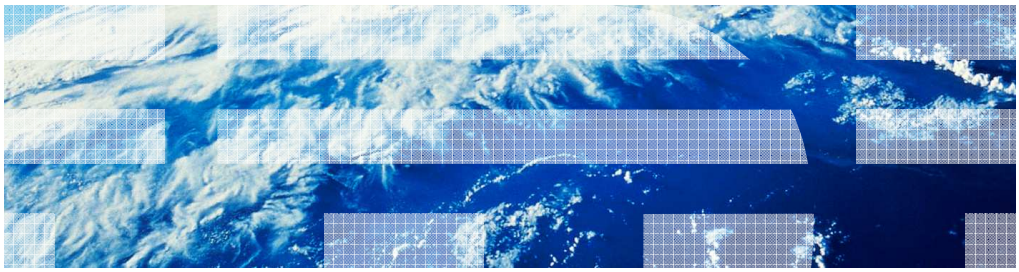
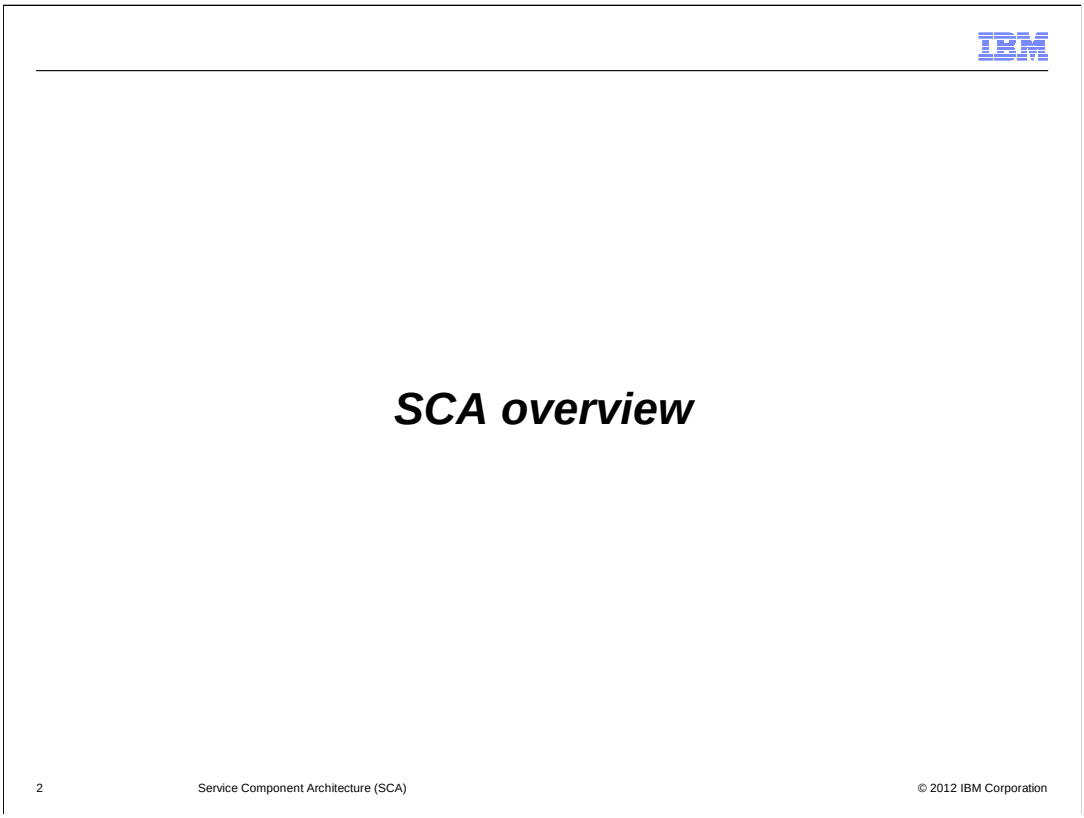
IBM WebSphere Application Server 8.5

