reference to a Service Component Architecture (SCA) reference that provides the same named operations. The rewiring does not require a change to existing code. Thus, you can use the SCA programming model without changing the implementation code.

To rewire an EJB reference to an SCA service in an SCA component, add a <reference> element to the JEE module component as shown. The reference names must match the name of the EJB reference. The contract of operations supported must also match. Because references must be promoted in the composite, the names of the references must be unique. After the EJB reference is promoted, all matching instances of the EJB reference in the module are rewired to the SCA service.

The EJB references are rewired to SCA references enabling Java EE modules to use SCA features without changing its implementation

As shown in the picture, the EAR is then used in SCA as a component implementation.



IBM Software Group

Component type of an SCA-enhanced Java EE archive

The promoted service from beancomponent

A component using <implementation.jee archive="application.ear"/>

9

JEE integration

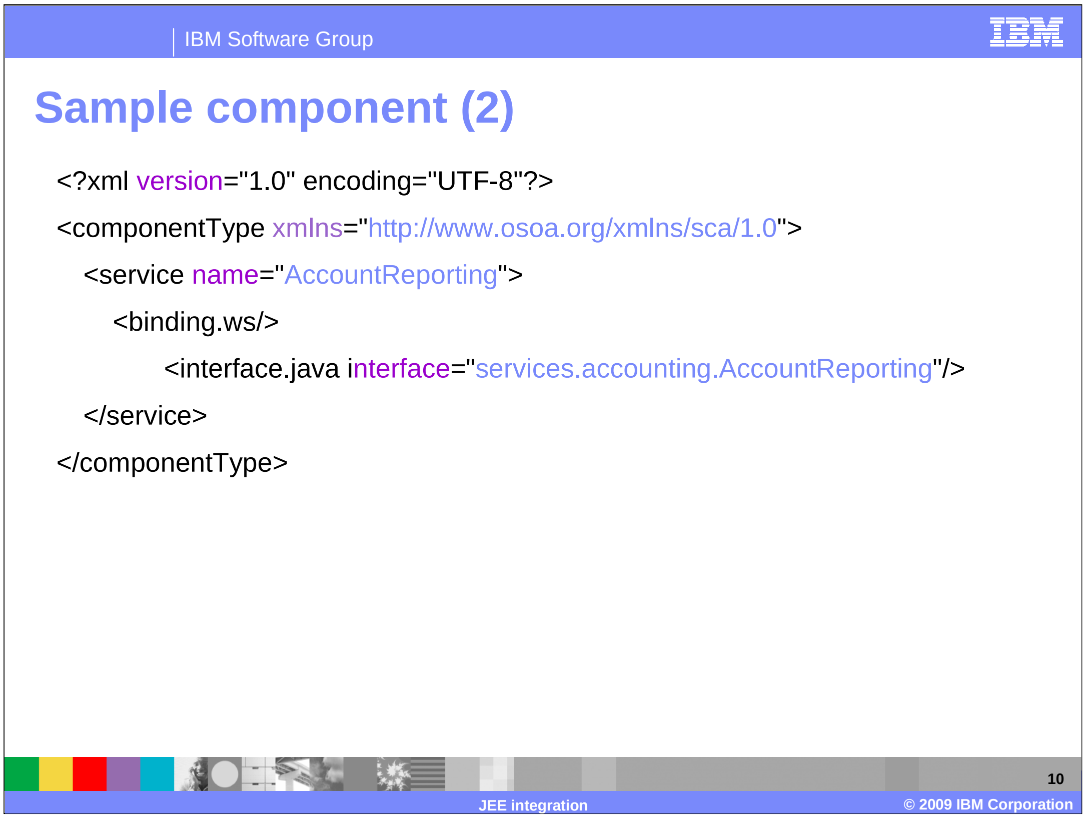
© 2009 IBM Corporation

A Java EE archive that contains an application composite has the component type of the application composite as its component type when used with the JEE implementation type.

