IBM System z Technology Summit

Revitalizing applications Part I – CICS application development

These materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. 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. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. 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 the applicable license agreement governing the use of IBM software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including 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 results similar to those stated here.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries:
ibm.com/legal/copytrade.shtmlAIX, CICS, CICSPlex, DataPower, DB2, DB2 Universal Database, i5/OS, IBM, the IBM logo, IMS/ESA, Power Systems, Lotus, OMEGAMON, OS/390, Parallel Sysplex, pureXML, Rational, Redbooks, Sametime, SMART SOA, System z , Tivoli, WebSphere, and z/OS.

A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office
Intel and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

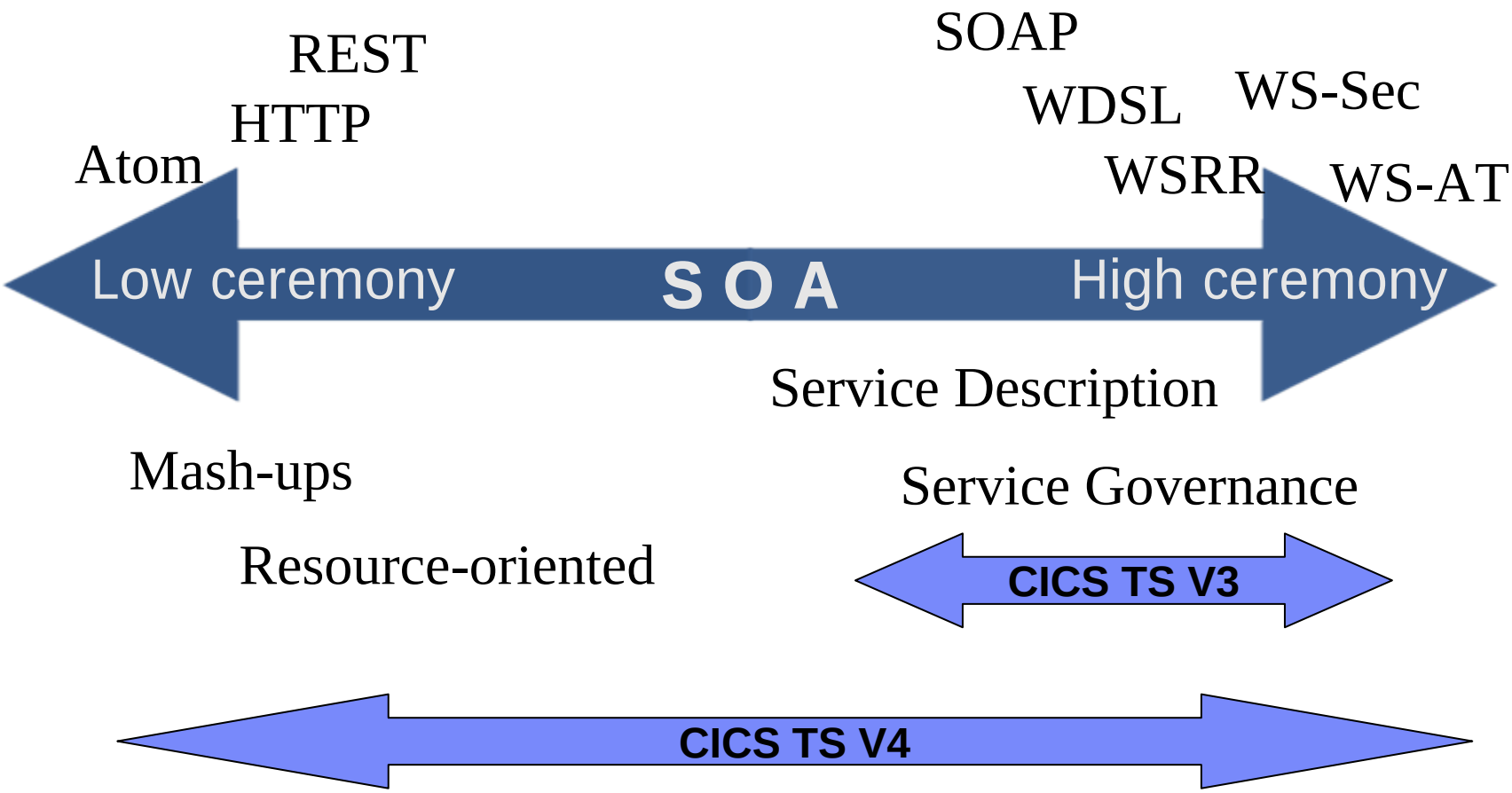
Session 2. Abstract

- **Breathe new life into existing applications by extending them into new business processes. Service Component Architecture (SCA) can enable you to identify and package applications (be they new or existing) in a more flexible way. Learn how component reuse in many composite applications simplifies the process of rapidly assembling and deploying new business applications. Mature SOA support in CICS includes data mapping and Web services addressing, providing faster and enhanced conversion between XML and language structures for all Web services, and APIs to use these services, independent of Web services.**

Agenda

- **Spectrum of Service enablement**
- **Service Component Architecture**
- **Bundles**
- **WS-Addressing**
- **XML Transforms and offload**

A Spectrum of Service Enablement



RESTful web services

- **REST**
  - REpresentational State Transfer - See http://en.wikipedia.org/wiki/Representational_State_Transfer
  - How the Web has always worked...
    - HTTP methods POST | GET | PUT | DELETE
      … all the verbs you'll ever need
    - Everything else is a *resource* – with a representation of its state

- **“Clean and meaningful URLs”**
  - For everything!
  - e.g. a file, a database, a TSqueue....
    - or a single record within each resource

Service Component Architecture

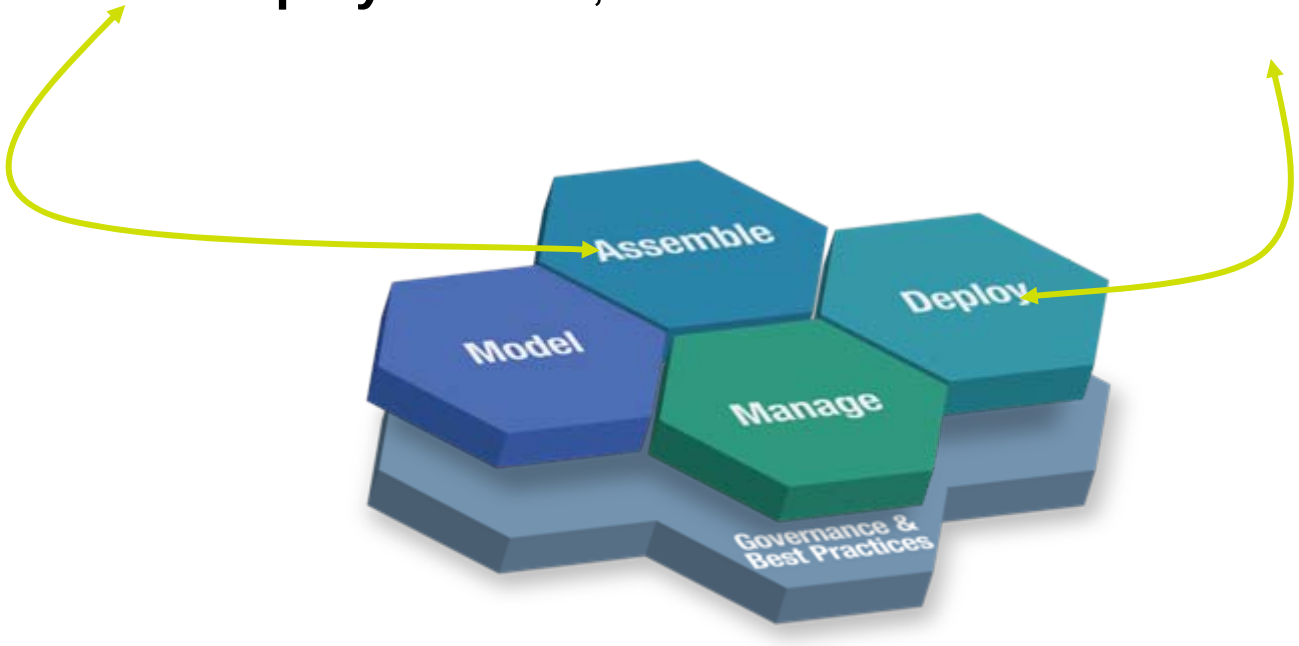
- **What is SCA**
- **Introduction to SCA support in CICS TS V4.1**
- **The Portfolio Scenario**
- **Defining and implementing SCA components using Rational Developer for System z**
- **Deploying and running SCA components in CICS TS**

SCA Terms and Relations to SOA Foundation

SCA is the development, deployment model of the **SOA** Foundation.

SCA is the open standard model for service assembly.

Assemble = develop interfaces, implementations, composites.
Deploy = define, install and run contributions



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).
- **"Open SCA" refers to Service Component Architecture as defined by the industry at the OSOA collaboration and OASIS**
  - "Classic SCA" refers to Service Component Architecture as defined and built by IBM and supported in a variety of WebSphere Family products starting with V6.

