

E19

# IMS Connector for Java: Details you need to Know!

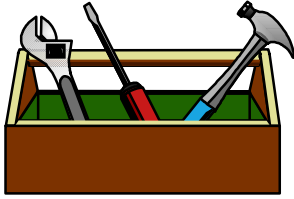
Judith Hill



St. Louis, MO

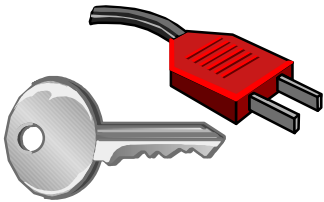
Sept. 30 - Oct. 3, 2002

# Agenda



## Application Development

- Developing service-oriented applications
- Conversational applications
- Transactions
- Security
- Exception handling



## Runtime Management

- Topology
- Connection management
- Setting Trace for Problem Analysis

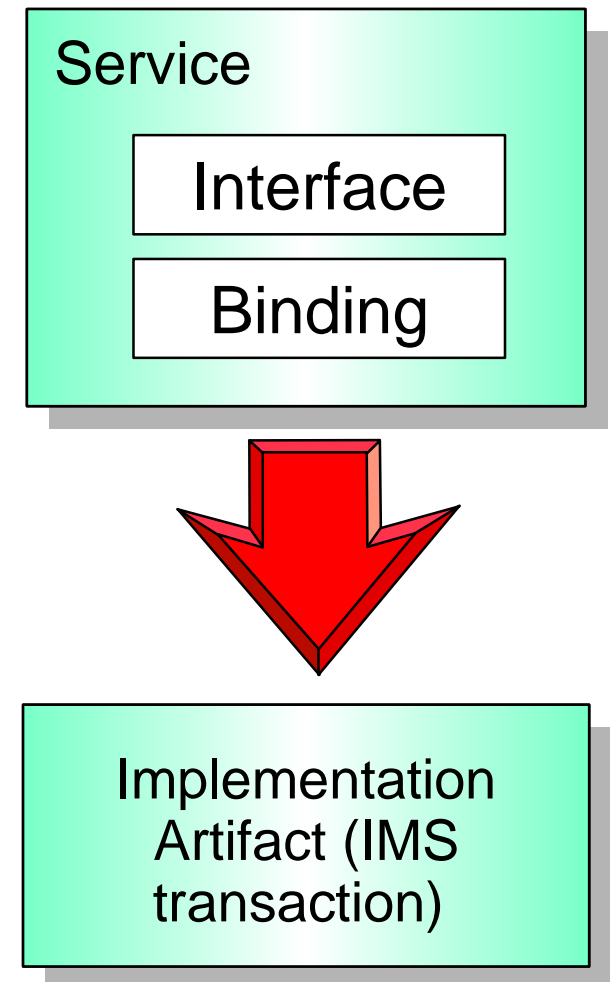


## Migration/Coexistence

- VisualAge for Java
- WebSphere Studio Integration Edition

# Enterprise Services

- WSAD IE is based on the concept of **Enterprise services**
- A service consists of three parts:
  - ▶ **Implementation** artifact
    - IMS transactions, EJBs, ...
  - ▶ **Interface**
    - Describes the available *operations* and the *messages* they exchange
  - ▶ **Binding**
    - Describes how to get to the implementation artifact
- Services are described by WSDL documents