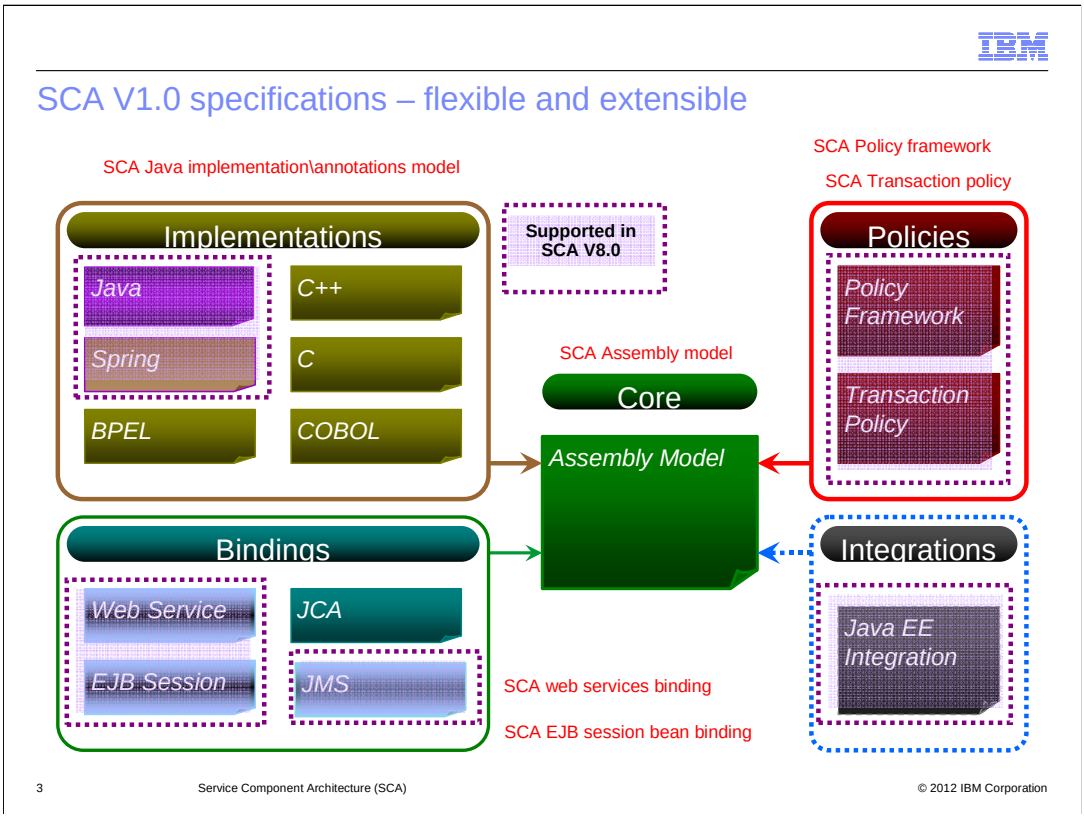
Service component architecture



This presentation describes support for Service Component Architecture (SCA) included in IBM WebSphere Application Server V8.5.



This section will cover the Service Component Architecture SCA support for OASIS 1.1 specifications overview.



SCA version 1.0 has specifications which allow for flexibility and extensibility when developing SCA applications. There are specifications in each group shown.

The assembly model specifies a component assembly model. The assembly model specifies how to define the structure of composite applications.

The policies group consists of two specifications: policy framework and transaction policy. These specify how to add infrastructure services to solutions. Some examples include, security, transactions, and reliable messaging. The implementations model supported in WebSphere SCA include the two Java specifications. It specifies how to write business services in Java, Spring, and Composite. The bindings group has specifications for how to wire services together. It specifies web services, EJB 2.0 and 3.0, and default binding specifications.

