



WebSphere® and CICS Transaction Server

The future of SOA on z/OS built on a smarter foundation of CICS TS V4, WAS V7 and Rational Developer for System z

Steve Kinder – SOA Foundation Architect

Ian Mitchell – Product Architect for CICS Transaction Server

Cindy Krauss – Product Architect, Rational Developer for System z

April 8, 2009



Welcome...

- Your speakers today:
 - Steve Kinder – SOA Foundation Architect
 - Ian Mitchell – Product Architect for CICS Transaction Server
 - Cindy Krauss – Product Architect, Rational Developer for System z
- What you will hear in this session:

*Building on the lessons learned in **Future-Proof your Applications with SCA: Programming model optimized for SOA** we will show you how SCA can be applied to your business applications running on z/OS.*

Using a simple application example we will show components wired across CICS and WebSphere Application Server (WAS) that will be used as an illustration of how CICS, Rational Developer for System z (RDz), and WebSphere Application Server for z/OS are delivering new capabilities in support of this important multi-lingual, multi-environment programming model.



Agenda

- Quick SCA in WAS recap
- Introduction to SCA support in CICS TS v4.1
- The Portfolio Scenario
- Defining and implementing SCA components using Rational Developer for System z
- Deploying and running SCA components in CICS TS
- Summary and Questions



Agenda

- **Quick SCA in WAS recap**
- Introduction to SCA support in CICS TS v4.1
- The Portfolio Scenario
- Defining and implementing SCA components using Rational Developer for System z
- Deploying and running SCA components in CICS TS
- Summary and Questions



SCA: What it is

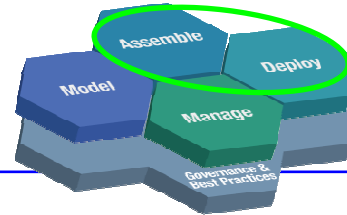
- Service Component Architecture.
- A concrete manifestation of an SOA way of thinking.
- Designed for building agile service oriented applications.
- A framework for implementing, assembling, composing and deploying services.
- Supports loose or tight coupling of coarse or fine grained services.
- Extends, exploits and complements existing technologies and standards.
- Language, Application Environment, Framework and Vendor neutral.
- Supports Java and Web Services, and more
- An extensible set of:
 - Protocol bindings (eg. SCA, WS, RMI, ...)
 - Implementation languages (eg. Composite, Java, ...)
 - Interface definitions (eg. WSDL, Java, ...)
 - Pluggable Data bindings (eg. PoJo, JAXB, ...)
 - Policies and Intents (eg. Integrity, Confidentiality).
- “Classic SCA” refers to Service Component Architecture as it is defined and built by IBM supported in a variety of WebSphere Family products starting with V6.
- “Open SCA” refers to Service Component Architecture as defined by the industry at both the OSOA collaboration