SCA: What it is NOT

- **Does not model individual** *workflows*
  - use BPEL or other workflow languages
- **Is not solely** *Web services*
  - SCA can use / may use Web services, but can also build solutions with no Web services content
- **Is not tied to a specific runtime environment**
  - distributed, heterogeneous, large, small
- **Does not force use of specific programming languages and technologies**
  - aims to encompass many languages, frameworks, technologies
- **embrace not replace**
  - adaptable to new technology

Key benefits of SCA

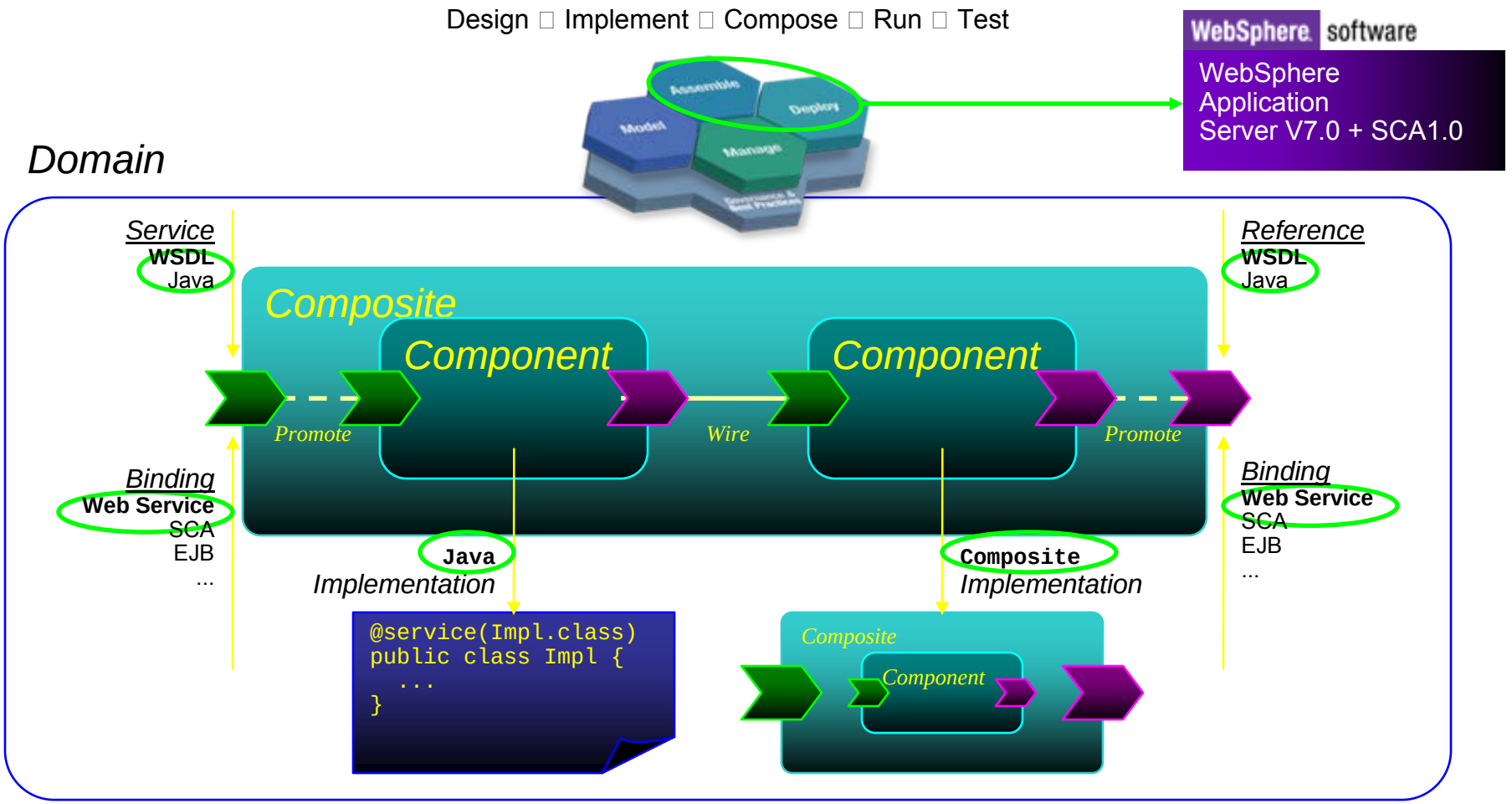
- *Separation of Concerns* **Developers in an SOA need only be concerned with what they need to be**
- *Loose Coupling* **Components integrate without need to know how others are implemented**
- *Flexibility* **Components can easily be replaced by other components**
- *Services* **can be easily invoked either synchronously or asynchronously**
- *Composition* **of solutions: clearly described**
- *Productivity* **Easier to integrate components to form composite application**
- *Heterogeneity* **Multiple implementation languages, communication mechanisms**
- *Declarative* **application of infrastructure services**
- *Simplification* **for all developers, integrators and application deployers**

SCA v1.0

- **OSOA – Consortium of industry vendors**
  - http://www.osoa.org
- **The OASIS Open Composite Services Architecture (CSA) Member Section advances open standards that simplify SOA application development.**
  - http://www.oasis-opencsa.org/
  - Open CSA brings together vendors and users from around the world to collaborate on the further development and adoption of the Service Component Architecture (SCA) and Service Data Objects (SDO) families of specifications
- **Apache Tuscany simplifies the task of developing SOA solutions by providing a comprehensive infrastructure for SOA development and management that is based on Service Component Architecture (SCA) standard.**
  - http://tuscany.apache.org/