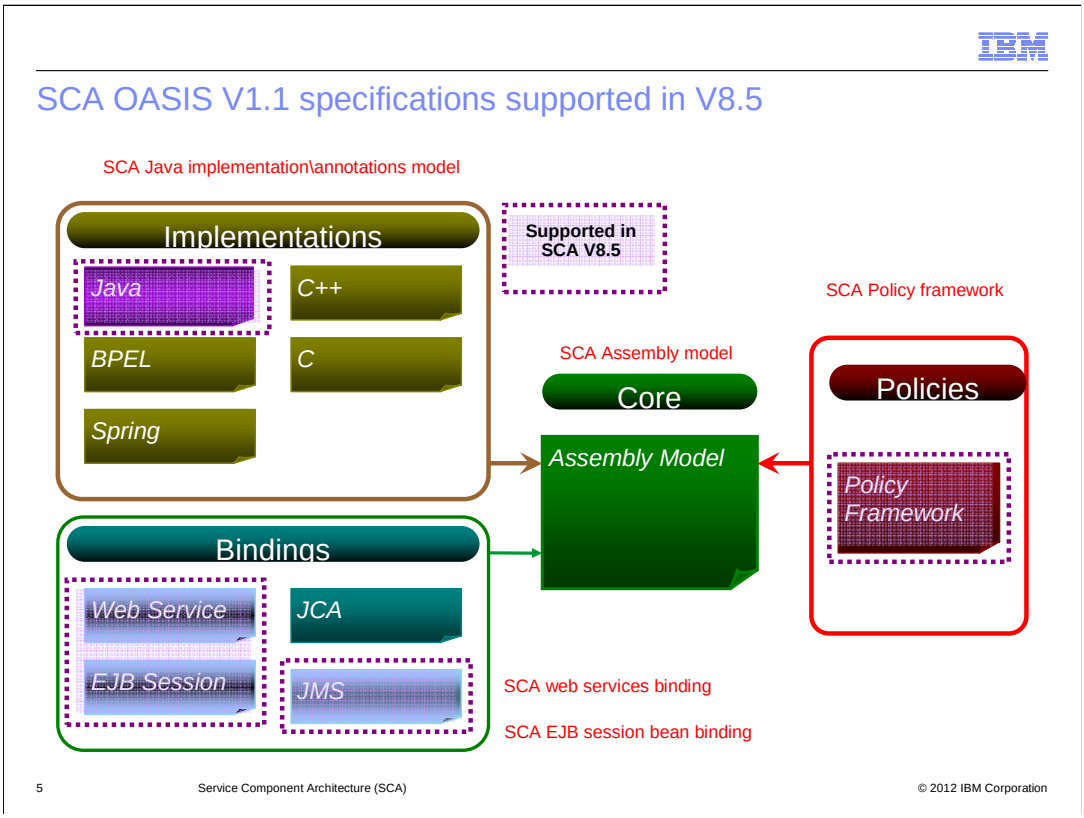
The integrations model includes Java EE integration in SCA.

Service component architecture (SCA) OASIS specifications

- Support for developing SCA applications based on the OASIS 1.1 specifications
  - OASIS SCA Assembly Model Specification V1.1
  - OASIS SCA Policy Framework Specification V1.1
  - OASIS SCA-J Common Annotations and APIs Specification V1.1
  - OASIS SCA-Bindings specifications:
    - SCA JMS Binding Specification V1.1
    - SCA web Service Binding Specification V1.1
    - SCA EJB Binding Specification V1.1
  - OASIS SCA-BPEL Specification V1.1
  - OASIS SCA-C-C++ Specification V1.1
  - OASIS SCA-Spring Specification V1.1

SCA OASIS V1.1 is a set of specifications which allow for flexibility and extensibility when developing SCA applications. The earlier OSOA SCA 1.0 specification was the creation of an informal industry collaboration. The Organization for the Advancement of Structured Information Standards (OASIS) is a global consortium which has formalized SCA into a set of standards.