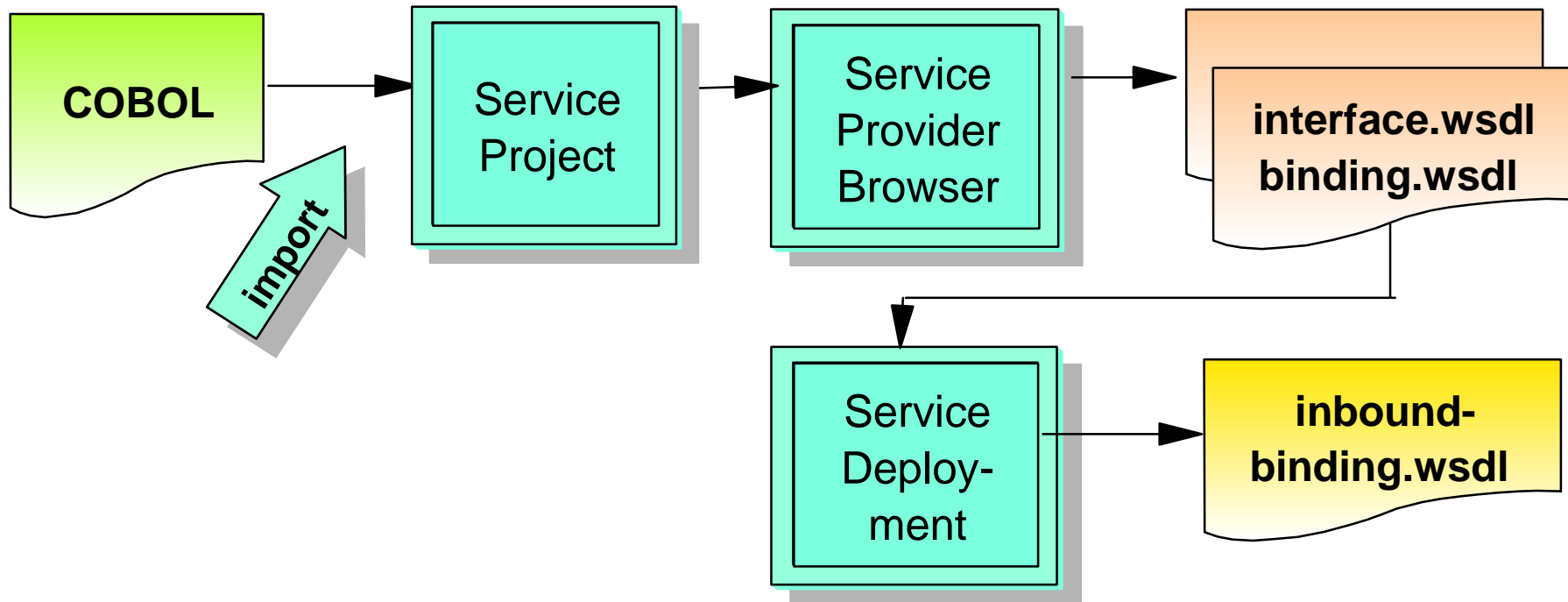


# WSAD IE Application Model

- A service-oriented application is composed of
  - ▶ **Service provider**
    - In general, server-side processing
    - Service is described in WSDL
    - Accessed by any type of Consumer
    - Accessed using a variety of protocols
  - ▶ **Service consumer**
    - In general, client-side processing
    - Uses information in WSDL to access Service provider
- WSAD IE provides tools and wizards to develop, test, and deploy service-oriented applications