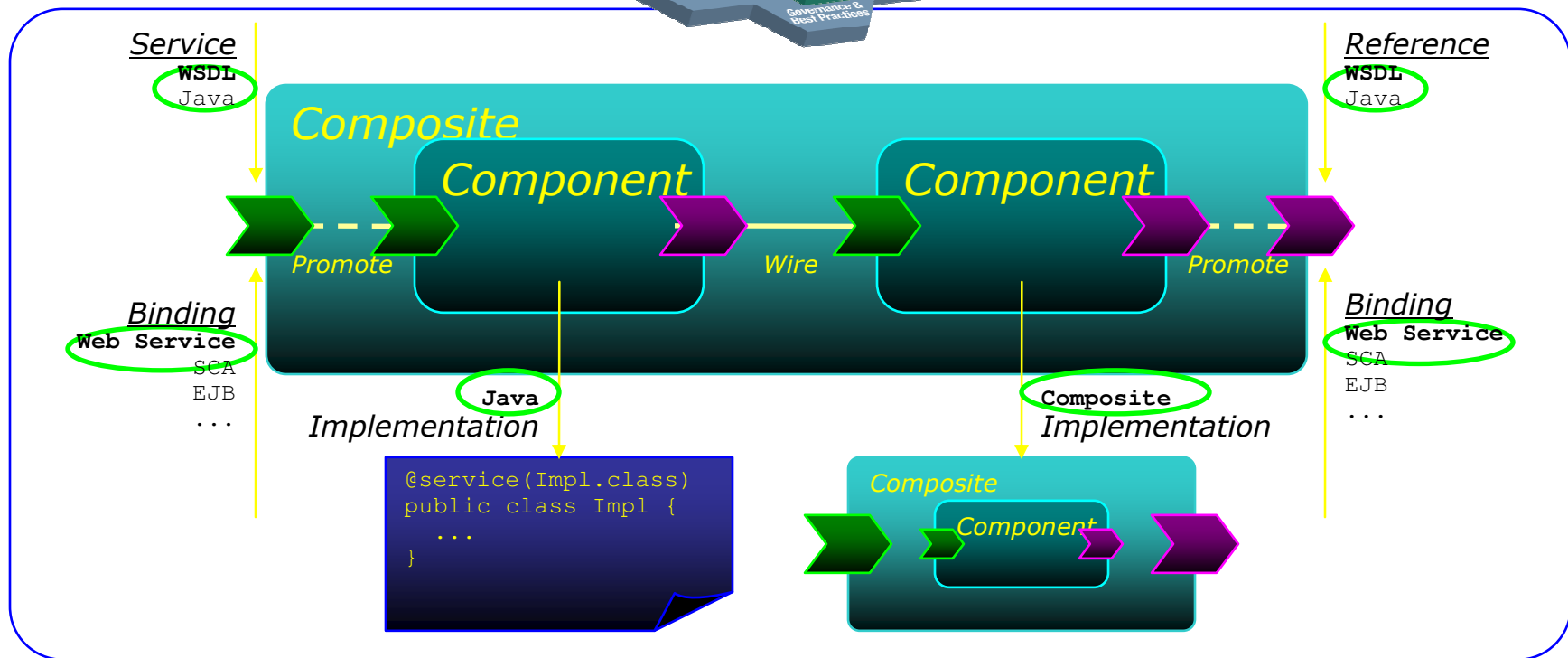
SCA Key Concepts

Design → Implement → Compose → Run → Test

WebSphere software
WebSphere Application Server v7.0 + SCA1.0



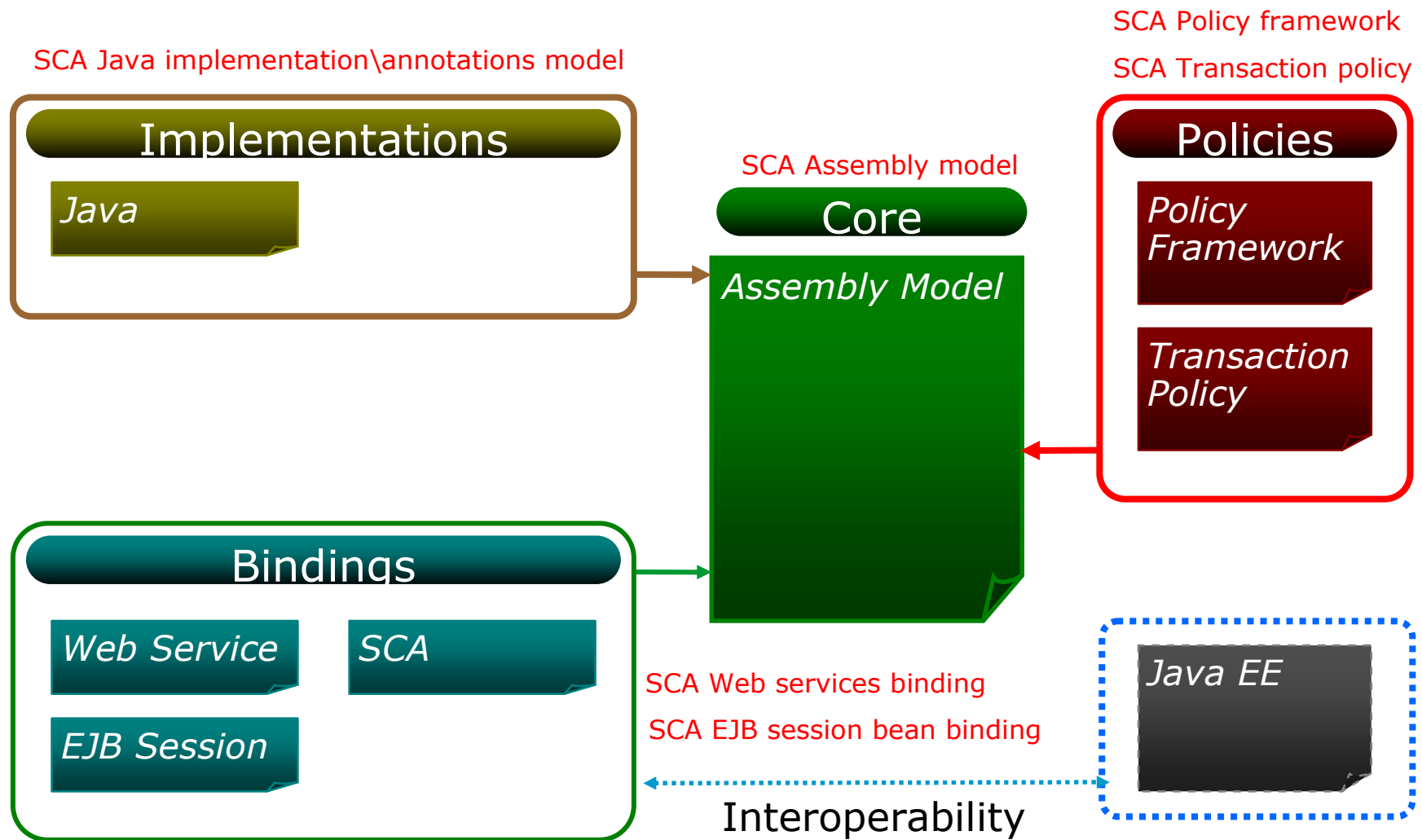
Domain



Reusability, Connectivity, Flexibility, Extensibility



SCA v1.0 Specifications – Flexible & Extensible





Agenda

- Quick SCA in WAS recap
- **Introduction to SCA support in CICS TS v4.1**
- The Portfolio Scenario
- Defining and implementing SCA components using Rational Developer for System z
- Deploying and running SCA components in CICS TS
- Summary and Questions



CICS Transaction Server v4.1 allows you to:

Compete for new opportunity by gaining insight into business processes and responding by modifying key business applications quickly and with confidence

– *Business Flexibility and Innovation*

Comply with corporate, industry and government policies to manage business risk of critical business applications

– *Governance and compliance*

Control costs by simplifying IT infrastructure and improving development and operations productivity through easier-to-use interfaces and functions

– *IT Simplification*

CICS TS v4.1 will be available 2nd or early 3rd quarter 2009.

Join the Open Beta – see <http://www.ibm.com/software/htp/cics/tserver/v41/openbeta/>



CICS TS v4.1: Competing with Flexibility and Innovation

- **Application Components**
 - **Bundles**
- Web 2.0 and RESTful features
 - **Atom Feeds**
- Web Services, SOAP and XML Enhancements
 - **WS-Addressing**
 - **Improved XML data mapping**
- **Java 6**
- **Event Processing**



CICS TS v4.1: Complying

- **Event Processing**
- **Meta-data for Resource Definitions**
- **Identity Propagation**
- **Security Enhancements for DB2 applications**
- **WebSphere Service Registry and Repository support**



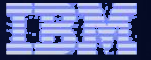
CICS TS v4.1: Controlling Costs with Simplification

- **CICS Explorer**
- **CICS Management Interface**
- **CSD Definition Repository API**
- **Discovery Library Adapter for CICS TS**
- **Dynamic Workload Management enhancements**
- **Extensions to CICS intercommunications over TCP/IP**
- **IPv6 support**
- **WebSphere MQ Group Attach**



CICS TS v4.1 Component Architecture

- Provide capability to easily develop flexible and reusable CICS application components
 - Rapid assembly and deployment of new Services
 - Express existing applications as re-usable components
- Separation of bindings from application code allows flexible infrastructure changes
- Reduce skills and effort required to view and manage business applications



Component Architecture in CICS TS v4.1

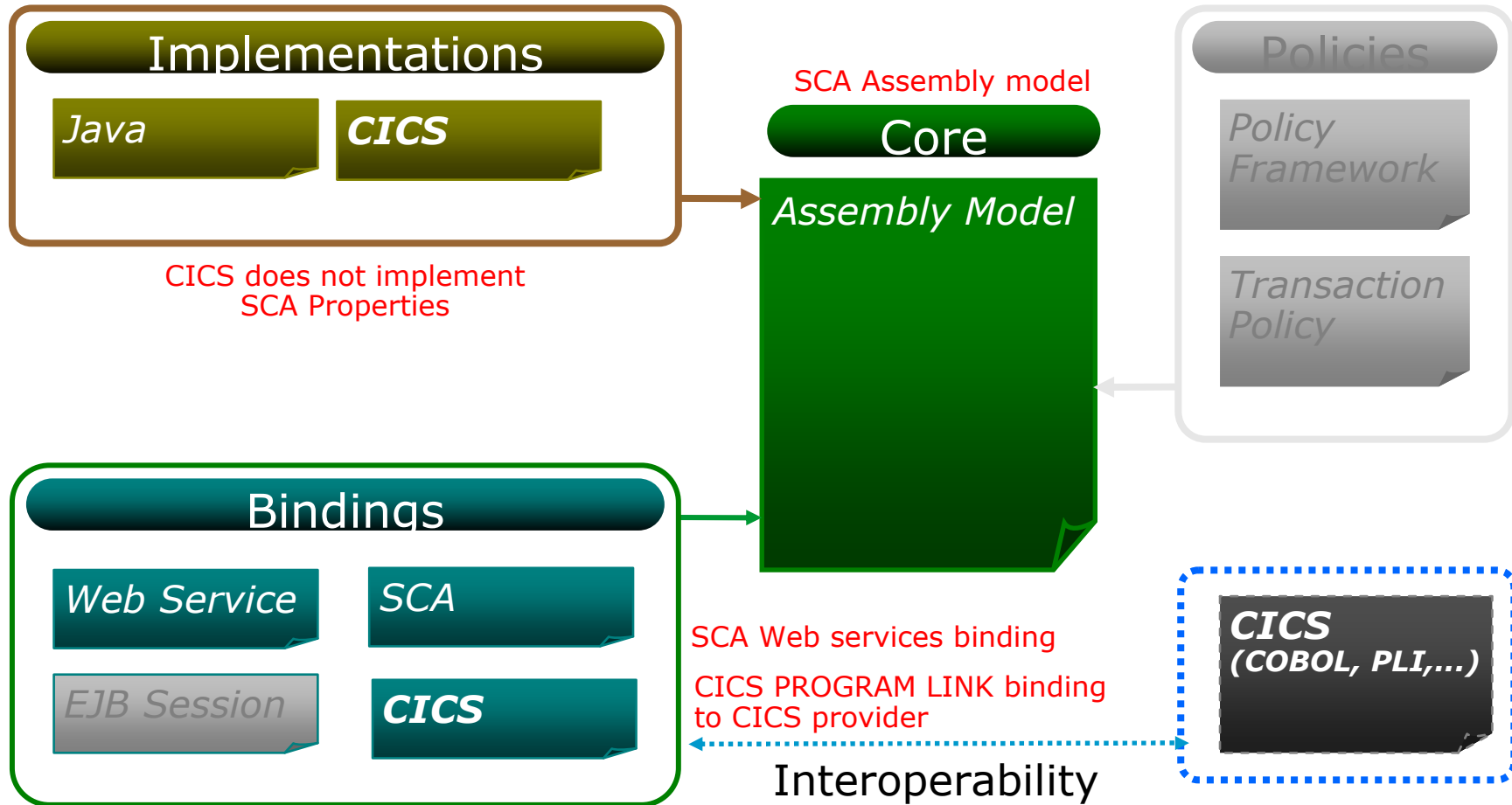
- Ability to install and manage business applications as single CICS components
 - Abstract away from programs, transactions, resources
- Ability to describe CICS application as SCA components (using SCDL)
- Application bindings provided by CICS and configured using SCDL
 - Services and References
 - Invocation locally and via web services
 - EXEC CICS INVOKE SERVICE
- RDz providing CICS component tooling to enable component definition, assembly and deployment



SCA v1.0 Specifications – CICS TS v4.1 capabilities

CICS Service provider/consumer model
(Channel & Container with WS-Bind data-mapping,
EXEC CICS INVOKE SERVICE command)

SCA Policy framework
SCA Transaction policy



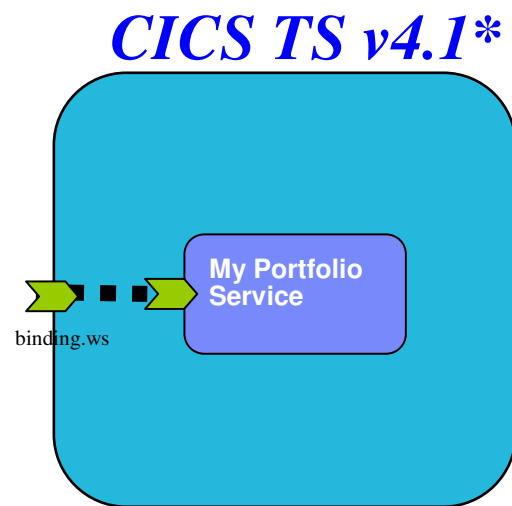


Agenda

- Quick SCA in WAS recap
- Introduction to SCA support in CICS TS v4.1
- **The Portfolio Scenario**
- Defining and implementing SCA components using Rational Developer for System z
- Deploying and running SCA components in CICS TS
- Summary and Questions