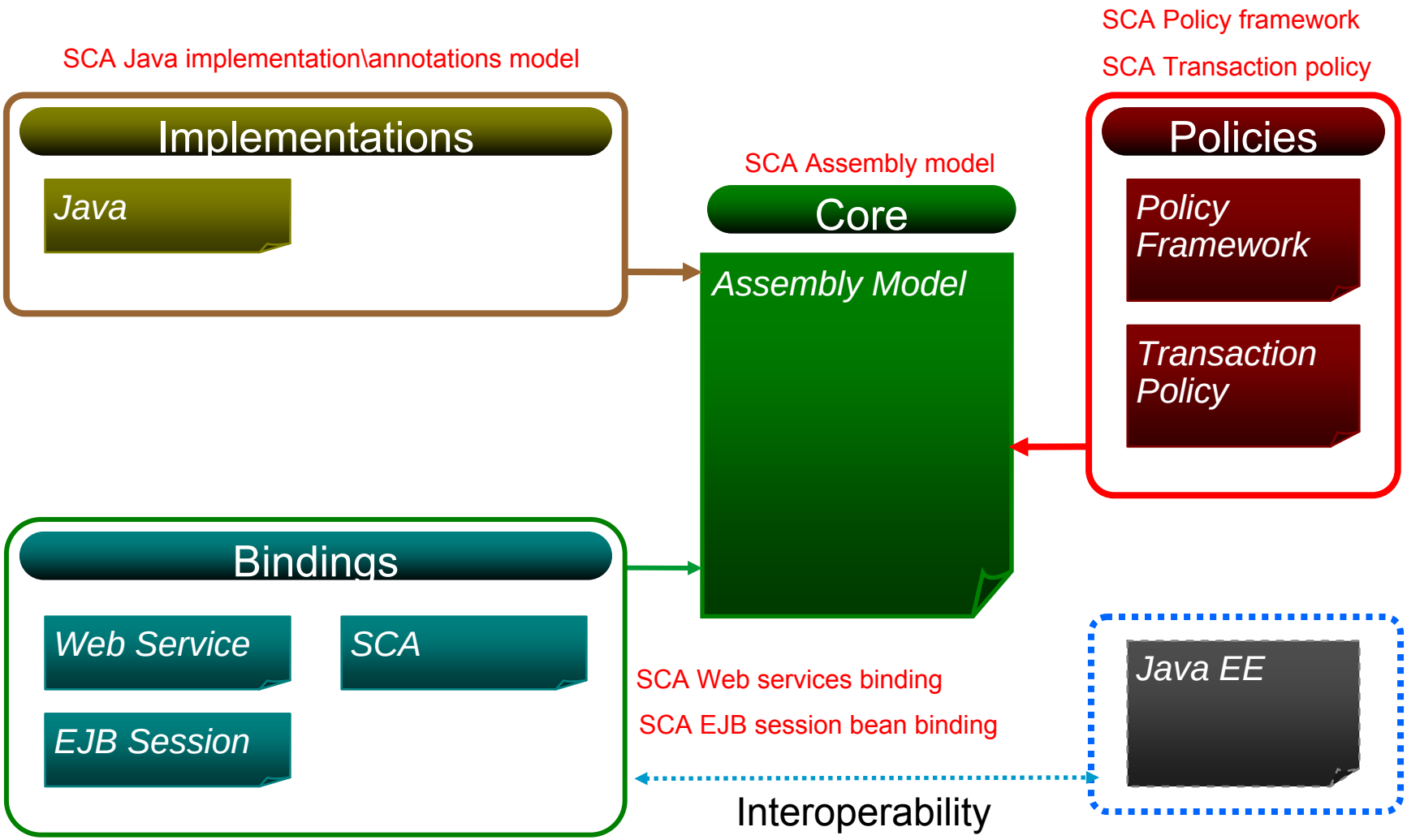
OASIS Open CSA

SCA Key Concepts



Reusability, Connectivity, Flexibility, Extensibility

SCA v1.0 Specifications – Flexible & Extensible



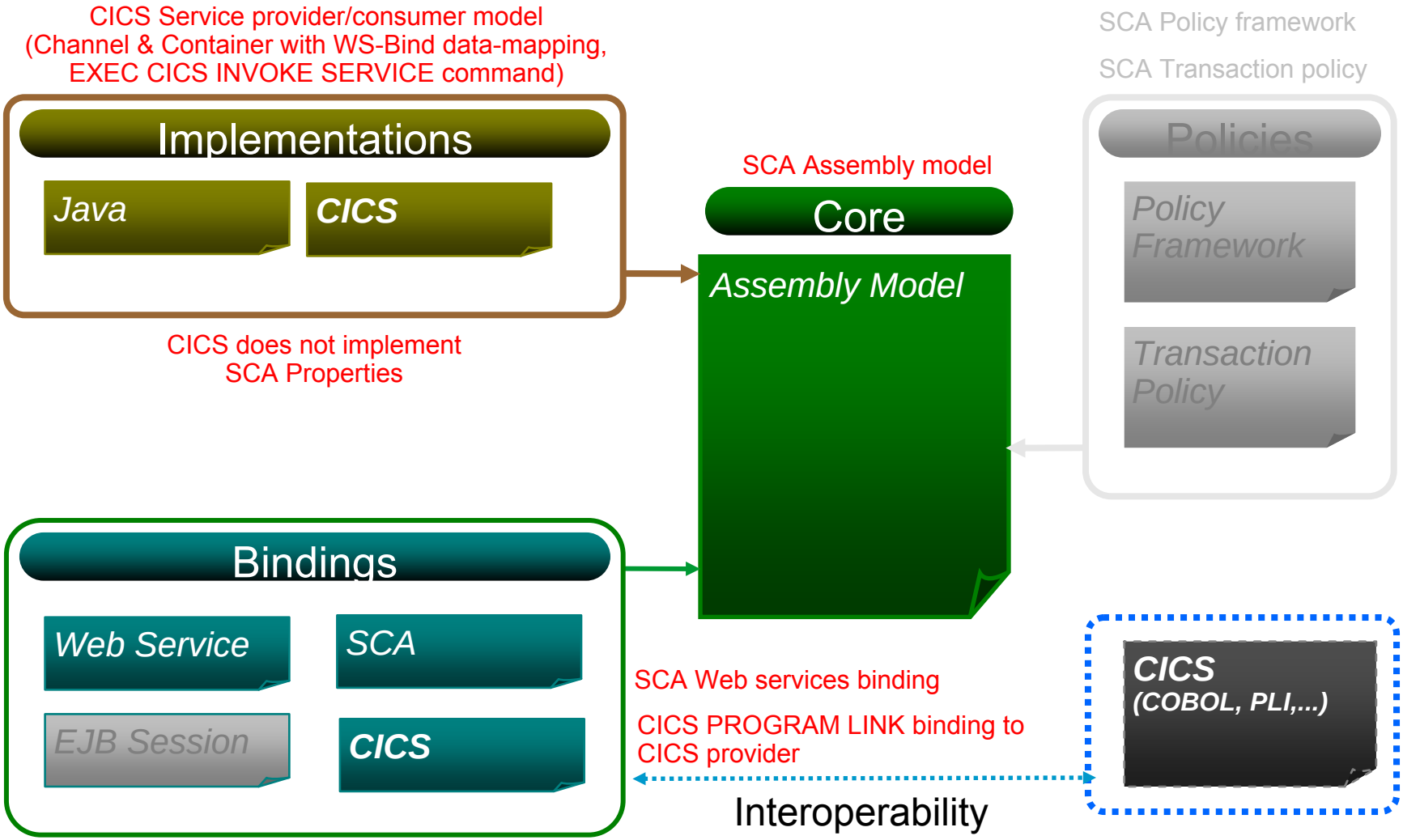
CICS TS V4.1 Component Architecture

- **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**

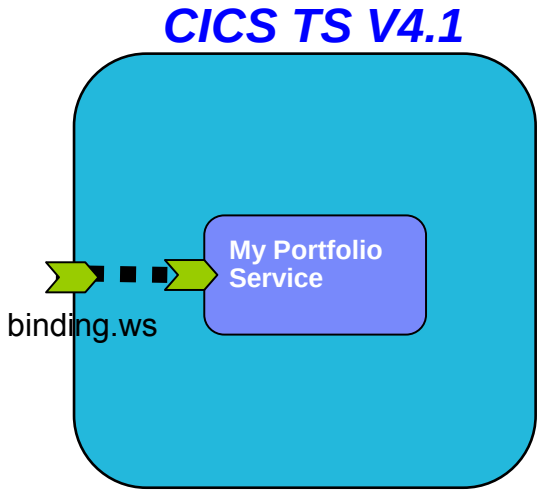
Component Architecture in CICS TS V4.1

- **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 provides CICS component tooling to enable component definition, assembly and deployment**

SCA v1.0 Specifications – CICS TS V4.1 capabilities

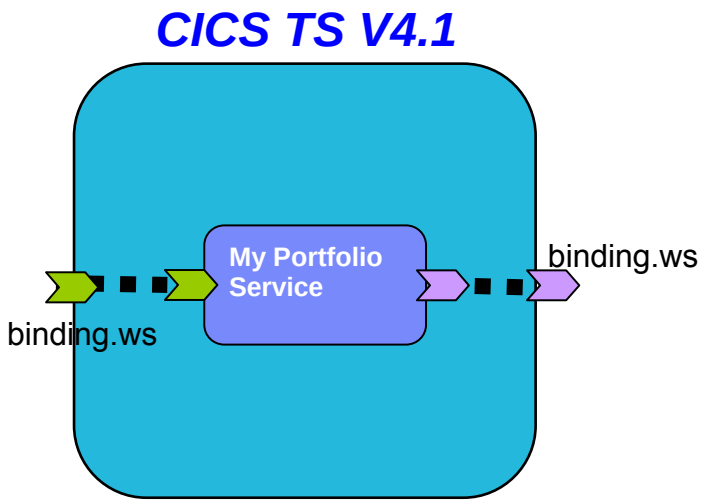


Service Component Architecture - Scenario



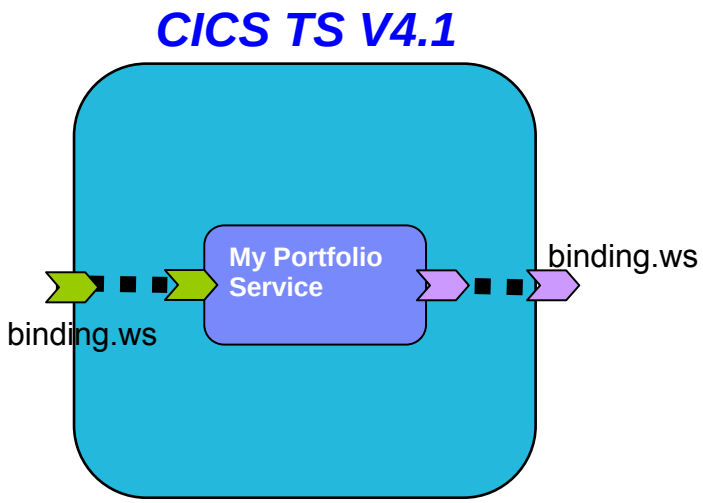

```
<component name = “MyPortfolioComponent”>
    <implementation.CICS program=”PORTFOLI”/>
    <service name=”MyPortfolioService”
        <binding.ws ... >
    </service>



</component>
```

Service Component Architecture – Scenario




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

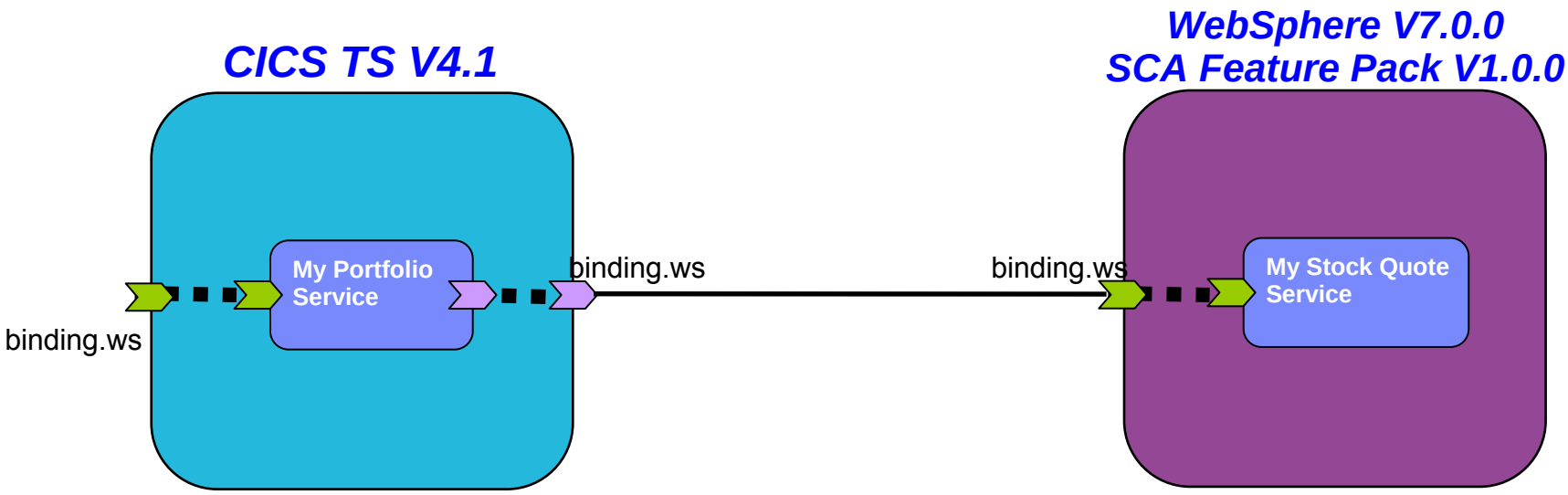
Service Component Architecture – Programming Model



CICS COBOL Program

```
EXEC CICS PUT CONTAINER(“SYMBOL”)
              CHANNEL(“QUOTESERVICE”)
              FROM(“IBM “)
.

EXEC CICS INVOKE SERVICE(“MyStockQuoteService”)
                 CHANNEL(“QUOTESERVICE”)
```