The OASIS consortium has published the following SCA specifications: OASIS SCA Assembly Model Specification V1.1, OASIS SCA Policy Framework Specification V1.1, OASIS SCA-J Common Annotations and APIs Specification V1.1, OASIS SCA-BPEL Specification V1.1, OASIS SCA-C-C++ Specification V1.1, OASIS SCA-Spring Specification V1.1,and OASIS SCA-Bindings specifications. OASIS SCA-Bindings specifications consist of SCA JMS Binding Specification V1.1, SCA web Service Binding Specification V1.1, and SCA EJB Binding Specification V1.1.



SCA V8.5 supports these OASIS V1.1 Specifications: OASIS SCA Assembly Model Specification V1.1, OASIS SCA Policy Framework Specification V1.1, OASIS SCA-J Common Annotations and APIs Specification V1.1, and OASIS SCA-Bindings specifications. OASIS SCA-Bindings specifications consist of SCA JMS Binding Specification V1.1, SCA Web Service Binding Specification V1.1, and SCA EJB Binding Specification V1.1. The OASIS SCA Policy Framework Specifications V1.1 support does not include OASIS policy set definitions but it includes WebSphere Application Server-centric policy attachments.

SCA OASIS support in V8.5

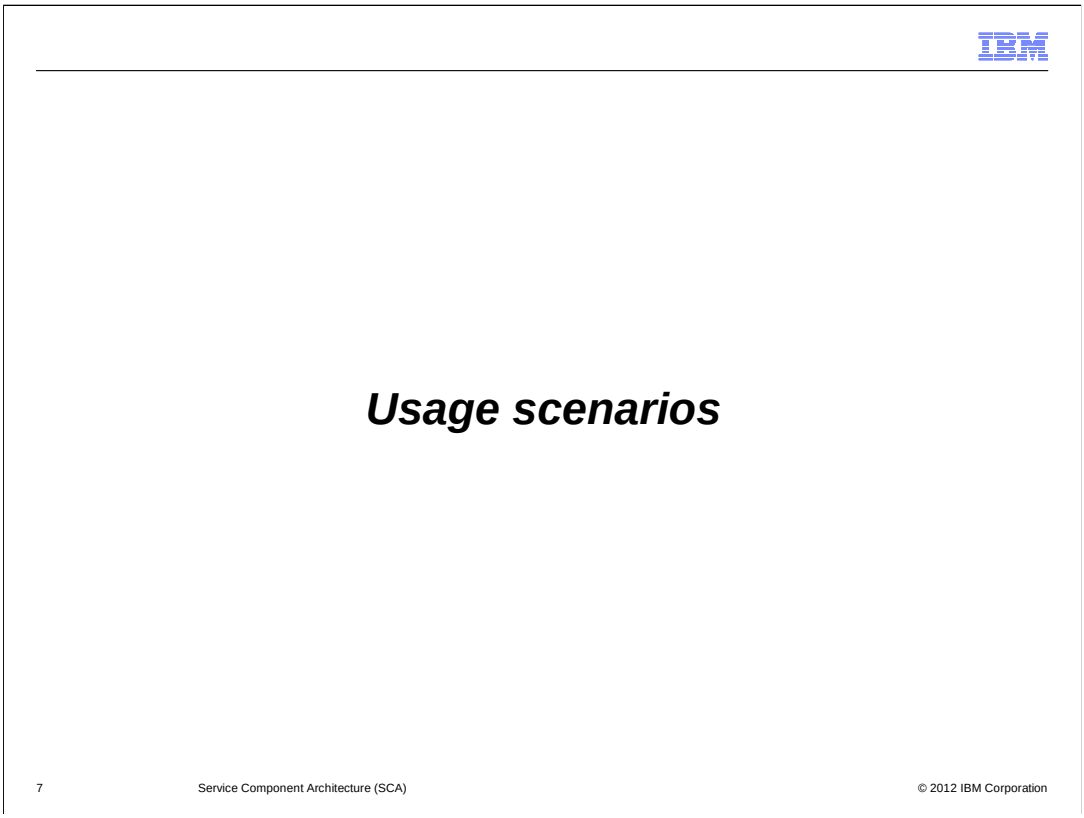
- IBM WebSphere Application Server V8.5 now supports deploying, administering and running applications based on OASIS 1.1 specifications
- Support for POJO, JAXB and SDO data types
- WebSphere Application Server does not support OASIS policy set definitions, however WebSphere Application Server-centric policy attachment is supported
- Support for Asynchronous invocation over SCA binding
- OSOA/OASIS interoperability supported over SCA and WS binding
- Support for SCA Client API

SCA OASIS in V8.5 supports data types such as POJO, JAXB and SDO.