Service Component Architecture - Scenario

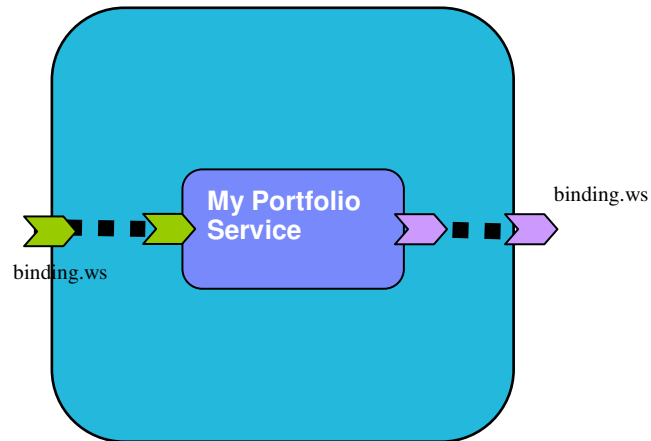


```
<component name = "MyPortfolioComponent">  
  <implementation.CICS program="PORTFOLI"/>  
  <service name="MyPortfolioService"  
    <binding.ws ... >  
  </service>  
  
</component>
```



Service Component Architecture – Scenario

*CICS TS v4.1**

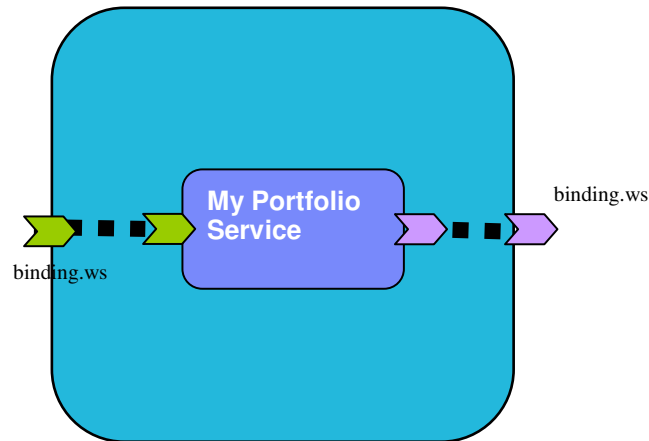


```
<component name = "MyPortfolioComponent">  
  <implementation.CICS program="PORTFOLI"/>  
  <service name="MyPortfolioService"  
    <binding.ws ... >  
  </service>  
  <reference name="MyStockQuoteService">  
    <binding.ws ... >  
  </reference>  
</component>
```



Service Component Architecture – Programming Model

*CICS TS v4.1**

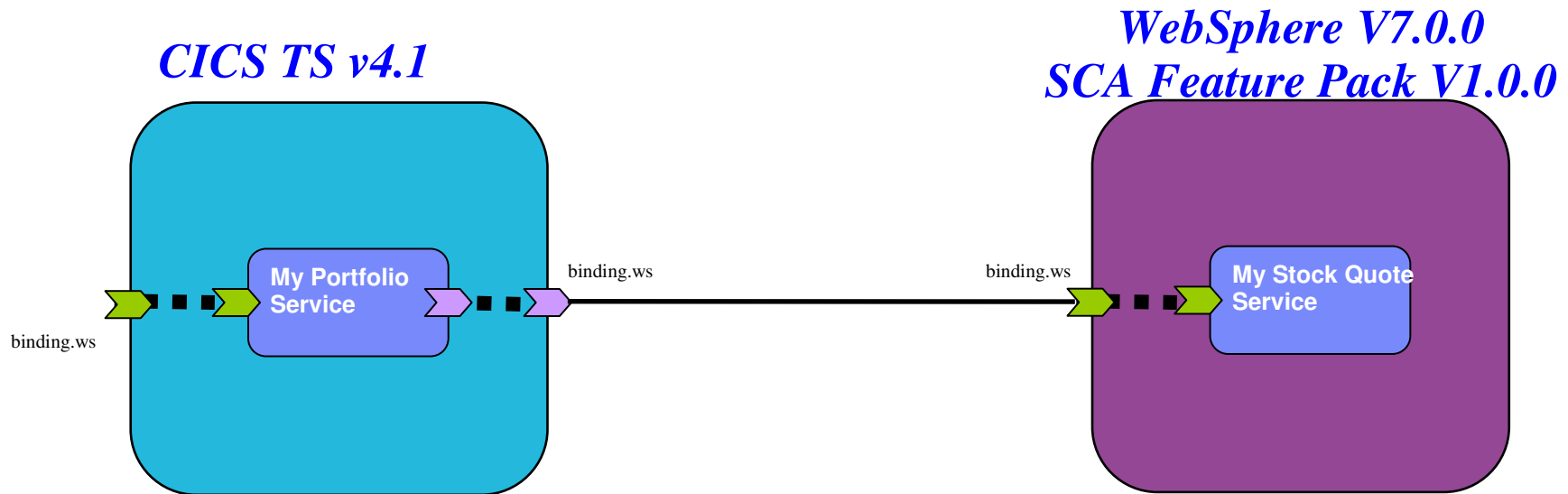


CICS COBOL Program

```
EXEC CICS PUT CONTAINER("SYMBOL")  
          CHANNEL("QUOTESERVICE")  
          FROM("IBM ")  
.  
  
EXEC CICS INVOKE SERVICE("MyStockQuoteService")  
          CHANNEL("QUOTESERVICE")
```



Service Component Architecture – Scenario

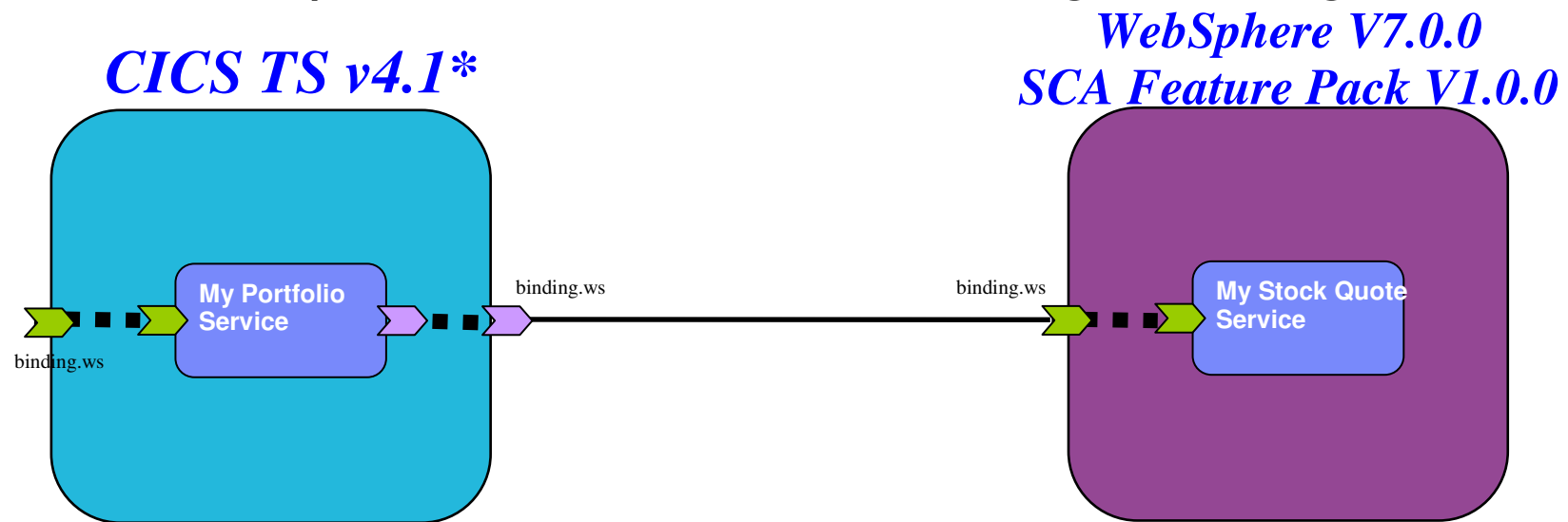


```
<component name = "MyPortfolioComponent">
  <implementation.CICS program="MyPortfolioService">
    <service name="MyPortfolioService">
      <binding.ws ... >
    </service>
    <reference name="MyStockQuoteService">
      <binding.ws ... >
    </reference>
  </component>
```

```
<component name = "MyStockQuoteComponent">
  <implementation.java
    class="mystockquoteImpl.class"/>
  <service name="MyStockQuoteService">
    <binding.ws ... >
  </service>
</component>
```



Service Component Architecture – Programming Model



CICS COBOL Program

```
EXEC CICS PUT CONTAINER("SYMBOLIC")
CHANNEL("QUOTESE")
FROM("IBM ")
.
EXEC CICS INVOKE SERVICE("MyStockQuote")
CHANNEL("QUOTESE")
```