Example: Let’s assume an SCA-enhanced Java EE archive ***application.ear*** that can be used as part of a larger non-Java EE contribution

that declares a service component in some other composite that uses the archive ***application.ear*** as implementation artifact. In that case the component type of the EAR archive will expose a service.

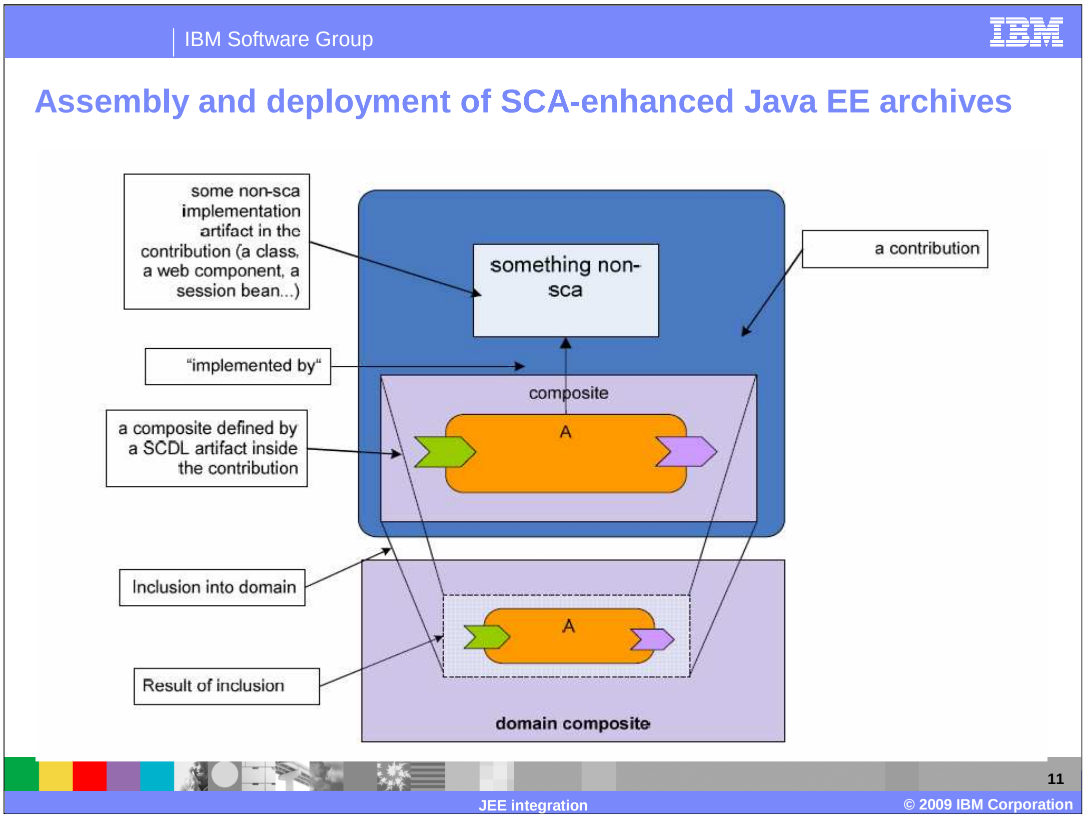
See next slide for an example.



Sample component (2)

```
<?xml version="1.0" encoding="UTF-8"?>
<componentType xmlns="http://www.osoa.org/xmlns/sca/1.0">
  <service name="AccountReporting">
    <binding.ws/>
        <interface.java interface="services.accounting.AccountReporting"/>
  </service>
</componentType>
```

In this example, assuming there is an AccountReporting service, the component type of the EAR archive (mentioned in previous slide) will expose the service, the ***AccountReporting***



IBM Software Group

Assembly and deployment of SCA-enhanced Java EE archives

some non-sca implementation artifact in the contribution (a class, a web component, a session bean...)