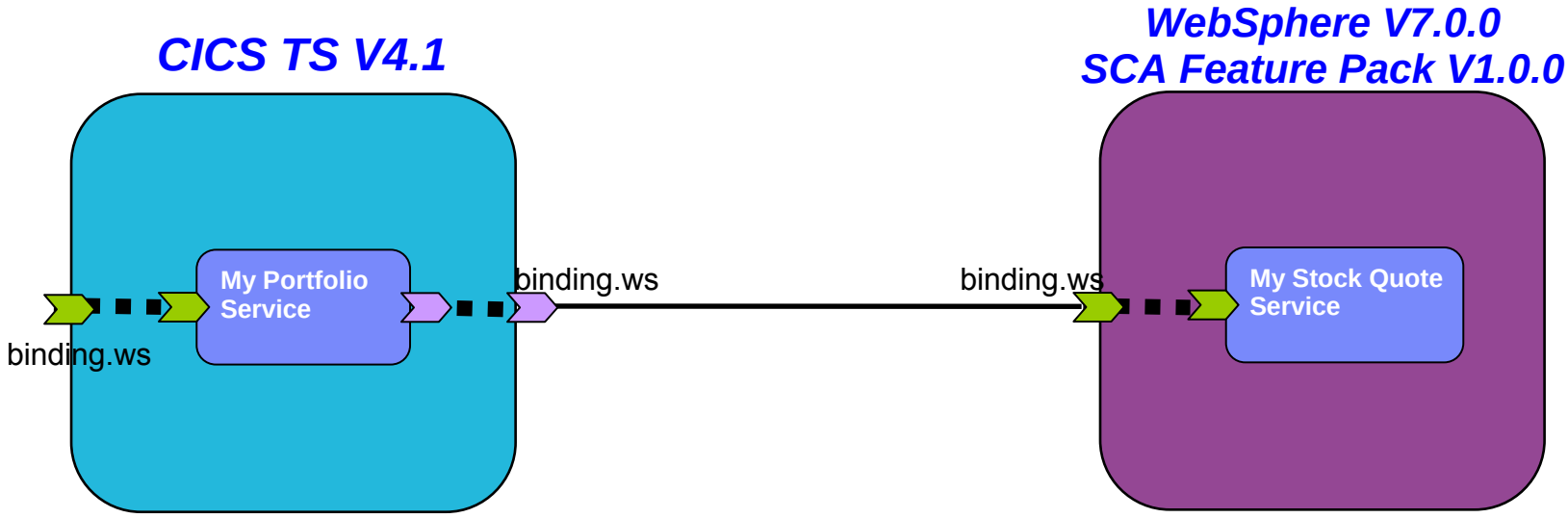
Service Component Architecture – Scenario




```
<component name = "MyPortfolioComponent">
    <implementation.CICS program="P
    <service name="MyPortfolioServi
        <binding.ws ... >
    </service>
    <reference name="MyStockQuoteSe
        <binding.ws ... >
    </reference>
</component>
```

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

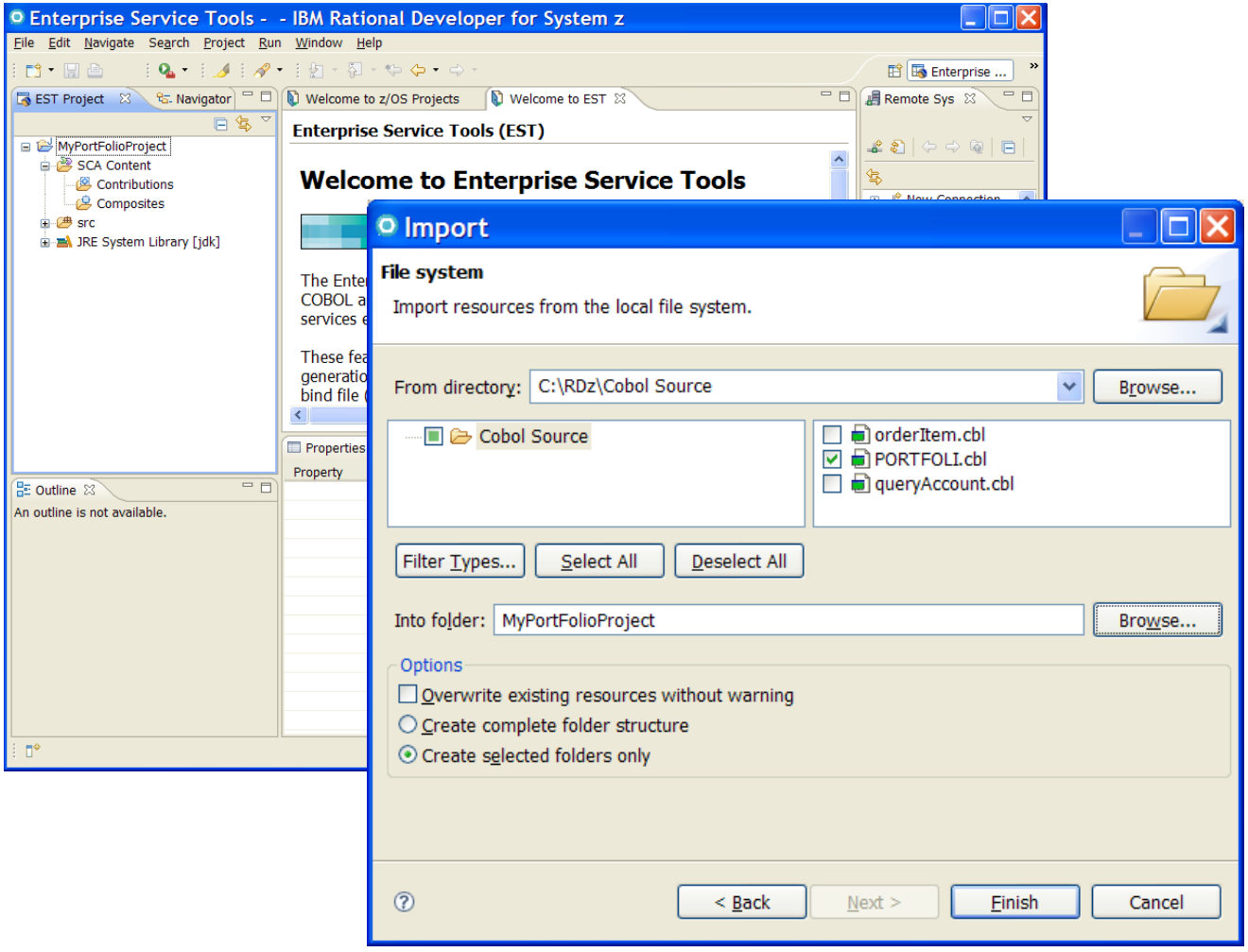
Service Component Architecture – Programming Model