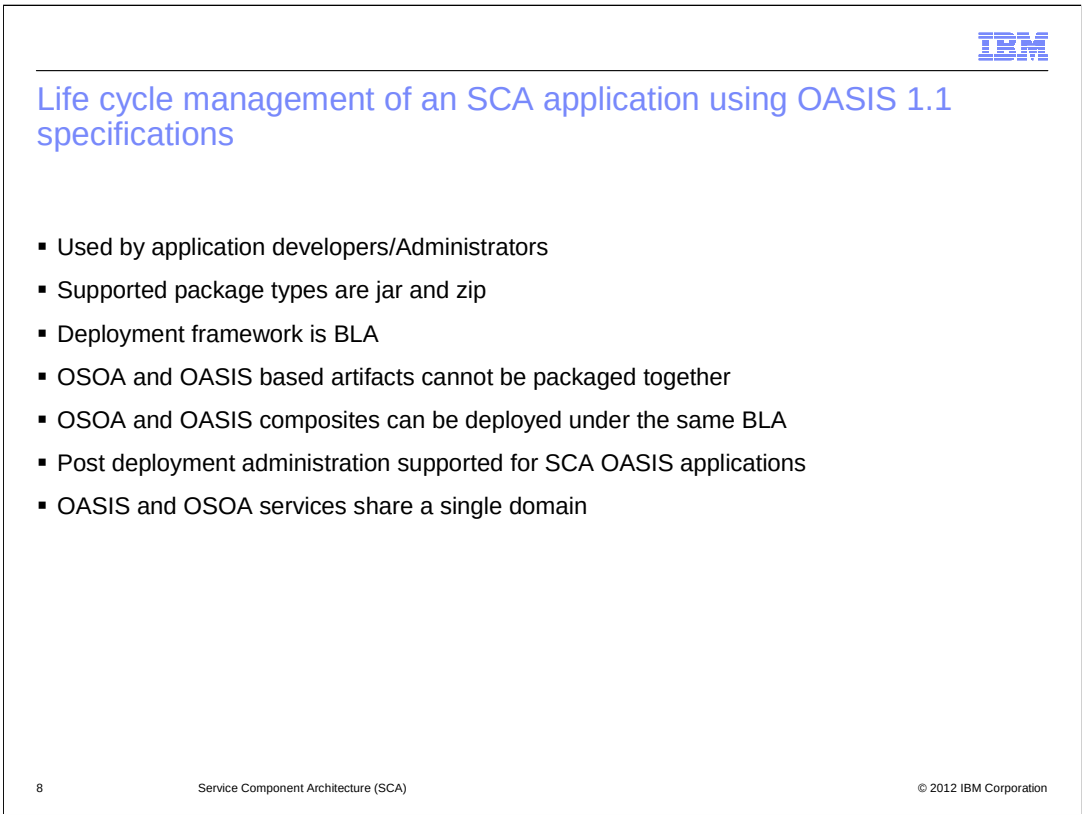
SCA OASIS specifications supports the asynchronous running of request-response services. This enables a client thread to continue doing other work while the service runs. You can use SCA OASIS annotations and APIs in Java interfaces to enable synchronization in services.

Interoperability between OSOA and OASIS applications is supported such that you can wire OSOA components to OASIS components and vice-versa using the supported implementation types and can be wired over default binding and web services bindings.

SCA Client API is supported for OASIS. You can use this API (from a web application for example) to get a reference to a service implementation and then invoke it’s methods.



Service Component Architecture (SCA) OASIS support is used in these scenarios.