something non-sca

a contribution

“implemented by“

composite

A

a composite defined by a SCDL artifact inside the contribution

Inclusion into domain

A

Result of inclusion

domain composite

11

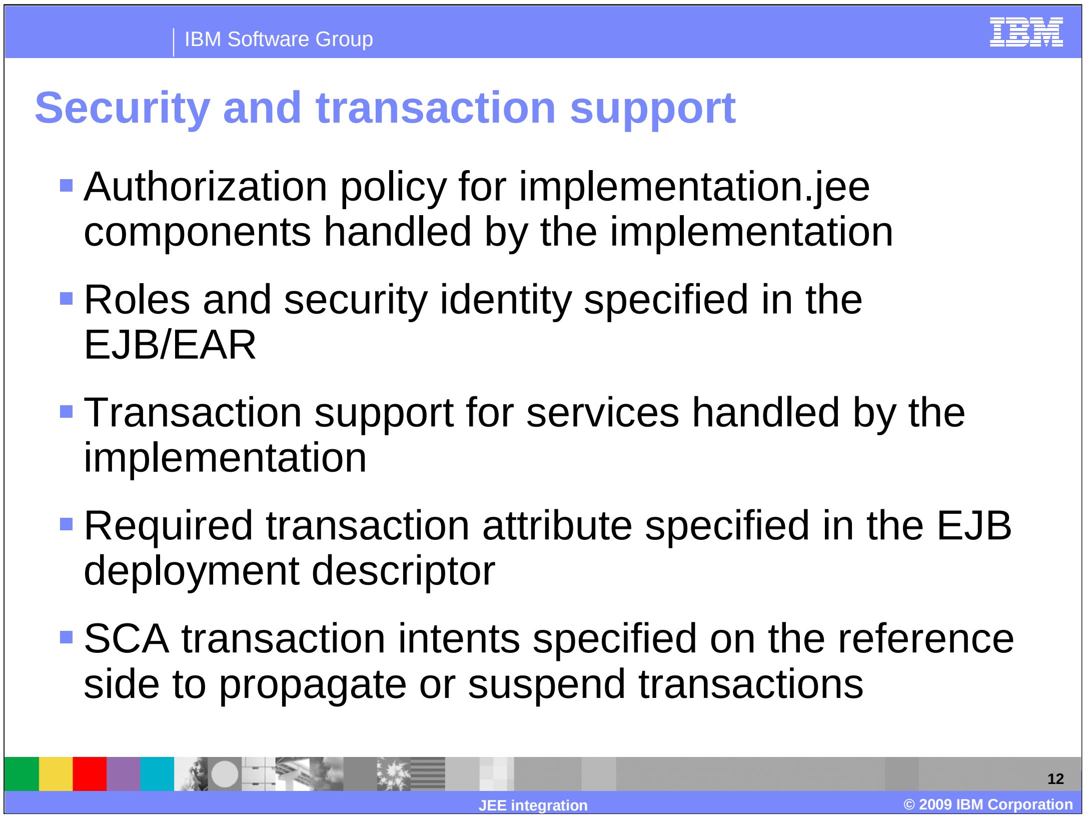
JEE integration

© 2009 IBM Corporation

This slide shows how to apply SCA assembly concepts when assembling and deploying SCA-enhanced Java EE applications.

A Java EE archive, for example a Java EE application or a Java EE module (a Web application, an ejb module), can be used as an SCA contribution.

[Recap: A contribution (contribution.xml) is a special xml document that describes how composites should be deployed. It contains information regarding which composites in the deployable jar are executable, what their namespaces are and which can be used by other contributions.]