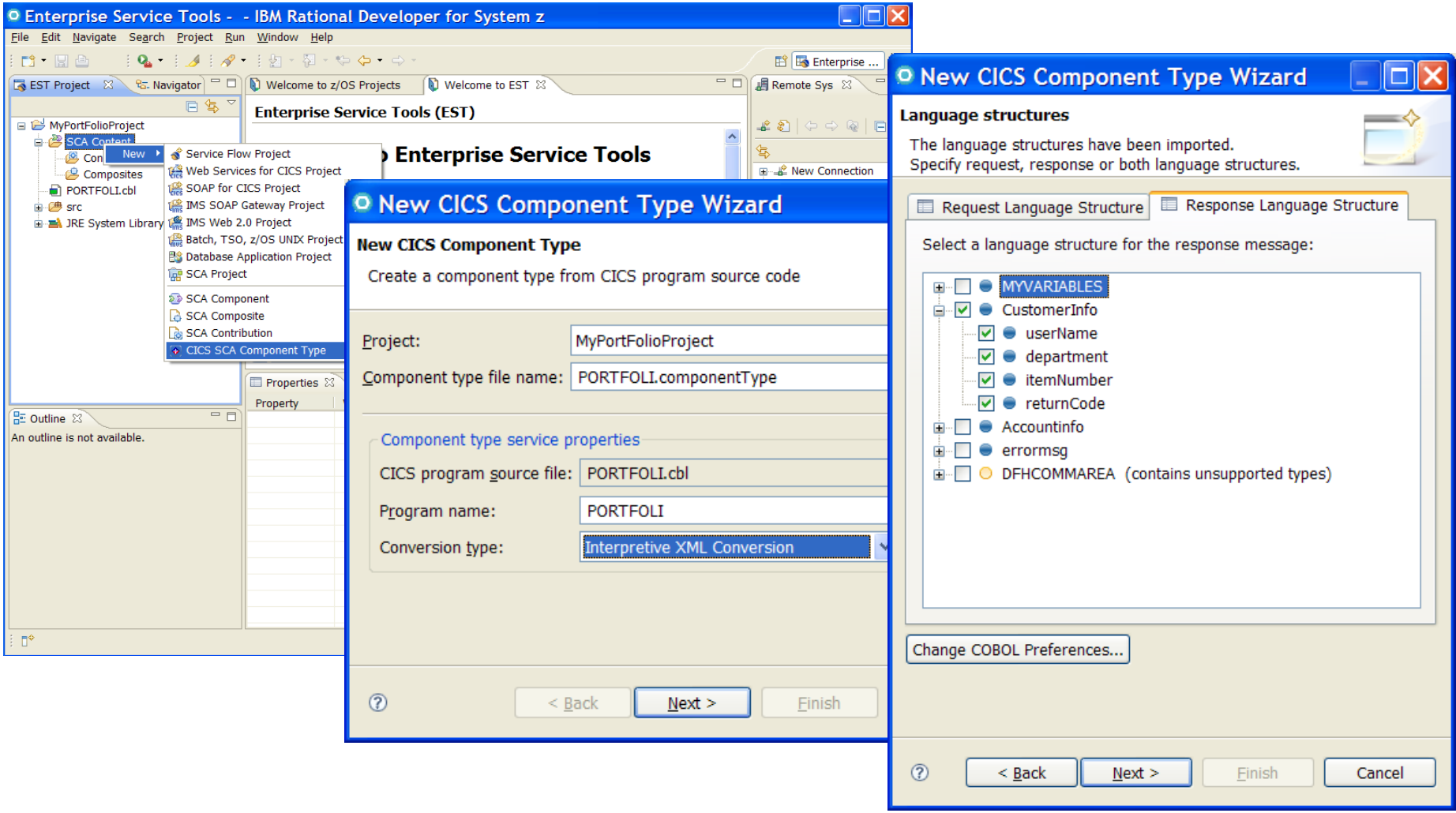

```
CICS COBOL Program

EXEC CICS PUT CONTAINER("SYMBO
              CHANNEL("QUOTESE
              FROM("IBM ")
.

EXEC CICS INVOKE SERVICE("MySt
                 CHANNEL("QUOT
```

```
WebSphere POJO

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

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

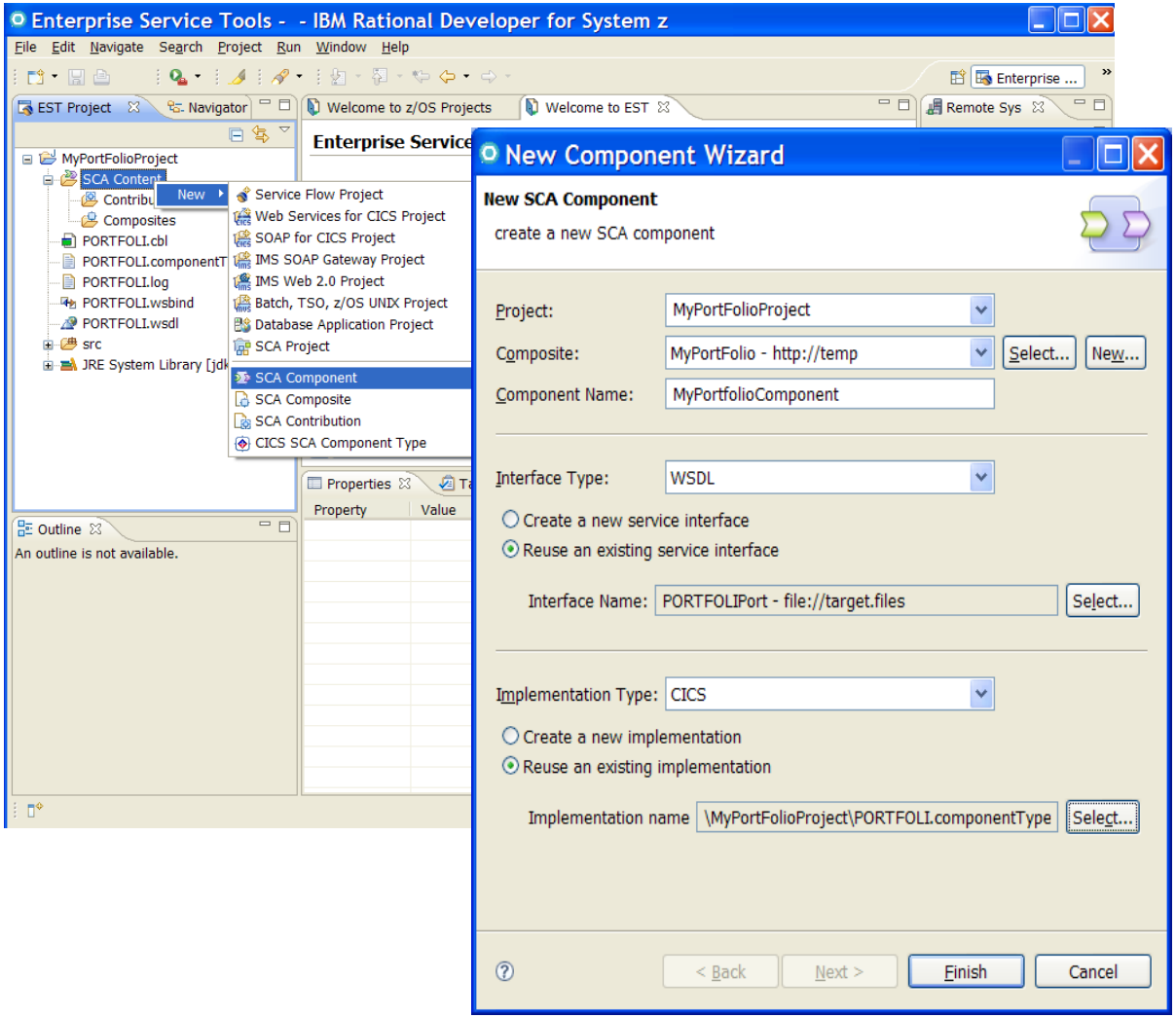
RDz SCA Tooling Support



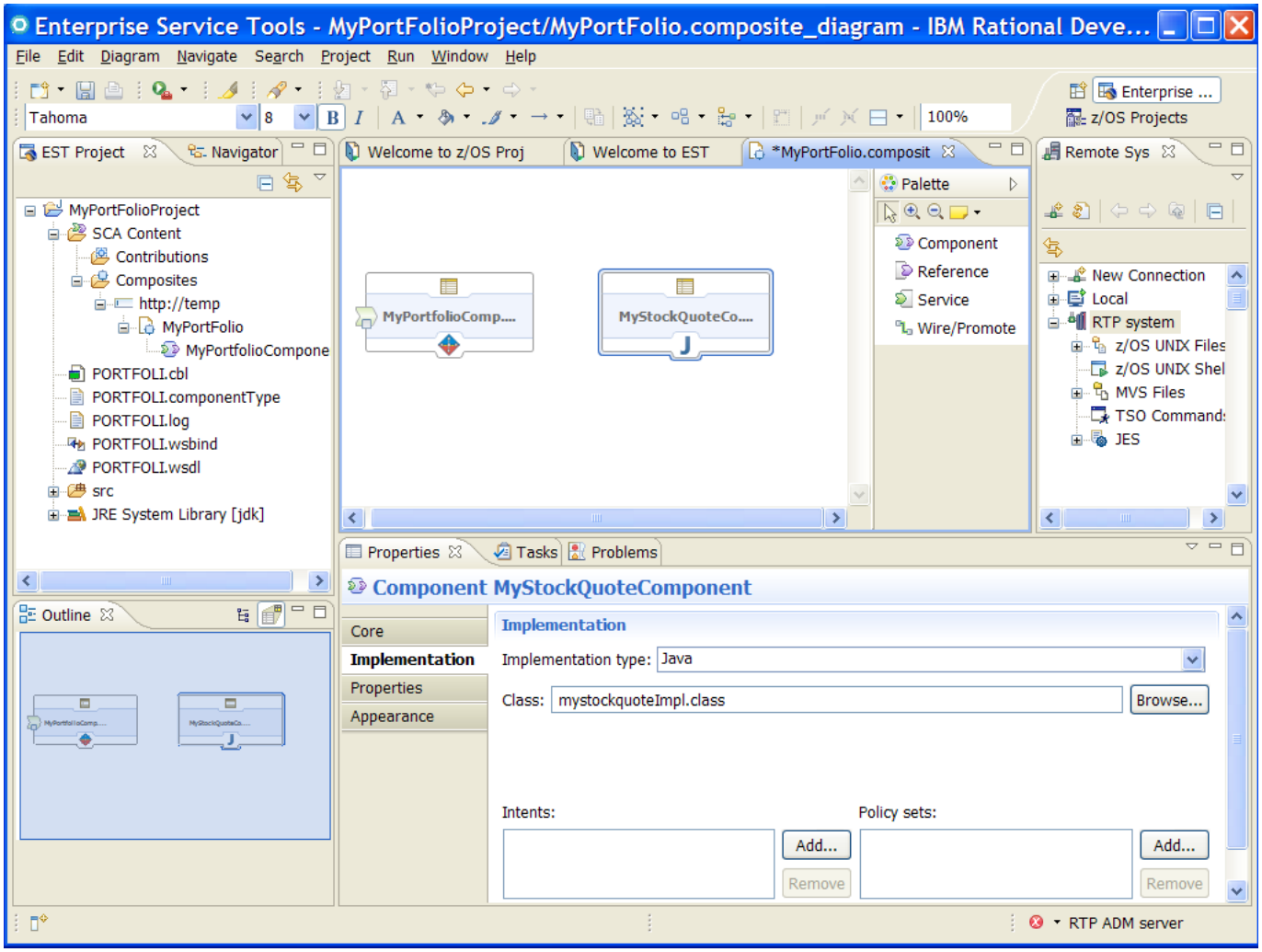
RDz SCA Tooling Support



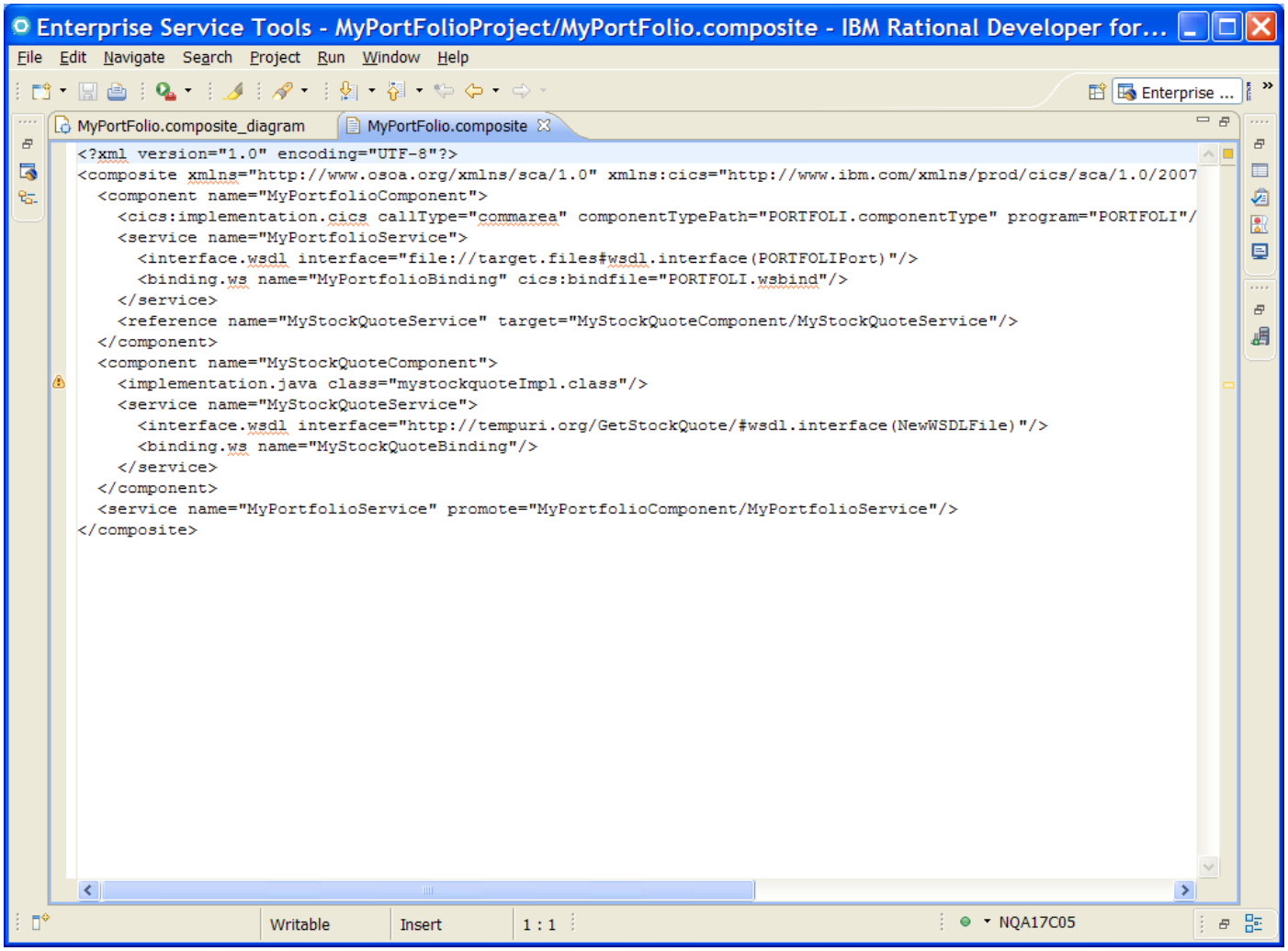
RDz SCA Tooling Support



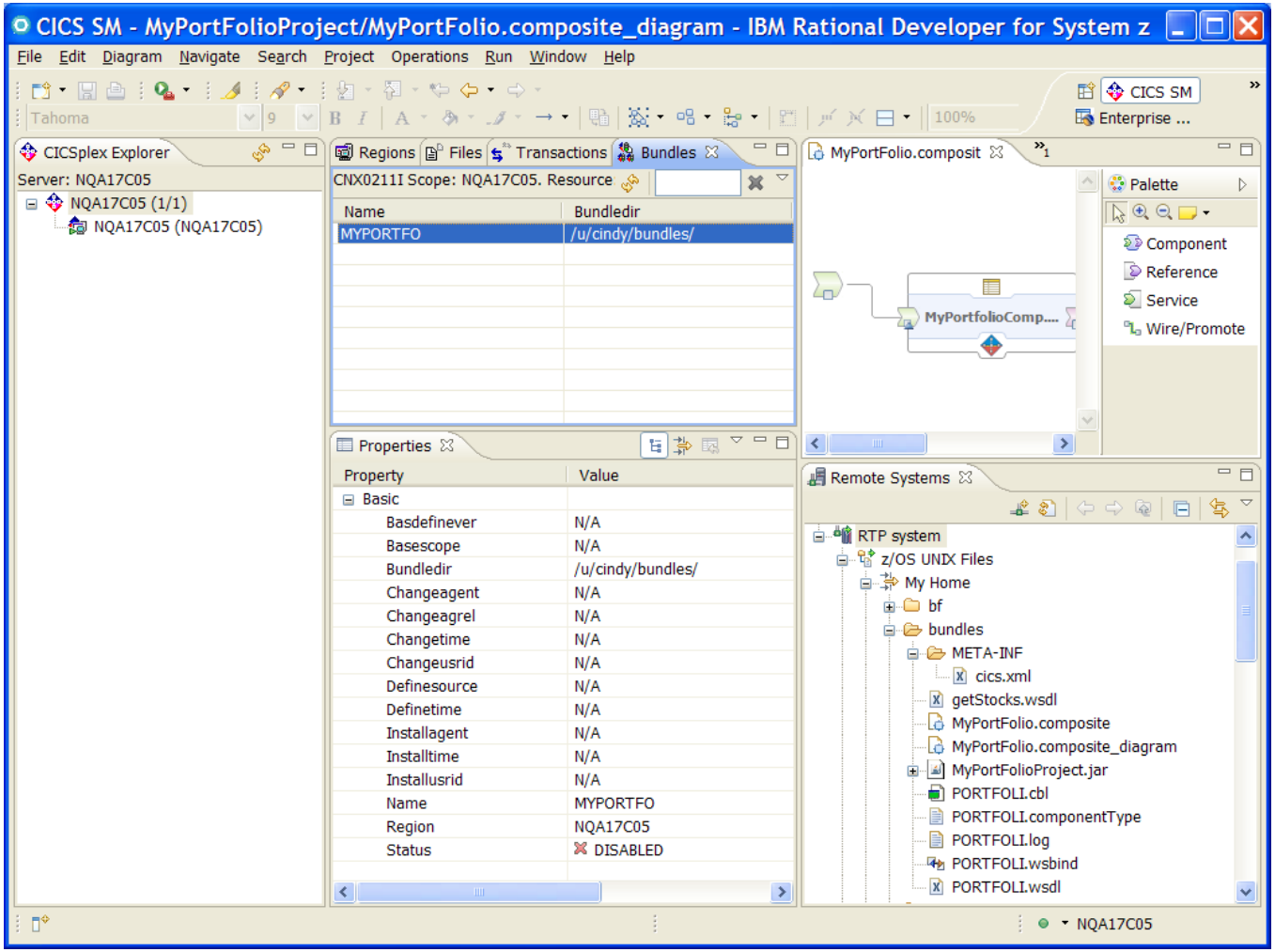