As an Application developer/Administrator, you can compose, deploy and administer an SCA application using OASIS 1.1 specifications.

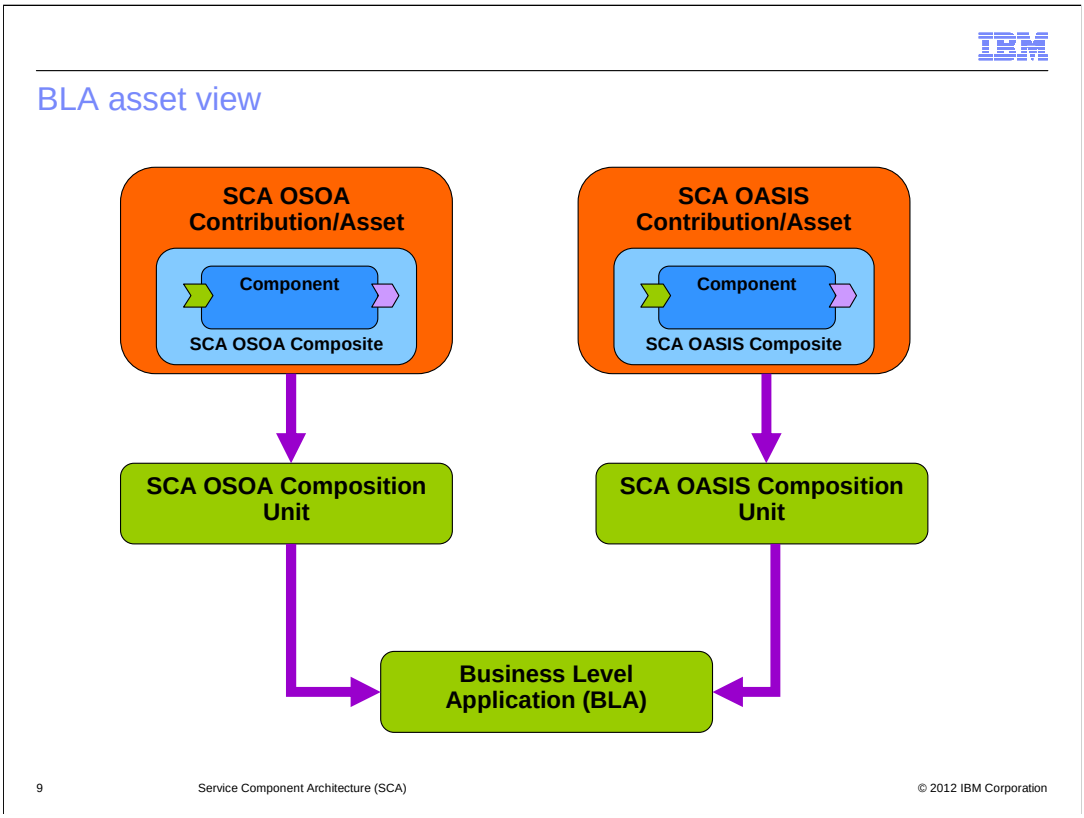
All of the existing support for SCA OSOA applications such as the supported package types, the deployment framework, the post deployment administration for view/edit, and in essence the full application life cycle management have been extended to SCA OASIS applications.

The SCA OASIS application can be packaged either as a jar or a zip contribution which is known as an asset. A deployable composite from this asset can then be deployed as a composition unit under a BLA, using either WSADMIN command or the administrative console.

Multiple composition units deployed from SCA OSOA asset and SCA OASIS asset can be deployed under a BLA.

An SCA asset can either be totally based off of OSOA specifications or OASIS specifications. They are mutually exclusive.

OASIS contributions are deployed to the same SCA domain/cell as the OSOA contributions. Hence the OASIS and OSOA services share a single domain allowing for interoperability to work by way of default binding.