IBM Software Group

Security and transaction support

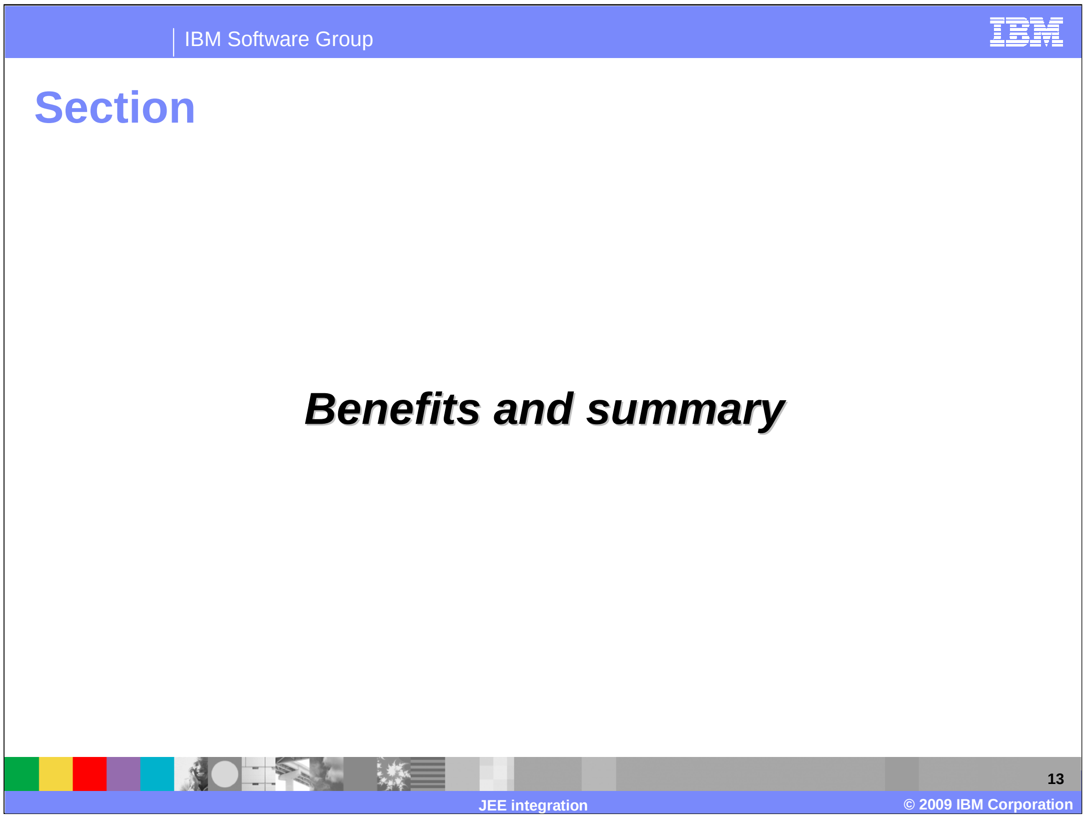
- Authorization policy for implementation.jee components handled by the implementation
- Roles and security identity specified in the EJB/EAR
- Transaction support for services handled by the implementation
- Required transaction attribute specified in the EJB deployment descriptor
- SCA transaction intents specified on the reference side to propagate or suspend transactions

12

JEE integration

Authorization policy for implementation.jee components is handled by the implementation. Roles and security identity should be specified in the EJB/EAR. SCA policy sets to handle authorization cannot be attached to the component.

Transaction support for services are also handled by the implementation. The required transaction attribute should be specified in the EJB deployment descriptor. SCA transaction intents can be specified on the reference side to propagate or suspend transactions.