RDz SCA Tooling Support



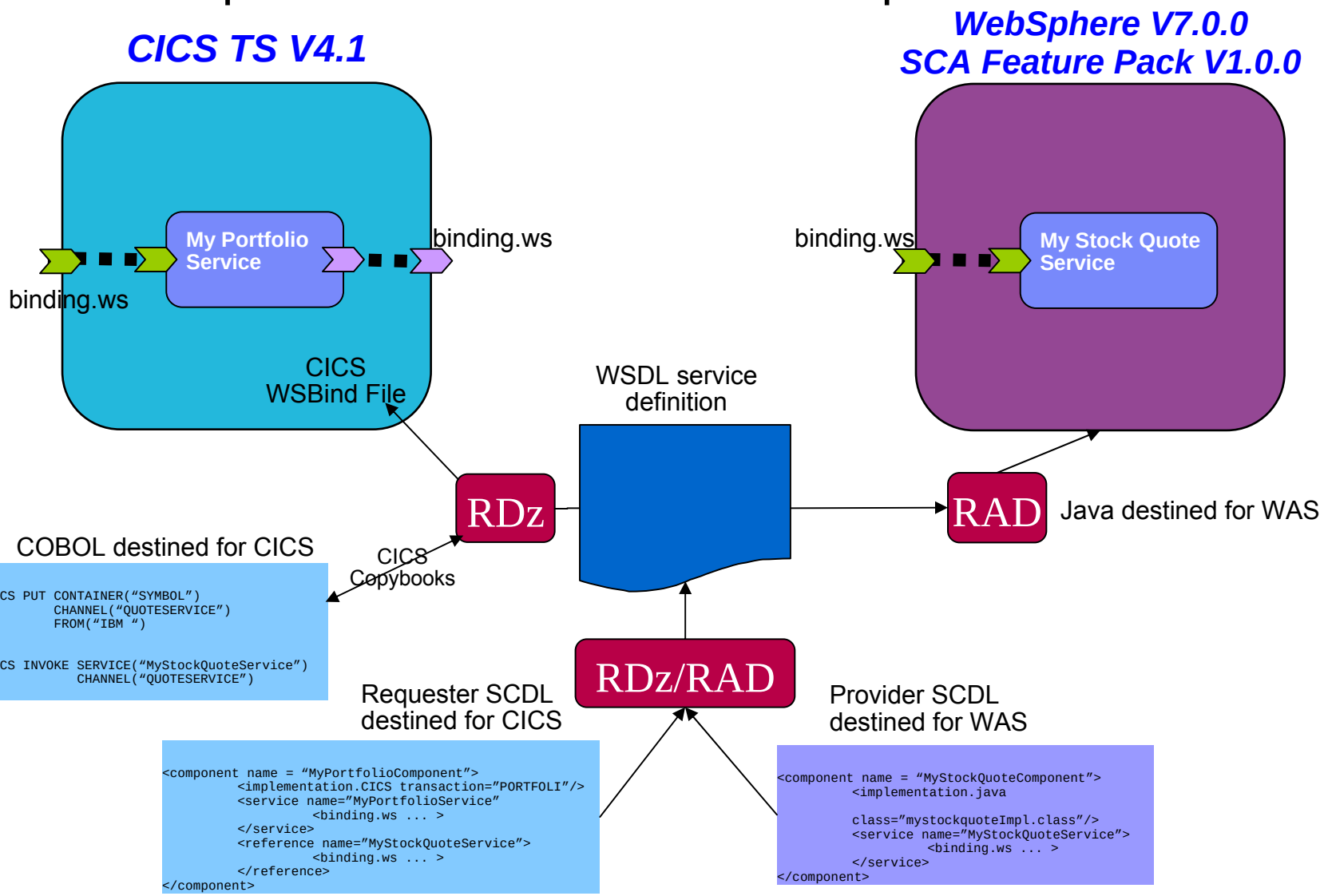
RDz SCA Tooling Support



RDz SCA Tooling Support



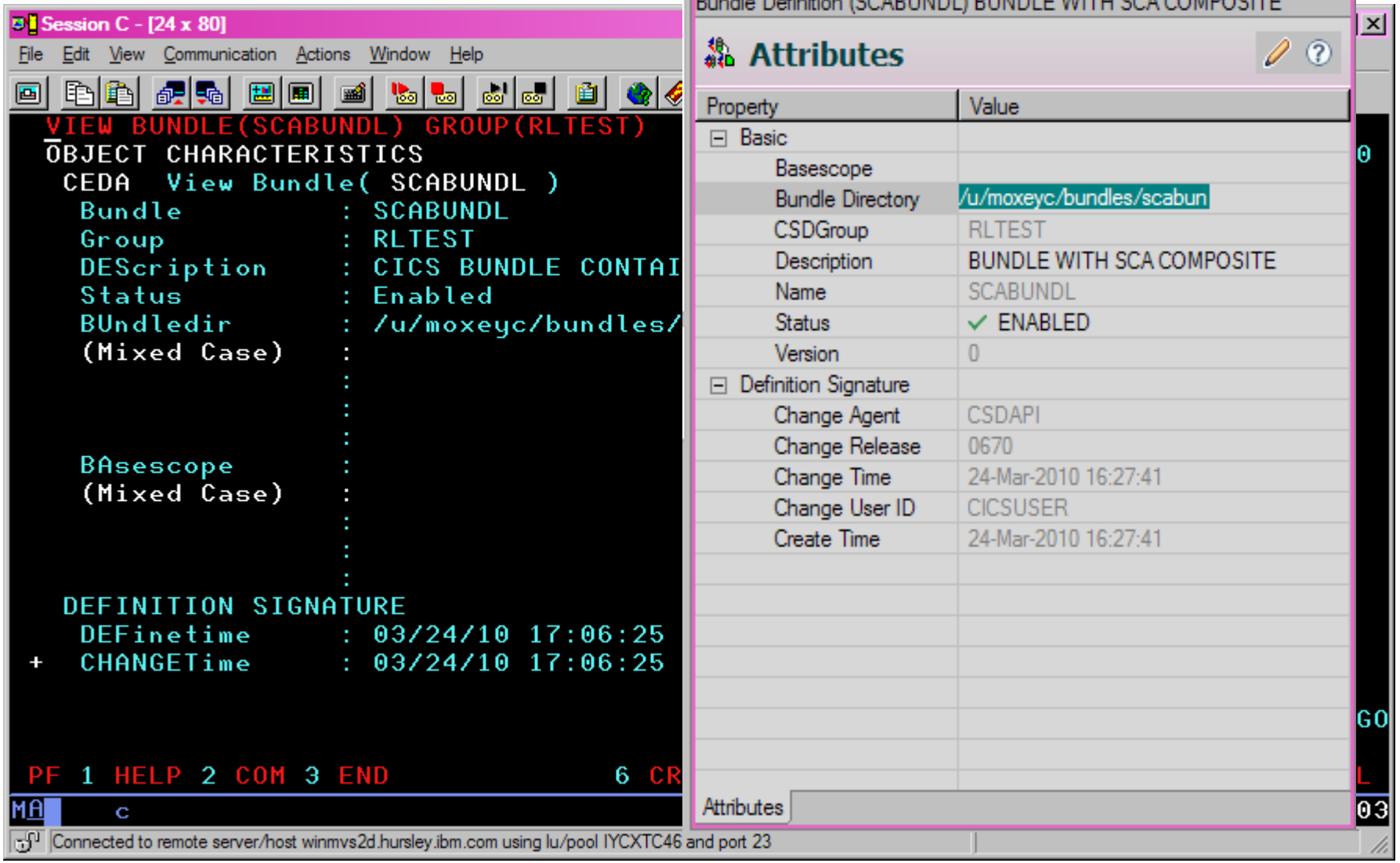
Service Component Architecture – Development Artifacts



What are Bundles?

- **Similar in concept to OSGI bundles for Java / Eclipse / WAS**
- **Provide a deployment and life cycle grouping for related application artefacts**
  - Provides a single point of management and control
  - The artefacts can be from a number of resource spaces
- **Allow such a grouping to express and police its dependencies**