WebSphere POJO

```
@Remotable
public interface stockQuote{
    public String myStockQuoteService();}

@Service(stockQuoteImpl.class)
public class stockQuoteImpl implements stockQuote
```



Agenda

- Quick SCA in WAS recap
- Introduction to SCA support in CICS TS v4.1
- The Portfolio Scenario
- **Defining and implementing SCA components using Rational Developer for System z**
- Deploying and running SCA components in CICS TS
- Summary and Questions

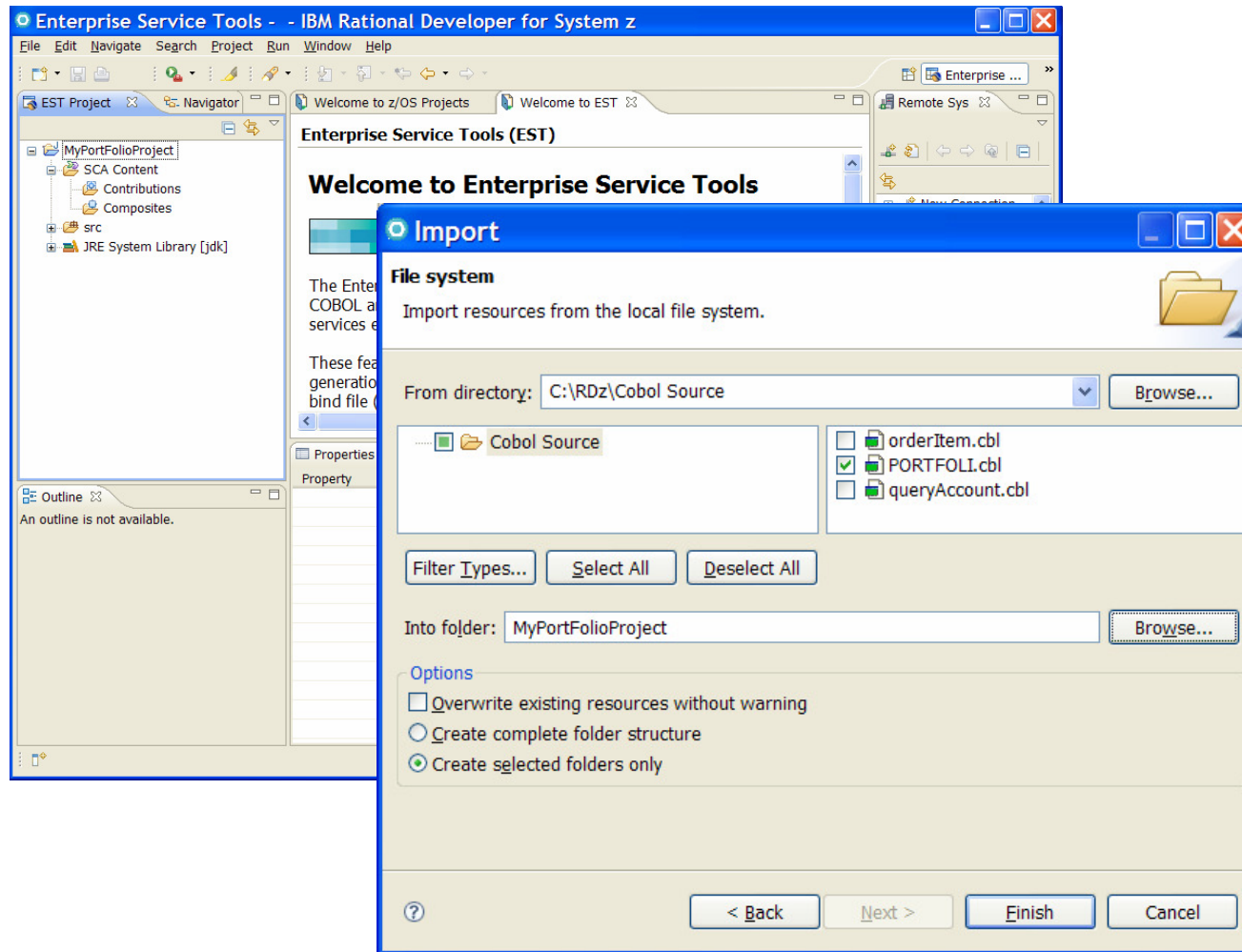


RDz SCA Tooling Support

The screenshot displays the IBM Rational Developer for System z interface. The main window shows the Enterprise Service Tools (EST) welcome page, which includes a 'New' menu. The 'New' menu is open, showing various project types, with 'SCA Project' selected. A 'New SCA Project Wizard' dialog box is overlaid on the right side of the screen. The wizard is titled 'Create a New SCA Project' and prompts the user to enter generic information about the project. The 'Project name' field contains 'MyPortFolioProject'. The 'Use default location' checkbox is checked, and the 'Location' field shows 'C:\RDz CA beta workspaces\MyPortFolioProject'. The 'Target Runtime' is set to '<none>'. Under 'Project Settings', the 'Implementation Types for SCA Components' section has checkboxes for 'Composite', 'Java', and 'CICS', all of which are checked. The wizard has 'Back', 'Next >', 'Finish', and 'Cancel' buttons at the bottom.



RDz SCA Tooling Support





RDz SCA Tooling Support

The screenshot displays the IBM Rational Developer for System z interface with the 'Enterprise Service Tools (EST)' project open. The 'New CICS Component Type Wizard' is active, showing the following configuration:

- Project:** MyPortFolioProject
- Component type file name:** PORTFOLI.componentType
- Component type service properties:**
 - CICS program source file: PORTFOLI.cbl
 - Program name: PORTFOLI
 - Conversion type: Interpretive XML Conversion

The 'Language structures' dialog box is also open, showing the 'Response Language Structure' tab. It lists the following language structures for the response message:

- MYVARIABLES
- CustomerInfo
 - userName
 - department
 - itemNumber
 - returnCode
- Accountinfo
- errormsg
- DFHCOMMAREA (contains unsupported types)

Buttons for '< Back', 'Next >', and 'Finish' are visible at the bottom of both dialog boxes.



RDz SCA Tooling Support

The screenshot displays the IBM Rational Developer for System z (RDz) interface with the Enterprise Service Tools (EST) project. The 'New CICS Component Type Wizard' is open, showing the configuration for a new component type named 'PORTFOLI.componentType' within the 'MyPortFolioProject'.

New CICS Component Type Wizard Configuration:

- Project:** MyPortFolioProject
- Component type file name:** PORTFOLI.componentType
- Component type service properties:**
 - CICS program source file: PORTFOLI.cbl
 - Program name: PORTFOLI
 - Conversion type: Interpretive XML Conversion

The wizard includes navigation buttons: < Back, Next >, and Finish.

New CICS Component Type Wizard - DFHLS2WS: High Level Language to WSDL Conversion