# Provider/Consumer Development

- Service Provider



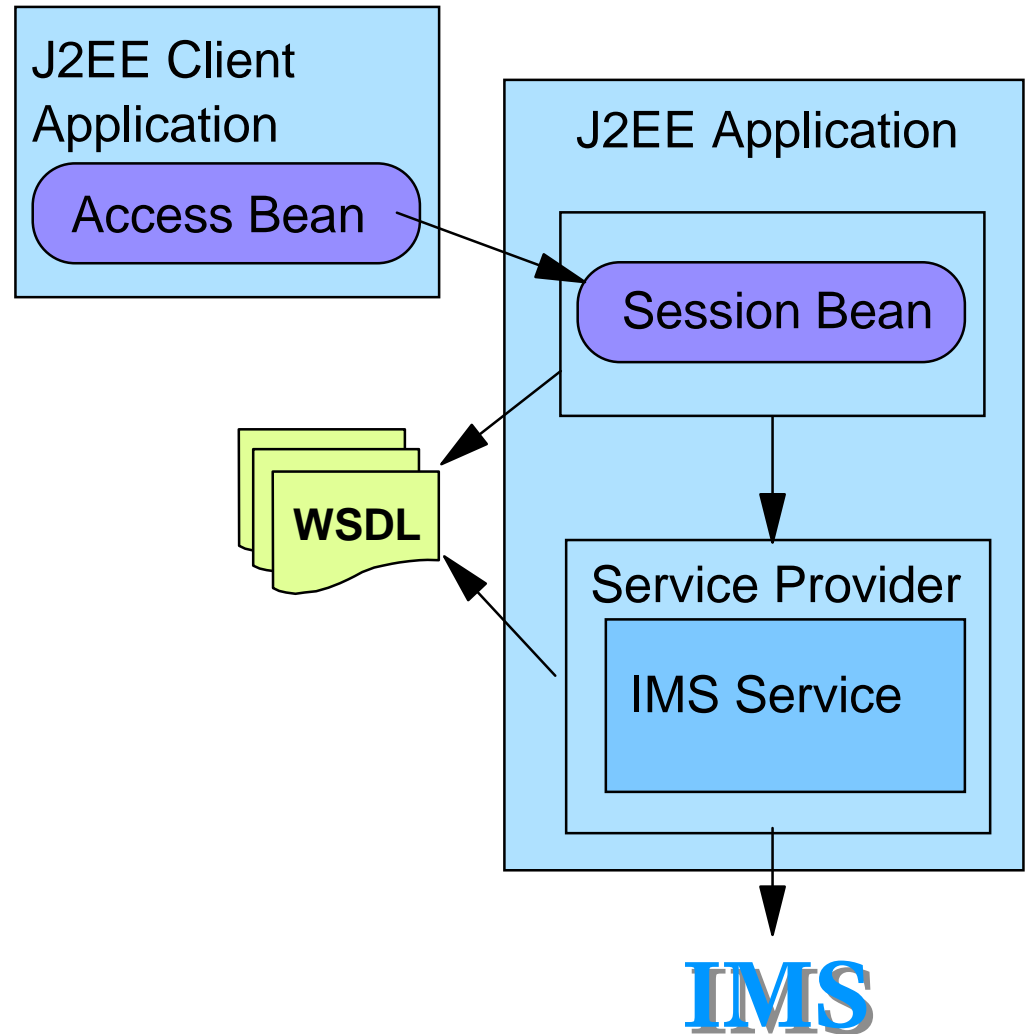
- Service Consumer



# Consumer J2EE Application

- **Service Deployment** wizard creates the Session Bean
- J2EE Client Application uses Access Bean to interact with Session Bean and thus the Service Provider
- Service Provider accesses IMS Connect/IMS using IMS Resource Adapter