  - Can express functional or resource related dependencies
- **Extensible**
  - Provide an extension point for Vendor or User artefacts to be deployed and managed alongside CICS Resources
  - Manifest File describes contents - “Imports”, “exports” , “defines”
  - User extensible via Callback program
- **CICS Resources which are “bundle-enabled”:**
  - Event Binding, XSD Bindfile, SCA Composite

BUNDLE Resource



Bundle Contents

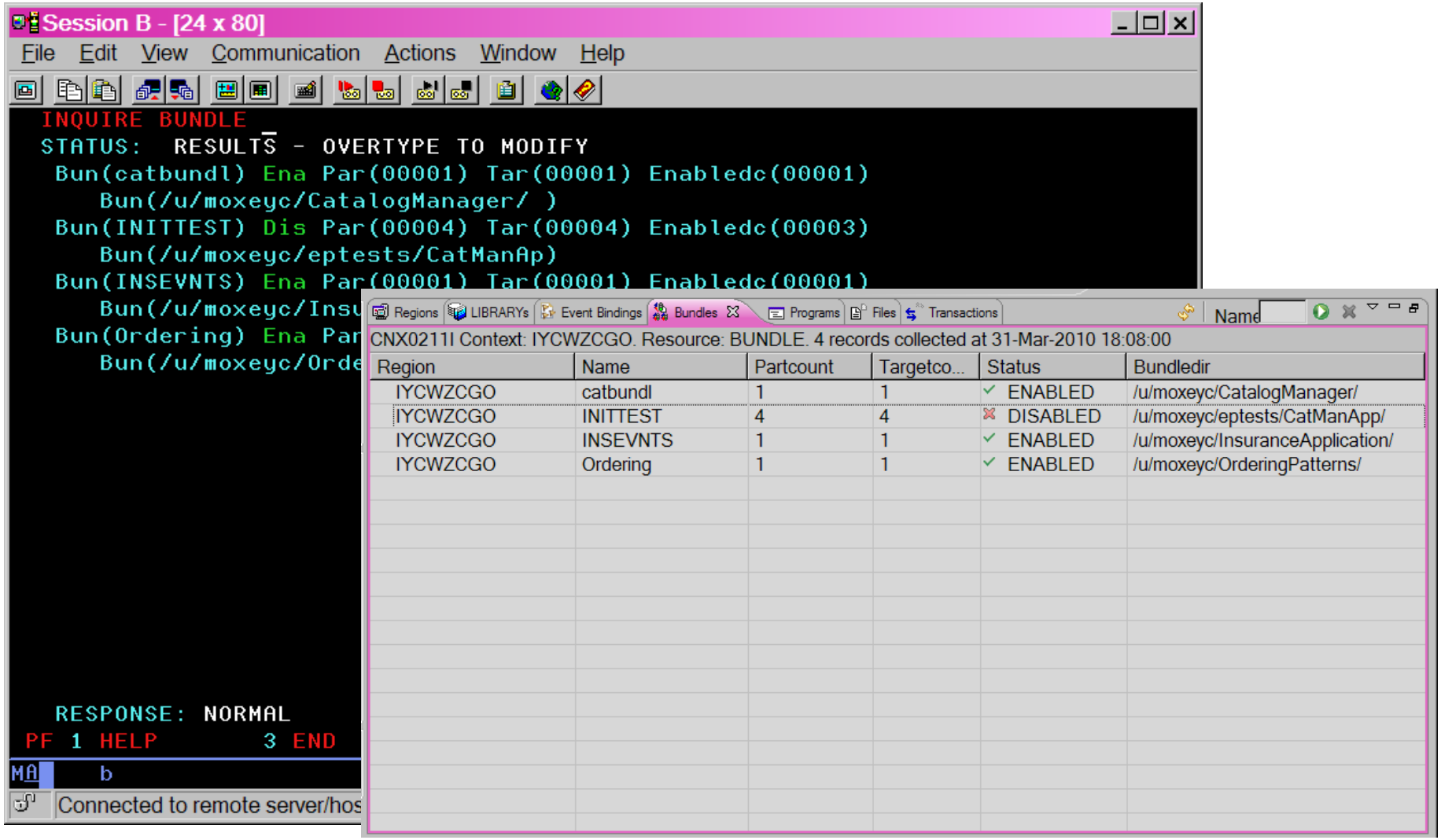
.../scabun
    /META-INF
        cics.xml ← Bundle manifest
    /scaproject
        testcomposite.scdl

cics.xml - Notepad

File Edit Format View Help