Specify targets for WSBIND and WSDL files

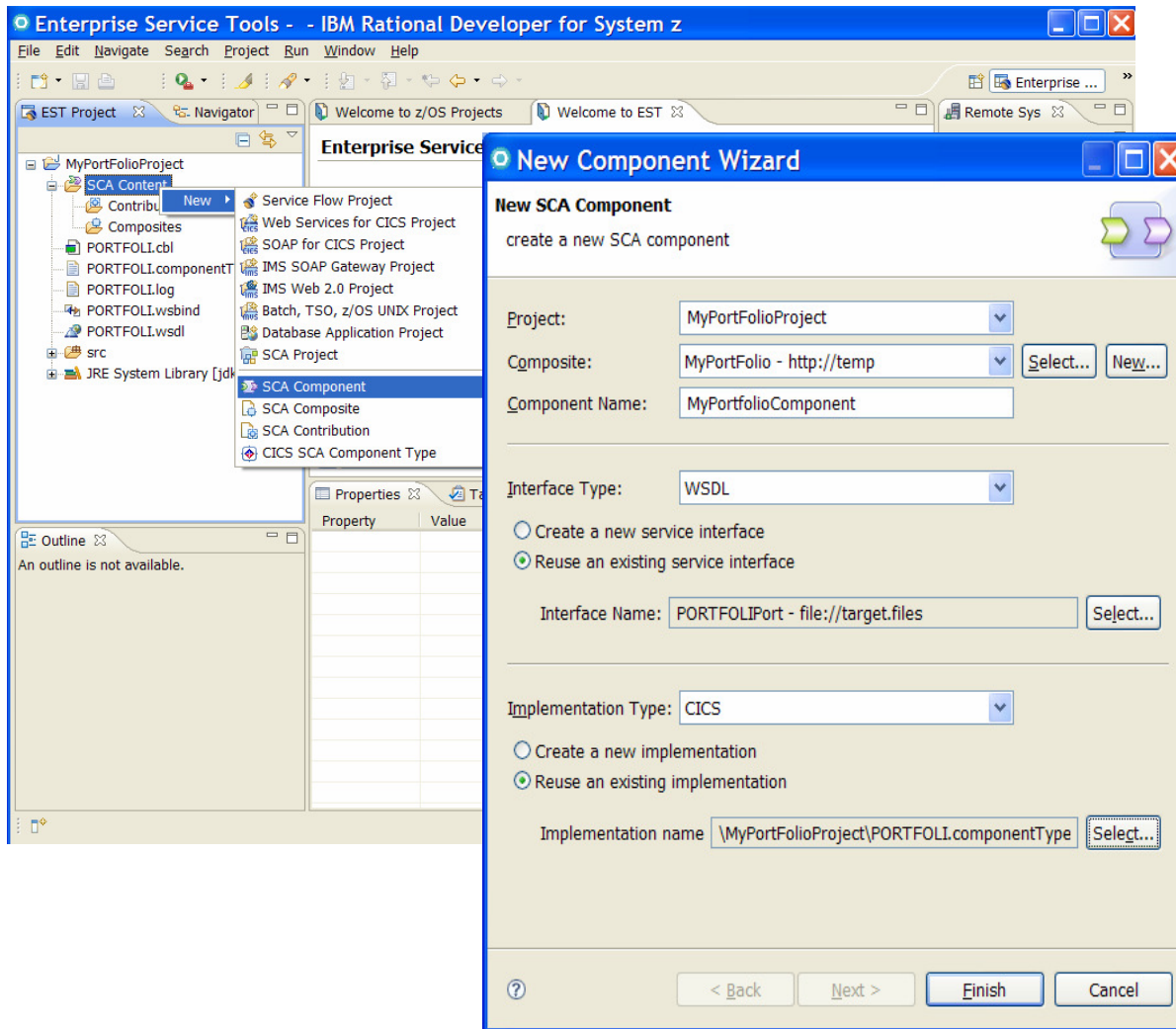
Service Artifacts

- File container: /MyPortFolioProject (Browse...)
- WSDL file name: PORTFOLI .wsdl
- WSBIND file name: PORTFOLI .wsbind
- Log file name: PORTFOLI .log
- Overwrite files

Navigation buttons: < Back, Next >, Finish, Cancel.