IBM Software Group

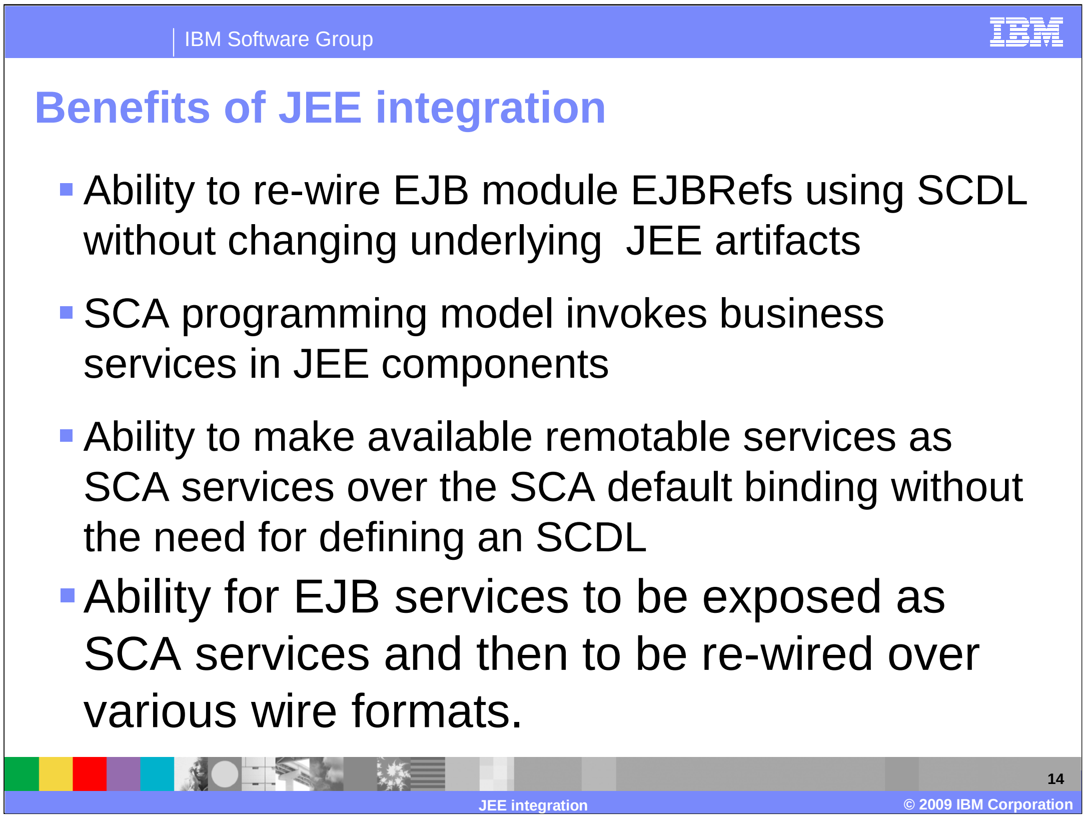
Section

Benefits and summary

13

JEE integration

To summarize



IBM Software Group

Benefits of JEE integration

- Ability to re-wire EJB module EJBRefs using SCDL without changing underlying JEE artifacts
- SCA programming model invokes business services in JEE components
- Ability to make available remotable services as SCA services over the SCA default binding without the need for defining an SCDL
- Ability for EJB services to be exposed as SCA services and then to be re-wired over various wire formats.

14

JEE integration

What are the benefits of JEE integration?

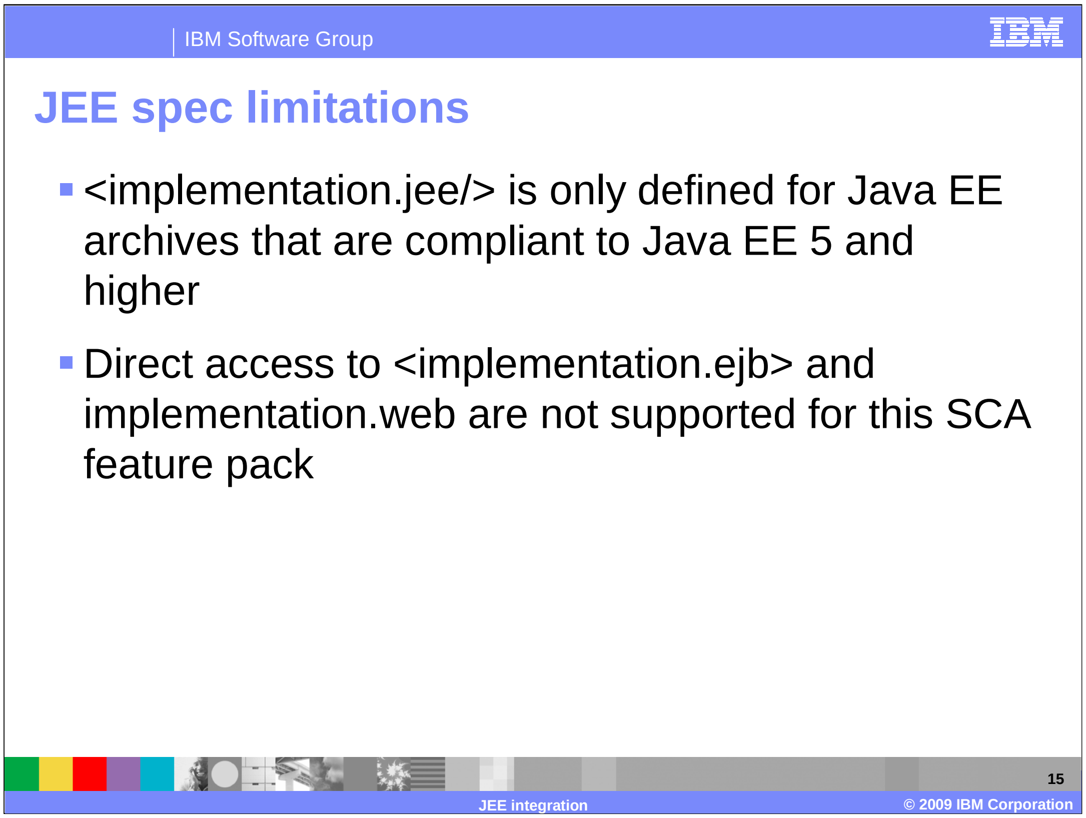
- *Ability to* reuse JEE components and modules as SCA implementations *leveraging* JEE business logic in SCA assemblies.