```
<?xml version="1.0"?>
<tns:manifest xmlns:tns="http://www.ibm.com/xmlns/prod/cics/bundle"
        bundleVersion="1" bundleRelease="0">

        <tns:define     name="MyComposite"
                        type="http://www.ibm.com/xmlns/prod/cics/bundle/SCACOMPOSITE"
                        path="scaproject/testcomposite.scdl" />

        <tns:import     name="PAYROLL"
                        type="http://www.ibm.com/xmlns/prod/cics/bundle/PROGRAM" />

        <tns:import     name="TaxQuery"
                        type="http://www.ibm.com/xmlns/prod/cics/bundle/WEBSERVICE" />

</tns:manifest>
```

Bundle Operations



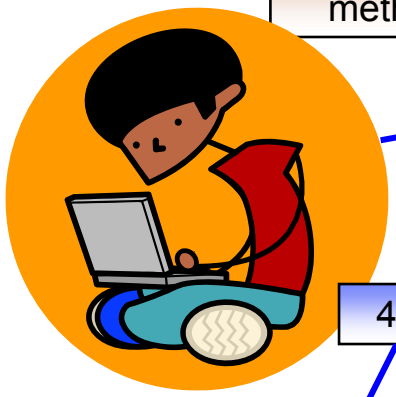
XML to language structure mapping services

- **New API to convert between XML and application data**
  - Map between XML and language structure
- **EXEC CICS TRANSFORM TRANSFORMTYPE(XMLTODATA ¦ DATATOXML)**
- **Command options depend on the direction of the transformation**
  - XMLTRANSFORM resource provides XML binding and schema used for the transformation
    - Required for DATATOXML, optional for XMLTODATA (depending on whether transforming or querying XML)
  - XMLTRANSFORM resource installed via Bundle support
- **CICS XML Assistant**
  - Batch utilities that transform XML into high-level language structures and vice versa
  - Generate metadata in XML bind file, stored on z/OS UNIX

WS-Addressing goals

- **Defines transport-neutral mechanisms to address Web services and support message transmission through networks**
- **Improves interoperability with other Web Services implementations such as .NET**
- **XML elements to identify Endpoints: EndpointReferences (EPRs)**
  - More than just a URI
  - Can have Reference Parameters and metadata
  - Allows for Psuedo-Conversational style web service requesters in CICS
- **WS-Addressing Message Addressing Properties (MAPs)**
  - Standard placeholders in the SOAP header for WS-Addressing information
  - Plus reference parameters in target EPR

Example of a WS-Addressing Resource Access Pattern



1. New shopper invokes a factory method to get a shopping cart

2. Service creates new EPR

3. Add Reference Parameter containing cart ID. Eg. Cart A

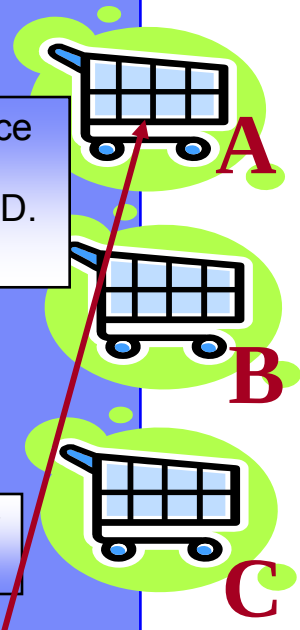
4. Return EPR in SOAP Body

5. This EPR is then used for subsequent invocations

6. IWS op=update

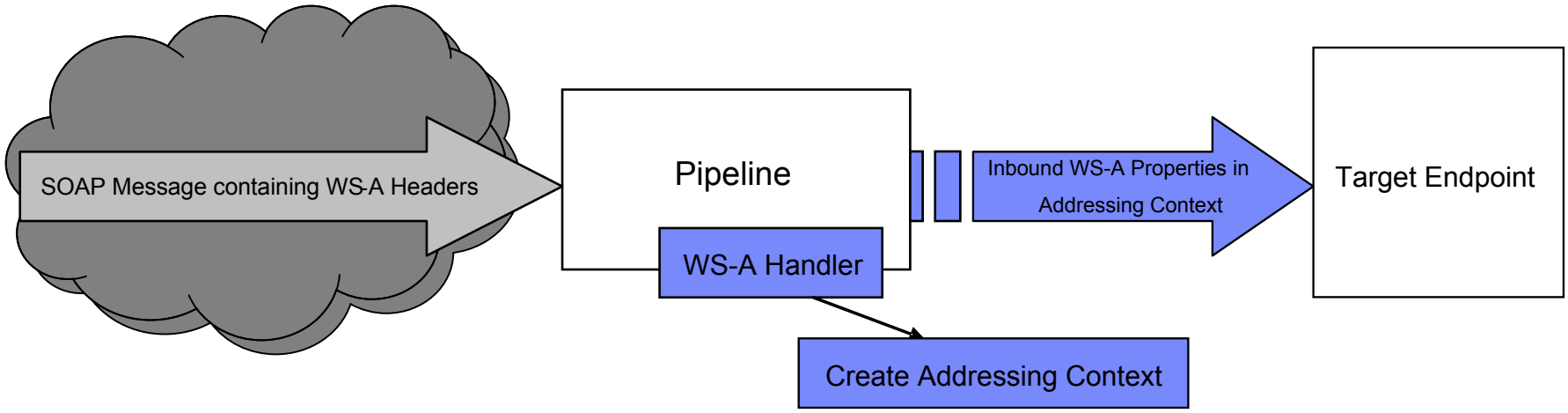
7. Service gets Reference Parameter From Addressing Context

8. Service knows which cart to update




```
<wsa:EndpointReference>
  <wsa:Address>www.market.com</>
  <wsa:ReferenceParameters>A</>
</wsa:EndpointReference>
```

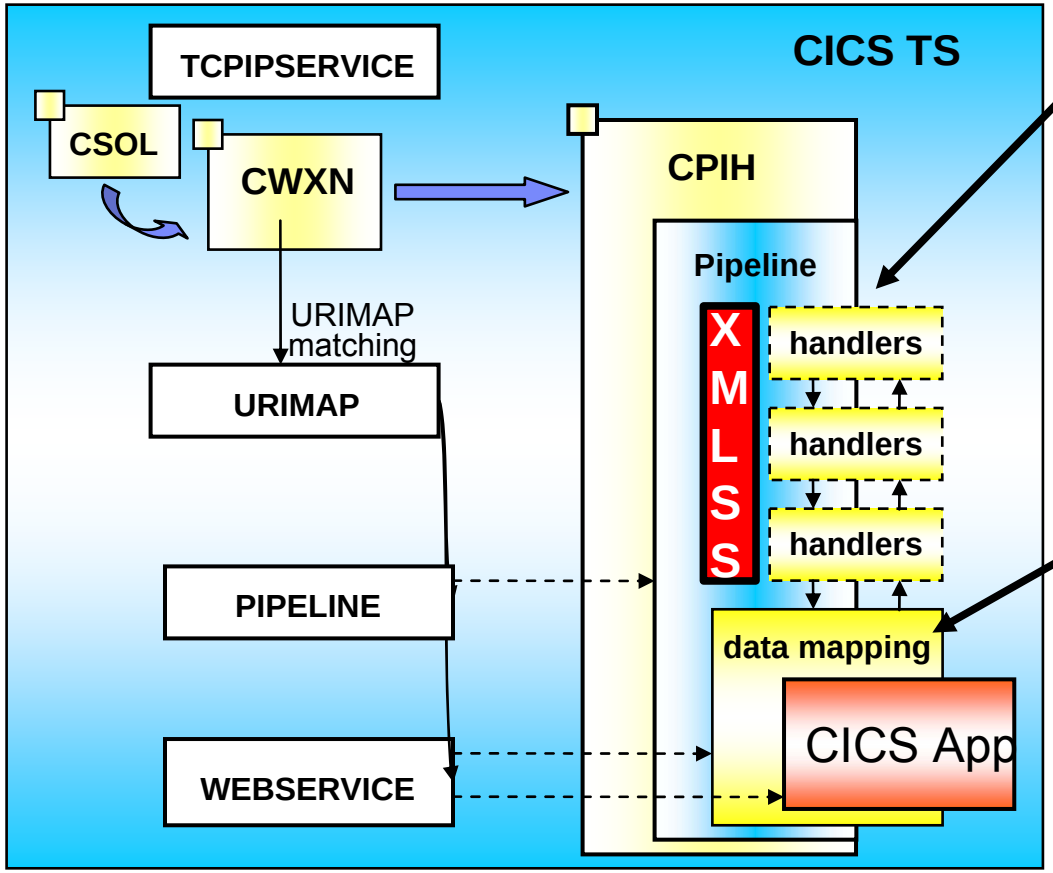
WS-Addressing in CICS



WS-Addressing in CICS

- **Pipeline configuration**
  - Configure Requester pipeline to use WS-Addressing handler giving specification version
  - Configure Provider pipeline to use WS-Addressing handler
- **Requester**
  - Requester application is not aware requests are WS-Addressed
    - CICS handles the required addressing responses
  - Requester is WS-A aware
    - Uses EXEC CICS API to create an addressing context and set Message Addressing Properties (MAPs)
- **Provider**
  - Provider application is not aware request/response is WS-Addressed
    - CICS handles the required addressing responses and routing
  - Provider is WS-A aware
    - Uses EXEC CICS API to interrogate Addressing Context and set MAPs (e.g. Get EPR to extract Reference parameters)
    - ReplyTo or FaultTo EPR used for reply endpoint (default is anonymous address to reply back to the requester)

CICS and z/OS XML System Services Parser (XMLSS)



- **First parse of message now uses XMLSS**
  - Locates the SOAP headers
  - Handler execution is outside of XMLSS
- **XMLSS is zAAP eligible**
  - Offloads MIPS for this element of the processing
- **"Shredding" of body into Containers or Commareas is *NOT* XMLSS**
  - and so ***NOT*** zAAP eligible.

Summary

- **Revitalize your applications using modern practices**
  - Package and deploy applications in standard way with Service Component Architecture
  - Exploit new CICS capabilities with Bundles
  - Improve interoperability with WS-Addressing
  - Reduce licence fees with zAAP offload of XML Transforms

Resources

For more information

CICS Development Technical Services

Engage the IBM Hursley CICS development team to ensure that you get the maximum value from your CICS investments.

Consultants are now available via a funded services engagement directly from CICS development providing a complete range of CICS services - no one has more experience!

- **For more information please contact: CICSDTS@uk.ibm.com**

CICS On Demand Seminars

Free customised technical agenda of CICS TS and CICS tools products

- Web Services, Events, Web 2.0, version upgrades, tooling etc.

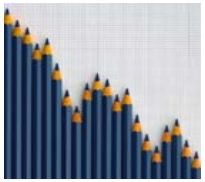
Targeted at AD, System Programmers and Architects at customers' own location

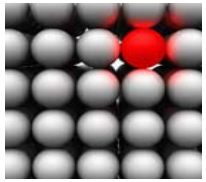
Demonstrates how to use the new features in the latest releases to leverage existing solutions

Can be tailored to customers interests

- **For more information please contact: cicssem@uk.ibm.com**

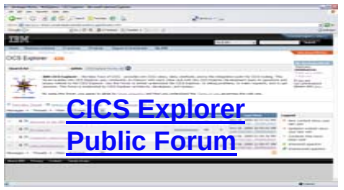





CICS Communities and Information

- **CICS Transaction Server V4.1**
  - http://ibm.com/cics/tserver/v41/
- **CICS Explorer**
  - Home page ibm.com/cics/explorer
  - Forum http://tinyurl.com/68bndw
- **Twitter**
  - Subscribe to the IBM_System_z channel & CICSfluff channel to get CICS news flashes
- **CICS Blog - Comment and opinion**
  - TheMasterTerminal.com
- **CICS eNews**
  - Subscribe for news about CICS and related products
- **CICS Links regular updates all in a single presentation deck**
- **YouTube channels**
  - CICS Explorer - Videos, demos and other cool stuff
  - CICSFluff - Other CICS videos



Thank You !