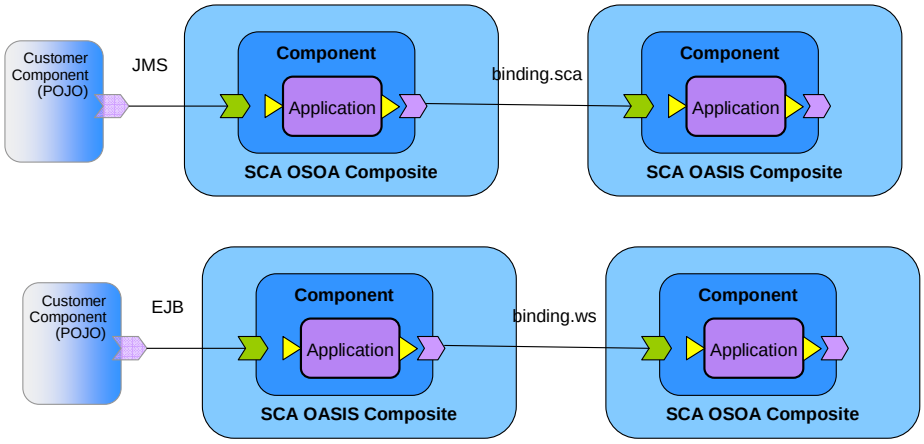


This slide presents a graphical representation of BLA, composition units and SCA OSOA and OASIS contributions.

The SCA OSOA and OASIS contributions are mutually exclusive and their artifacts cannot be mixed and matched. Each OSOA and OASIS composite is deployed as a separate Composition Unit. As BLA is an administrative unit, multiple composition units representing OSOA and OASIS composites can be deployed under a single BLA.

Interoperability between OSOA and OASIS applications

- Used by application developers migrating their OSOA applications to OASIS applications



You can wire OASIS and OSOA SCA components together when both SCA composites are running in a single product cell. Only remote interfaces can be used. Interfaces must be annotated in a way consistent with the run time in which they are deployed. OASIS interfaces marked with the asyncInvocation intent cannot interoperate with OSOA services. The product does not perform interface or policy matching. Incompatibility will result in runtime exceptions.

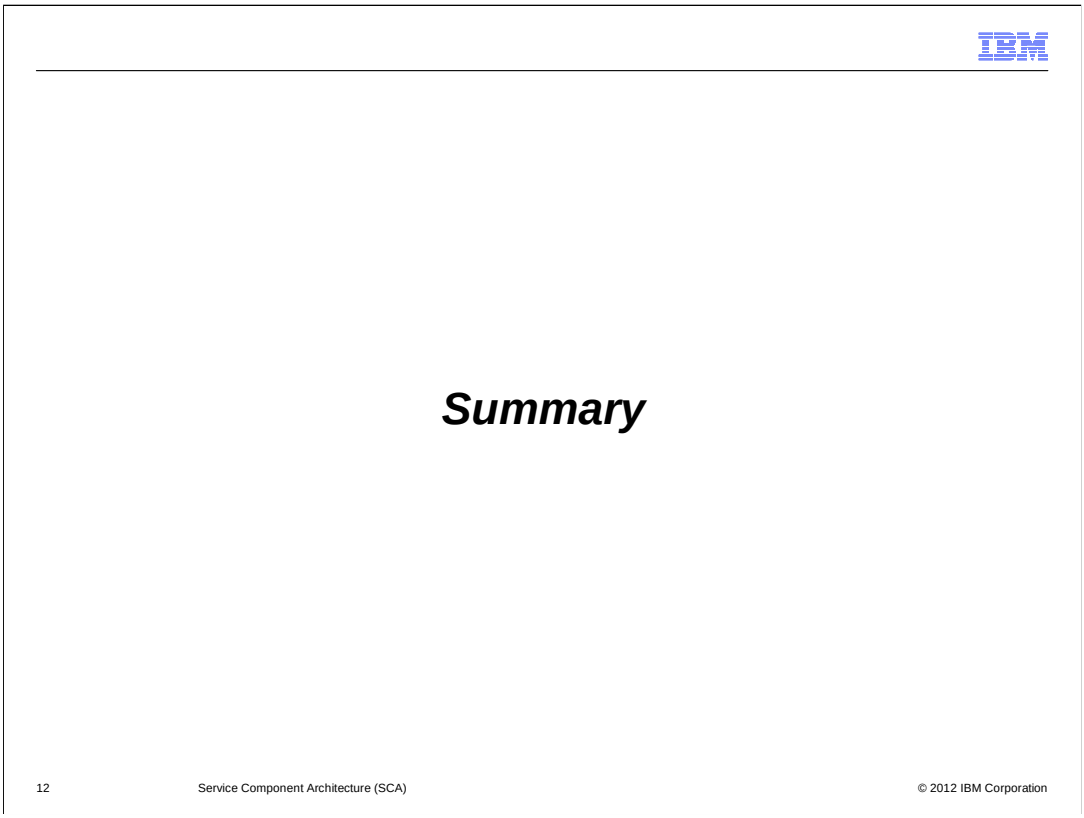
Key differences between OSOA and OASIS specifications SCA assembly model