- *Ability for* SCA components to be exposed as services when those services exist in an EAR simplifying application packaging experience when using both SCA and JEE.

- *Ability to* rewire EJB Module EJBRefs using Service Component Definition Language (SCDL) without changing the underlying JEE artifacts *making it possible to* rewire existing JEE applications without changing them.

- *Ability to* take advantage of the SCA programming model to invoke business services in JEE components *allowing for* SCA domain ease-of-use characteristics to access those services.

- *Ability to* make available remotable services as SCA services over the SCA default binding without the need for defining a SCDL. This *allows for* existing JEE services to be wired to from SCA applications over the default binding. Thus leveraging the ease-of-use characteristics of the SCA domain without the need for writing specific SCDL.



JEE spec limitations

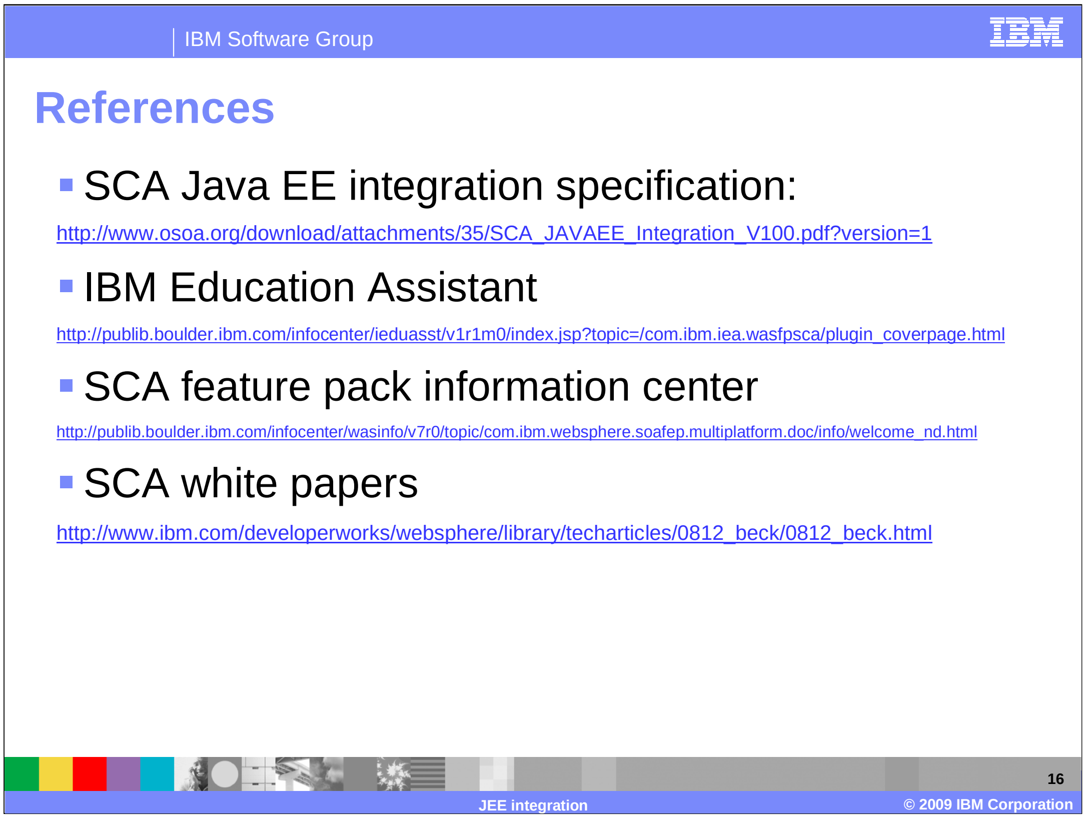
- <implementation.jee/> is only defined for Java EE archives that are compliant to Java EE 5 and higher
- Direct access to <implementation.ejb> and implementation.web are not supported for this SCA feature pack