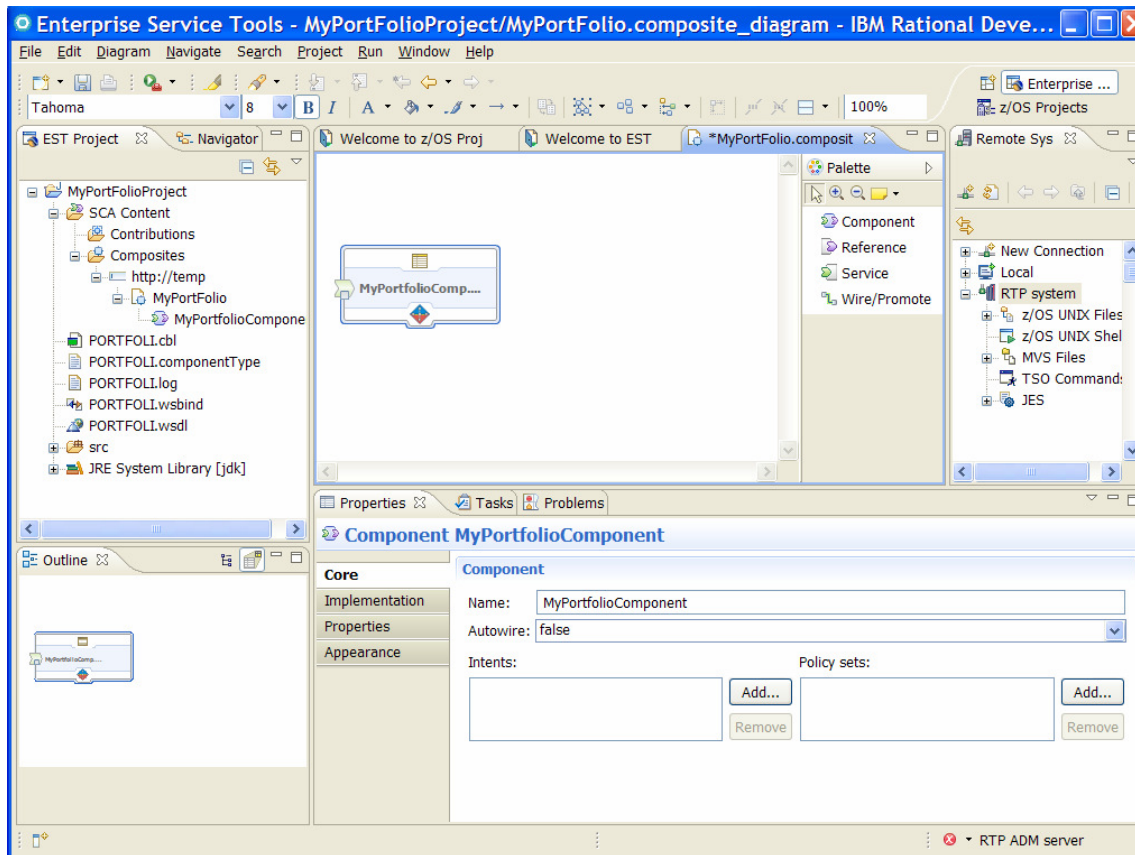


RDz SCA Tooling Support



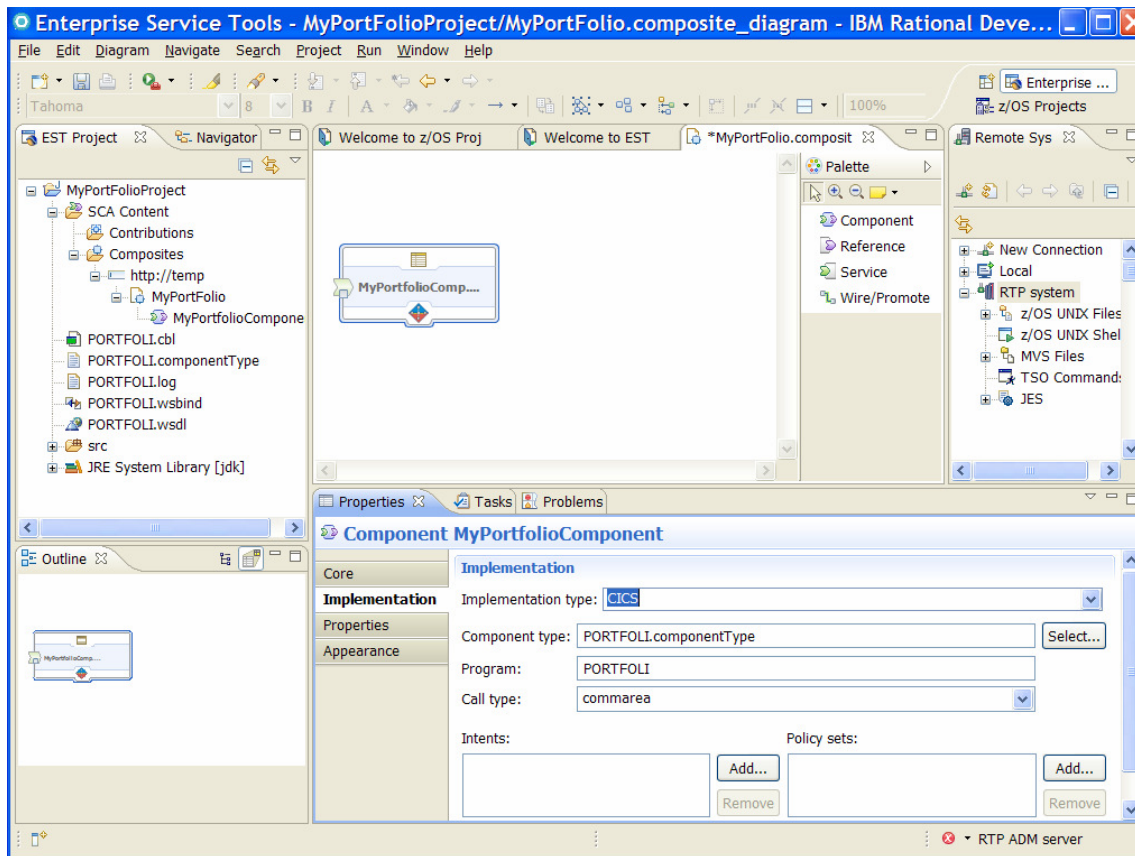


RDz SCA Tooling Support



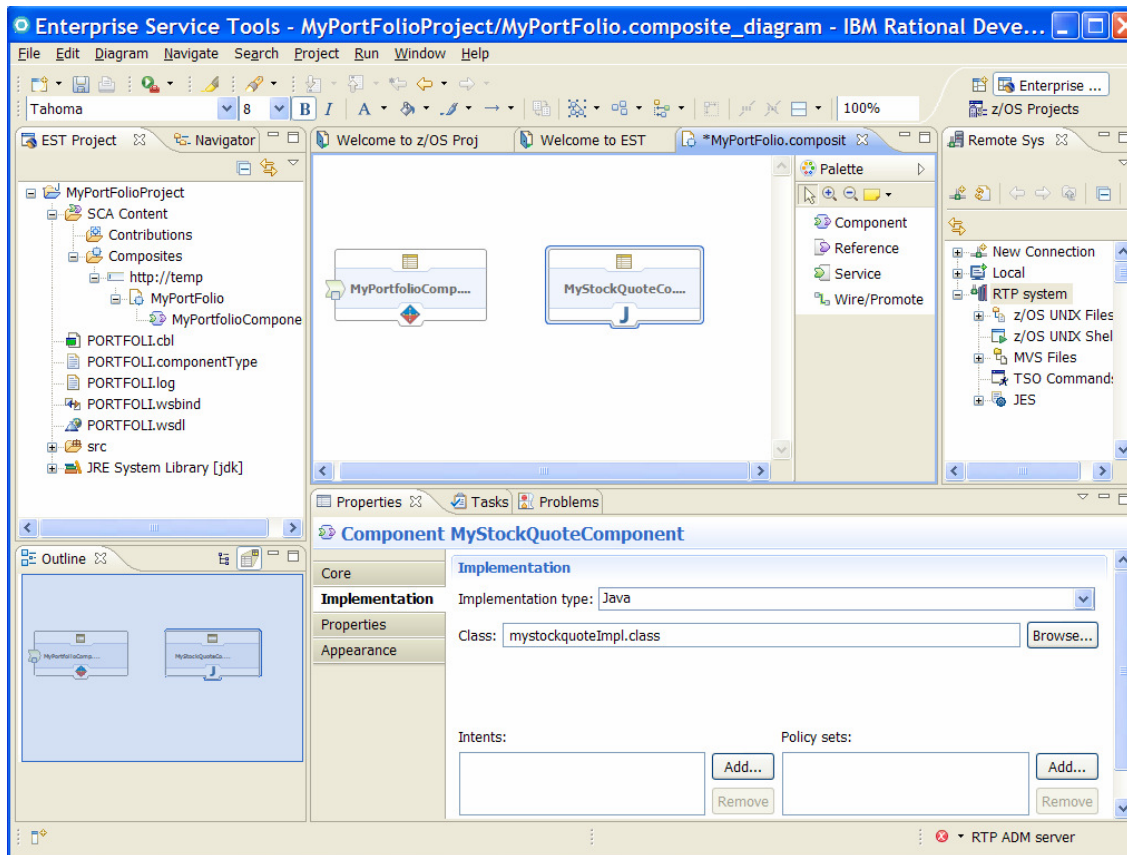


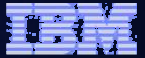
RDz SCA Tooling Support



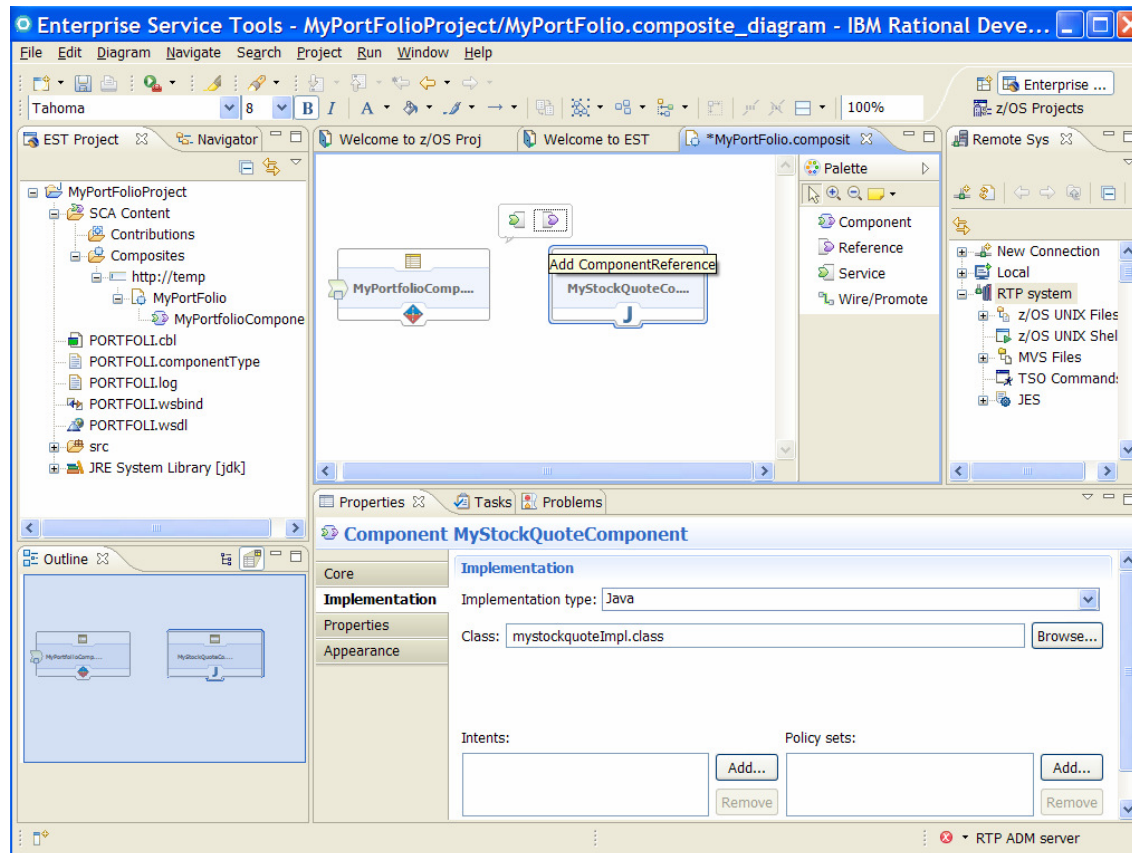


RDz SCA Tooling Support



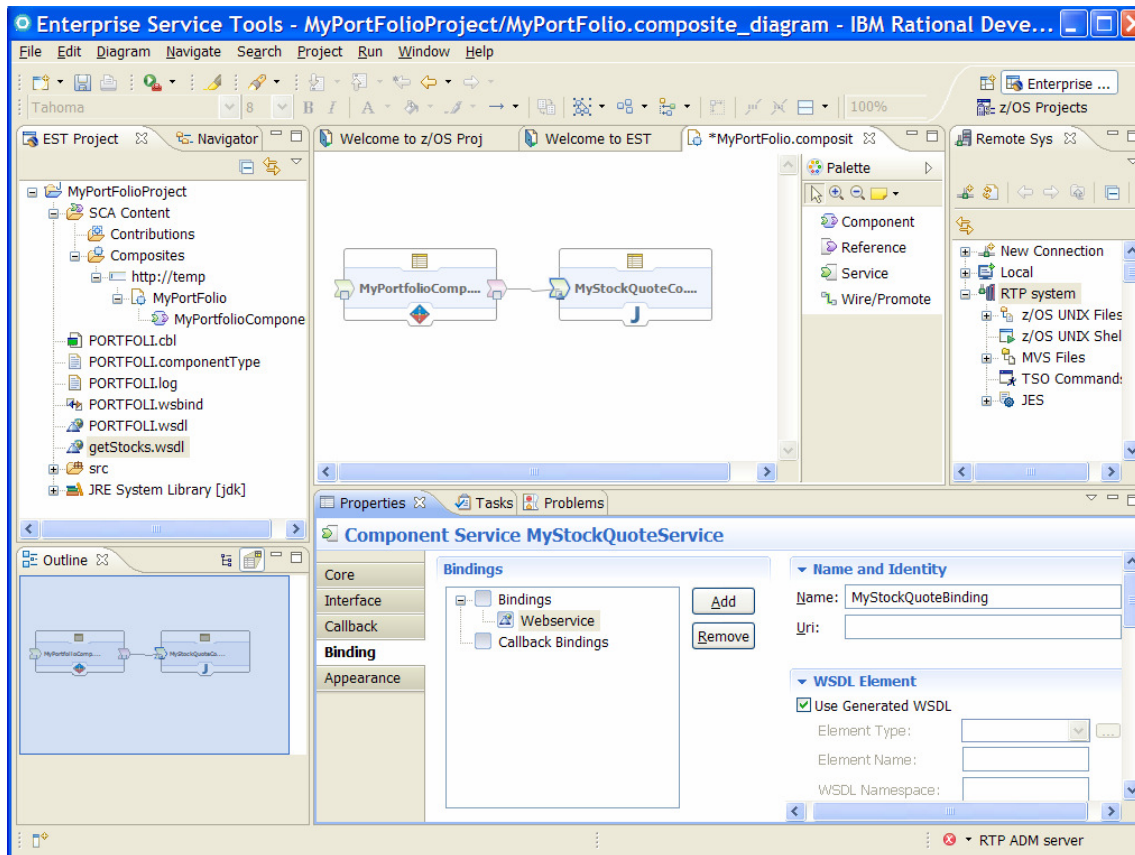


RDz SCA Tooling Support



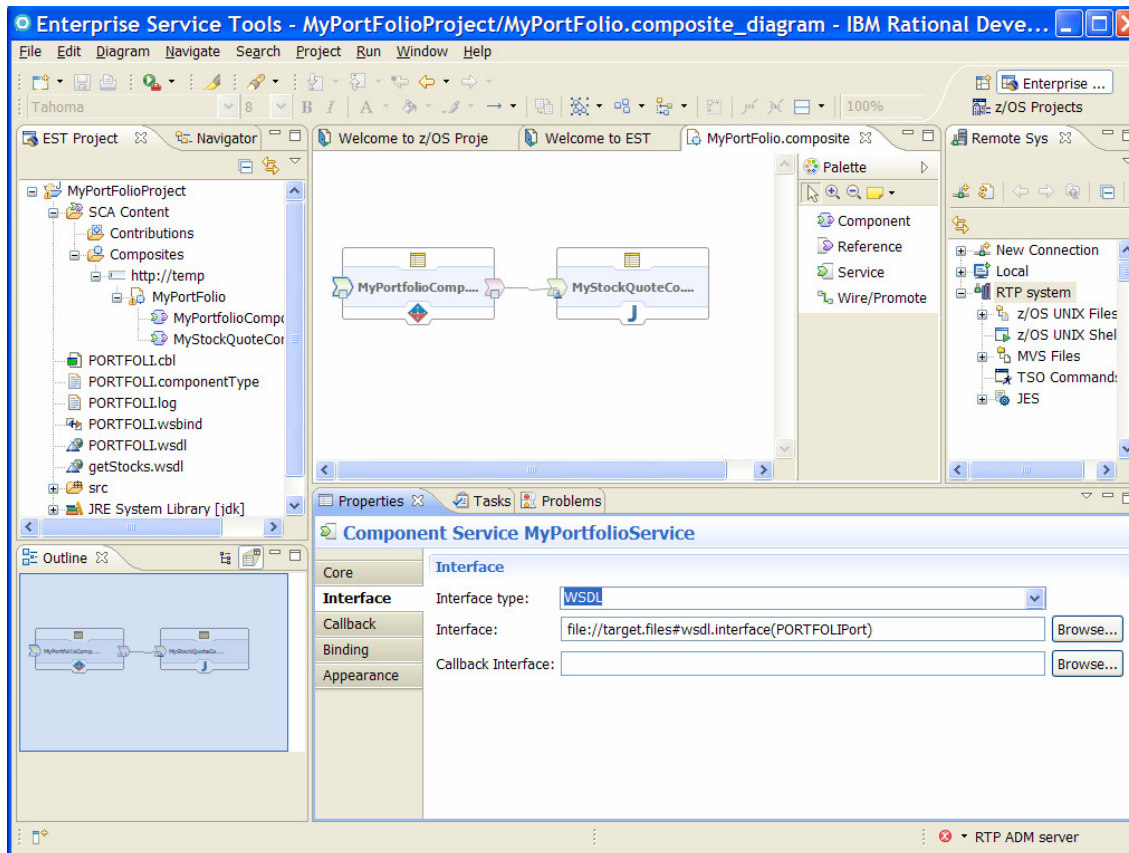


RDz SCA Tooling Support



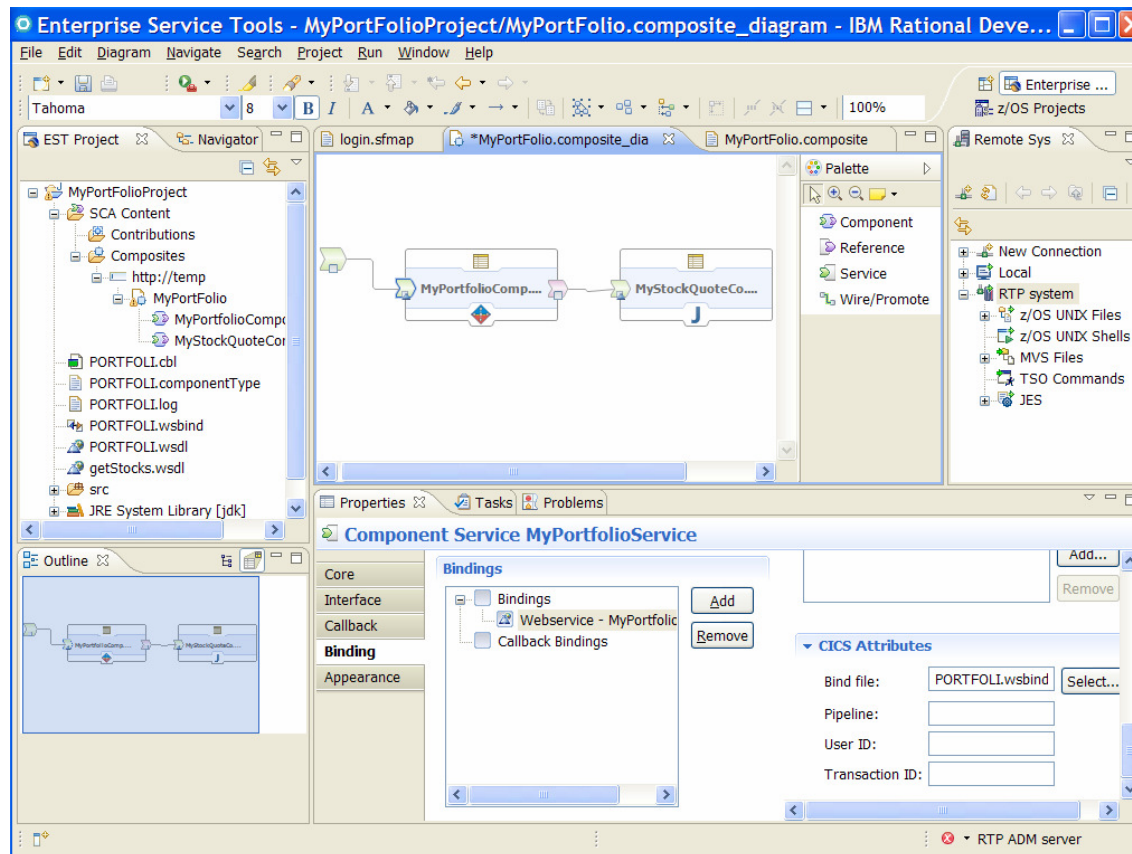


RDz SCA Tooling Support





RDz SCA Tooling Support





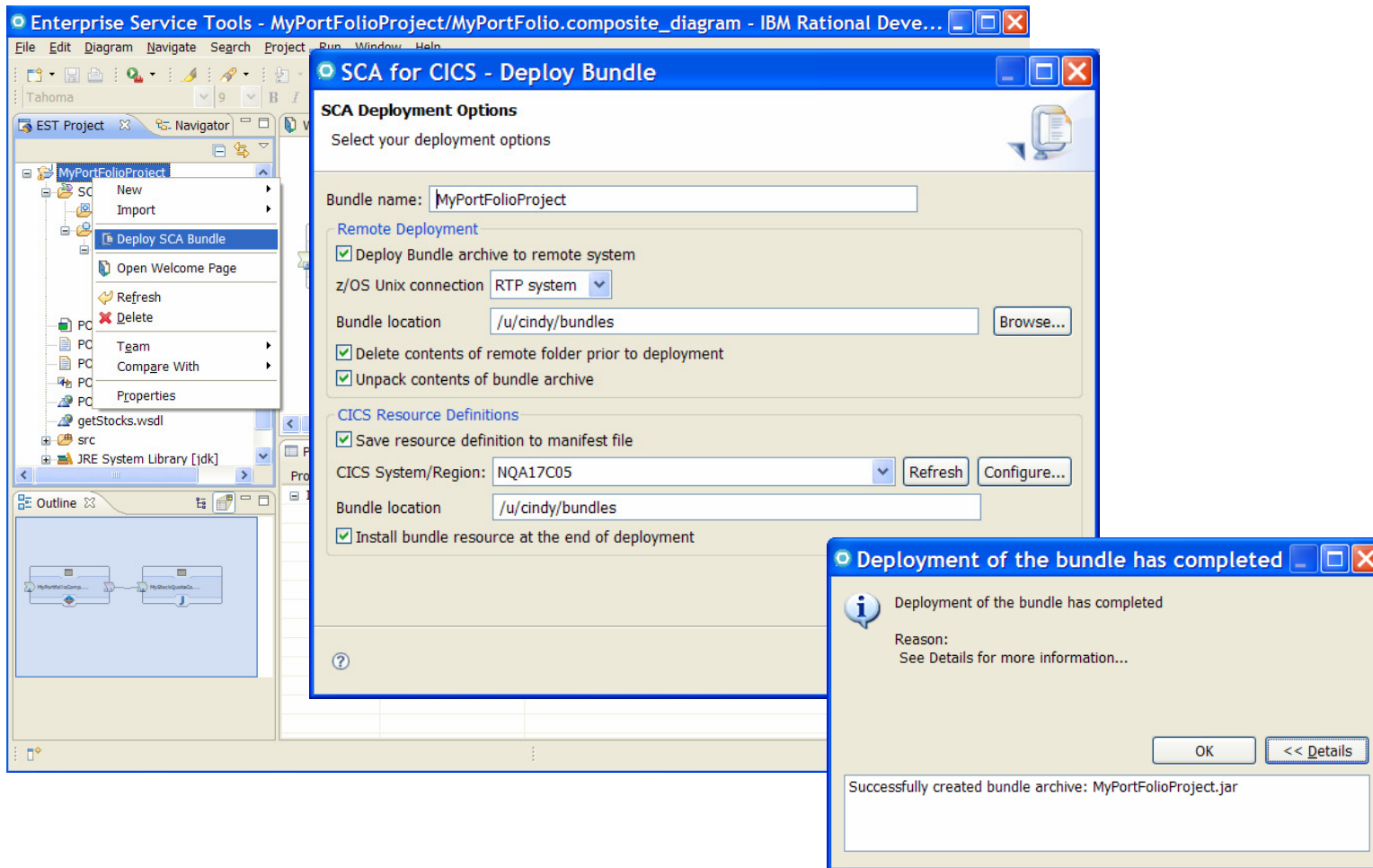
RDz SCA Tooling Support

```

<?xml version="1.0" encoding="UTF-8"?>
<composite xmlns="http://www.osoa.org/xmlns/sca/1.0" xmlns:cics="http://www.ibm.com/xmlns/prod/cics/sca/1.0/2007">
  <component name="MyPortfolioComponent">
    <cics:implementation.cics callType="commarea" componentTypePath="PORTFOLI.componentType" program="PORTFOLI"/>
    <service name="MyPortfolioService">
      <interface.wsdl interface="file://target.files#wsdl.interface(PORTFOLIPort)"/>
      <binding.ws name="MyPortfolioBinding" cics:bindfile="PORTFOLI.wsbind"/>
    </service>
    <reference name="MyStockQuoteService" target="MyStockQuoteComponent/MyStockQuoteService"/>
  </component>
  <component name="MyStockQuoteComponent">
    <implementation.java class="mystockquoteImpl.class"/>
    <service name="MyStockQuoteService">
      <interface.wsdl interface="http://tempuri.org/GetStockQuote/#wsdl.interface(NewWSDLFile)"/>
      <binding.ws name="MyStockQuoteBinding"/>
    </service>
  </component>
  <service name="MyPortfolioService" promote="MyPortfolioComponent/MyPortfolioService"/>
</composite>
    
```