Type	OSOA	OASIS
Namespace	http://www.osoa.org/xmlns/sca/1.0	http://docs.oasis-open.org/ns/opencsa/sca/200912
XSD		Extensibility removes the occasional UPA issue. The sca:extensions elements are in various places.
composite.xml	Reference targets *component*/*service* put non-configured bindings on the reference.	Reference targets *component*/*service*/*binding* do not put non-configured bindings on the reference. The target string identifies them and the configuration is pulled from the service.
Asynchronous invocation	Not supported	Asynchronous invocation is supported. Activate it by including the asyncInvocation intent on an interface (or service or reference).
Conversations	Supported	Not supported
Operation configuration	Supported in services or references	Not supported in services or references
Interface	Cannot be marked removable from within the composite	Can be marked as removable from within the composite
Wire format		Bindings can have a wireFormat child element
Operation selector		Bindings can have an operationSelector child element
Wires		New replace semantics
Domain level		References and services are ignored
definitions.xml	Has binding element	Does not have binding element

The fundamental difference between an OSOA application and an OASIS application is the namespace. The quickest and easiest way to identify if an application is based on OSOA specifications or OASIS specifications is by checking for these notable differences as outlined in the table. The definitions.xml no longer includes a binding element in OASIS applications. In OASIS new optional binding element children have been added: wireFormat and operationSelector.

There are other differences in Java API and Annotations which are minor. There are also differences in JMS Binding specifications which are not included in the table.

Standardized support for SCA Client API in OASIS is now available, unlike in OSOA. This API can be used to get a reference to a service implementation by way of it's method getService(). A web application is a good example where this API is very useful to invoke SCA services without the WAR being deployed as an SCA composite within the SCA domain.



This section contains a summary.

Summary

- Support for OASIS specifications 1.1 limited to certain implementation and binding types
- Same life cycle management for both OSOA and OASIS applications
- Some specification level differences between OSOA and OASIS
- There is no migration support to migrate OSOA applications to OASIS specifications

SCA in WebSphere Application Server V8.5 has the same support for the OSOA specification that was provided in WebSphere Application Server V8.0 and contains new support for OASIS specifications. The OASIS support is limited to certain implementation and binding types. The same life cycle management applies for both OSOA and OASIS applications OSOA applications easily migrate to OASIS specifications as the differences between them are minimal. SCA V8.5 does not provide any migration utilities for customers to migrate OSOA applications to OASIS specifications.

References

- SCA Overview
  - http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-nd-mp&topic=csca_overview
- DeveloperWorks
  - http://www.ibm.com/developerworks/websphere
- OASIS Open CSA website for SCA V1
  - http://www.oasis-opencsa.org/sca

See these references for additional information about Service Component Architecture (SCA) OASIS 1.1 support.

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 email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WAS_V85_SCA.ppt

This module is also available in PDF format at: ../WAS_V85_SCA.pdf

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

Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. 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

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.