Various parts of this specification are limited with respect to what version of Java EE specifications they

refer and apply to.

- <implementation.jee/> is only defined for Java EE archives that are compliant to Java EE 5 and higher

- Direct access to <implementation.ejb> and implementation.web are not supported for this SCA feature pack



References

- SCA Java EE integration specification:

http://www.osoa.org/download/attachments/35/SCA_JAVAEE_Integration_V100.pdf?version=1

- IBM Education Assistant

http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.wasfpsca/plugin_coverpage.html

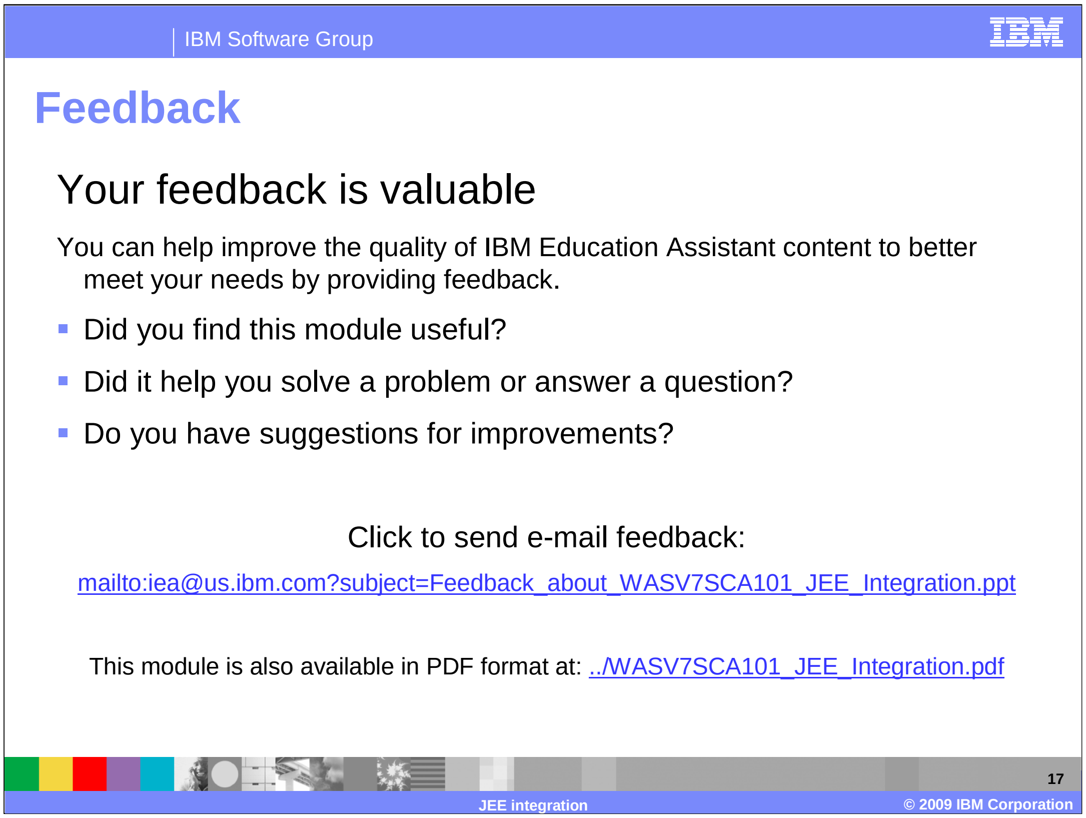
- SCA feature pack information center

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.soafep.multiplatform.doc/info/welcome_nd.html