RDz SCA Tooling Support





RDz SCA Tooling Support

The screenshot displays the IBM Rational Developer for System z interface for CICS SM tooling. The main window shows a composite diagram for 'MyPortFolio.composit'. The interface is divided into several panes:

- CICS Explorer:** Shows the server 'NQA17C05' and a bundle 'NQA17C05 (NQA17C05)'.
- Regions:** A table listing regions and their bundle directories.
- Properties:** A table of system properties for the selected bundle.
- Remote Systems:** A tree view showing the file structure of the remote system, including 'z/OS UNIX Files', 'My Home', 'bundles', and various files like 'cics.xml', 'getStocks.wsdl', and 'PORTFOLI.wsdl'.
- Diagram Editor:** Shows a composite diagram with a component 'MyPortfolioComp...' and a palette with options like 'Component', 'Reference', 'Service', and 'Wire/Promote'.

Name	BundleDir
MYPORTFO	/u/cindy/bundles/

Property	Value
Basdefinever	N/A
Basescope	N/A
BundleDir	/u/cindy/bundles/
Changeagent	N/A
Changeagrel	N/A
Changetime	N/A
Changeusrid	N/A
Definesource	N/A
Definetime	N/A
Installagent	N/A
Installtime	N/A
Installusrid	N/A
Name	MYPORTFO
Region	NQA17C05
Status	DISABLED

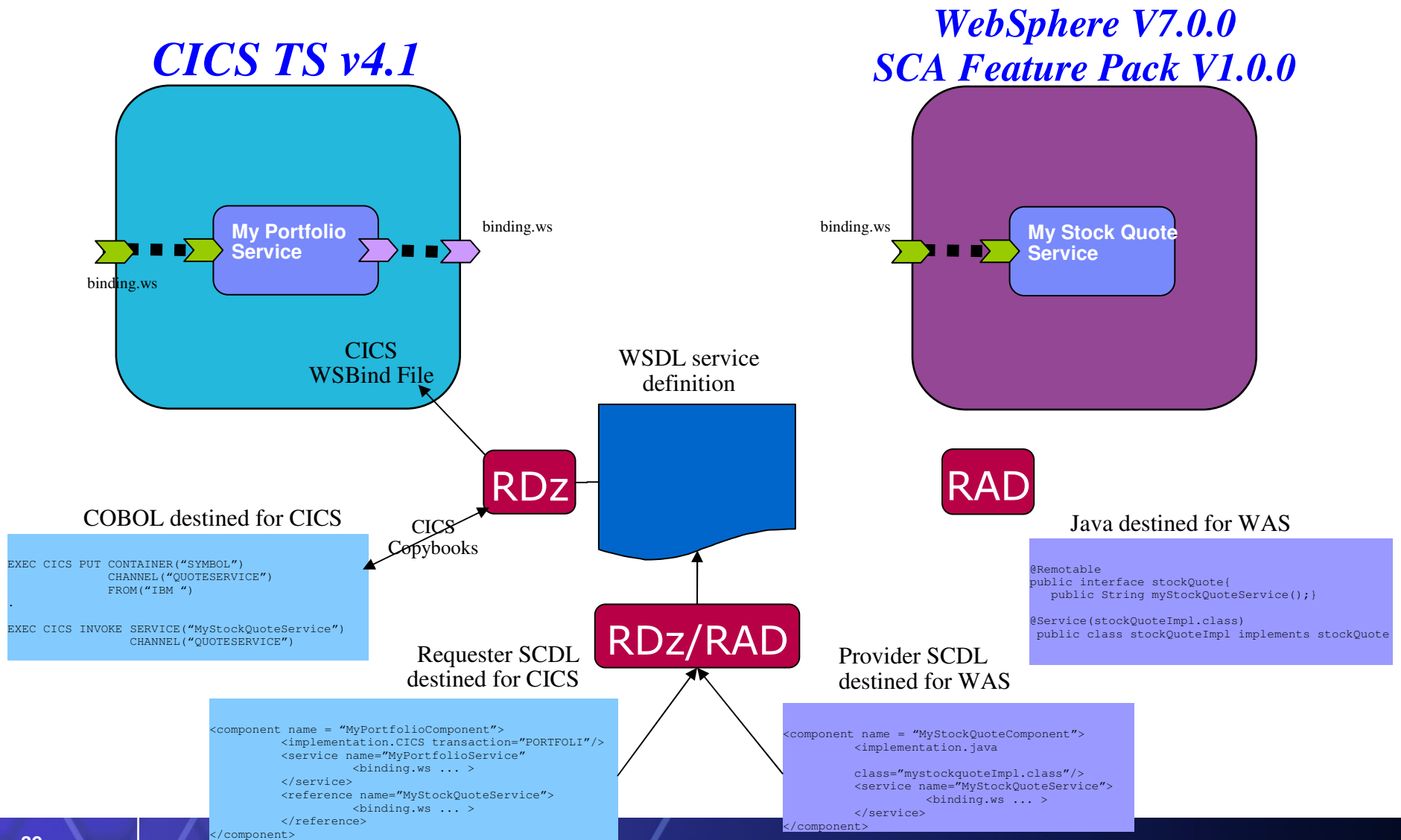


Agenda

- Quick SCA in WAS recap
- Introduction to SCA support in CICS TS v4.1
- The Portfolio Scenario
- Defining and implementing SCA components using Rational Developer for System z
- **Deploying and running SCA components in CICS TS**
- Summary and Questions



Service Component Architecture – Development Artifacts





Agenda

- Quick SCA in WAS recap
- Introduction to SCA support in CICS TS v4.1
- The Portfolio Scenario
- Defining and implementing SCA components using Rational Developer for System z
- Deploying and running SCA components in CICS TS
- **Summary and Questions**



References

- CICS Transaction Server v4.1
 - Open Beta: <http://www.ibm.com/software/hcp/cics/tserver/v41/openbeta/>
- Rational Developer for System z
 - <http://www.ibm.com/software/awdtools/rdz/>
- Open Service Oriented Architecture Web site for SCA v1.0 Specifications
 - <http://www.osoa.org/>
- OASIS Open CSA Web site for SCA v1.x
 - <http://www.oasis-opencsa.org/sca>
- Apache Tuscany Web site
 - <http://incubator.apache.org/tuscany/>
- SCA feature pack support website
 - <http://www.ibm.com/support/docview.wss?rs=180&context=SSEQTP&dc=DB600&uid=swg21329175>
- DeveloperWorks
 - <http://www.ibm.com/developerworks/websphere>



WebSphere software