- SCA white papers

http://www.ibm.com/developerworks/websphere/library/techarticles/0812_beck/0812_beck.html

References



Feedback

Your feedback is valuable

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

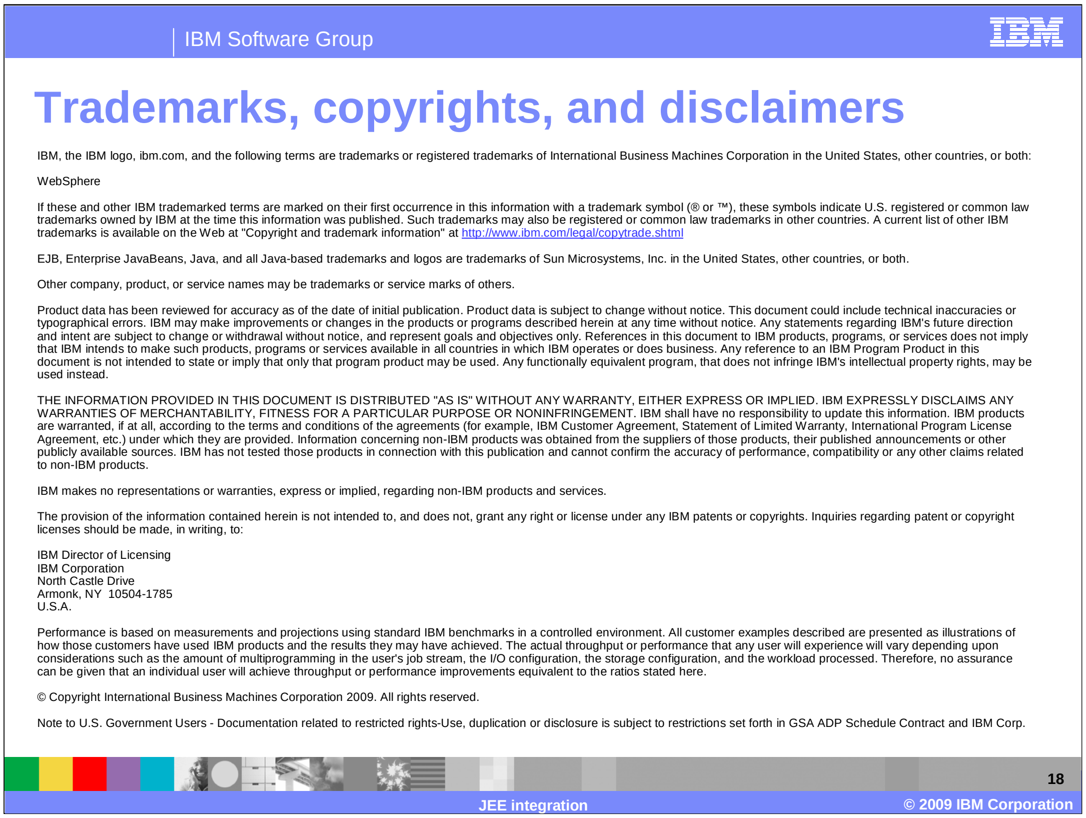
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WASV7SCA101_JEE_Integration.ppt

This module is also available in PDF format at: ../WASV7SCA101_JEE_Integration.pdf

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



Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

WebSphere

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

EJB, Enterprise JavaBeans, Java, and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.