Service Consumer



# Consumer Web Service Application

## ■ Service Deployment

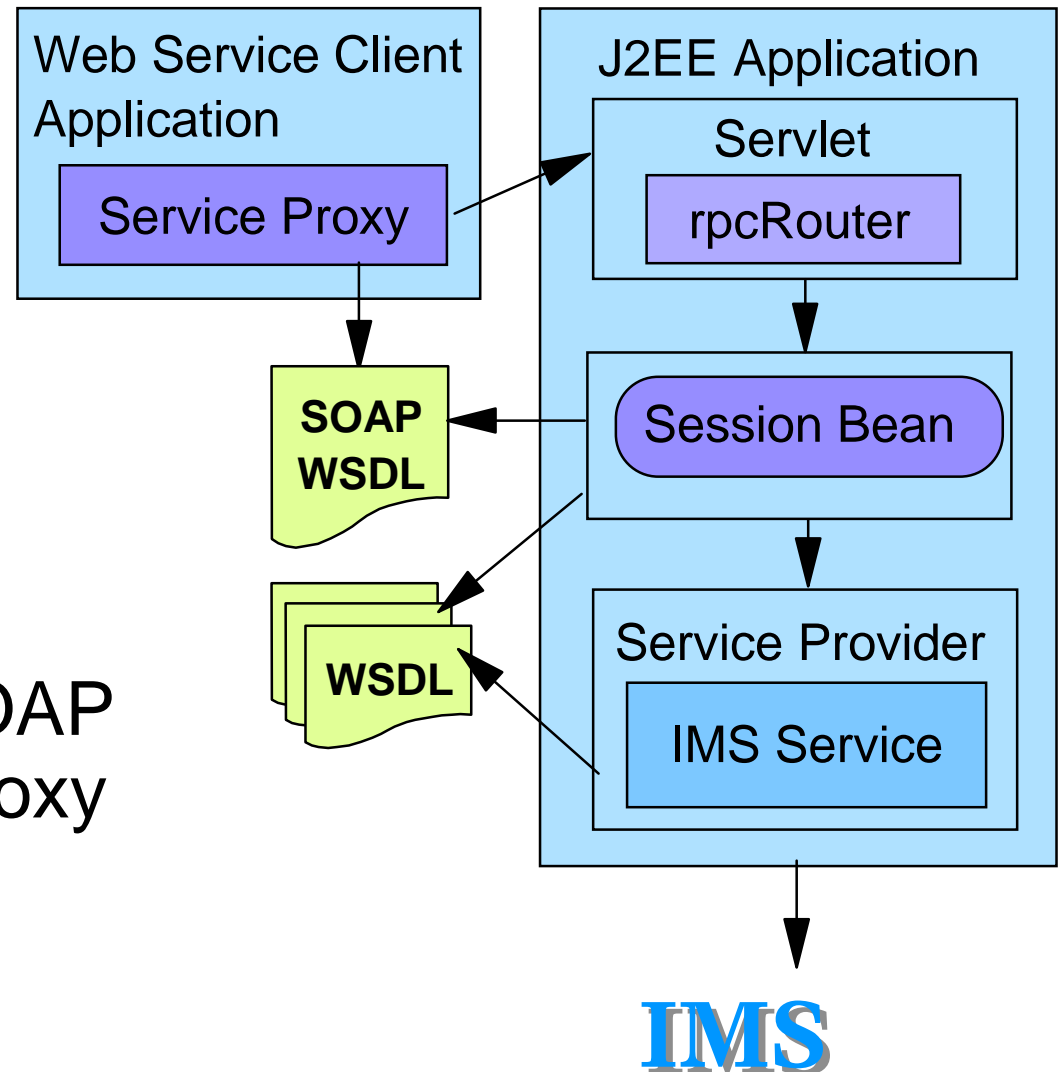
wizard creates

- ▶ Session Bean
- ▶ SOAP Servlet (rpcRouter)
- ▶ SOAP.wsdl

## ■ Service Proxy wizard creates Service Proxy

## ■ Web Service Client Application accesses SOAP service using Service Proxy

Service Consumer



# Testing Service Applications

- Testing in the **WebSphere EE v4.1 Test Environment** of WSAD IE 4.1
  - ▶ **Java "stand-alone" applications**
    - Run/Debug (Test Environment server is not used)
  - ▶ **J2EE applications and Web Service applications**
    - Create a Server Instance and Configuration
    - Add Enterprise Application project to Server Configuration
    - Add the IMS Resource Adapter
    - Add a J2C IMS Connection Factory
      - Specify JNDI name
      - Specify properties (host name, port, max connections, etc.)
    - Start the Server
    - Run/Debug



Service perspective

IMS Resource Adapter

The screenshot displays the 'Server - Application Developer Integration Edition' interface. The 'Navigator' on the left shows a project named 'IMSSample' with sub-projects 'IMSSampleEAR', 'IMSSampleEJB', and 'IMSSampleWeb'. The 'Server Configuration' view at the bottom left shows 'myServer' with the deployed application 'IMSSampleEAR'. The main workspace shows the configuration for 'myServer' in the 'J2C' tab. It includes a table for 'J2C Resource Adapters' with one entry: 'IMS' (IMS Connector for Java, IBM Corporation, 1.2.4). Below it is a table for 'J2C Connection Factories' with one entry: 'cf\_myIMS' (myIMS). The 'Resource Properties' section lists: HostName (java.lang.String, CSMDEC13.STL.IBM.COM), PortNumber (java.lang.Integer, 9999), DataStoreName (java.lang.String, SOCKEYE), and UserName (java.lang.String). The 'Servers' view at the bottom right shows 'myServer' is 'Stopped' with the message 'The server should be republished'. Red arrows point from the text labels to the corresponding elements in the interface.

Resource Adapter name	Display name	Vendor name	Spec version
IMS	IMS Connector for Java	IBM Corporation	1.2.4

Name	JNDI name	Description
cf_myIMS	myIMS	

Name	Type	Value
HostName	java.lang.String	CSMDEC13.STL.IBM.COM
PortNumber	java.lang.Integer	9999
DataStoreName	java.lang.String	SOCKEYE
UserName	java.lang.String	

Server Instance	Server Configuration	Status	Server State
myServer	myServer	Stopped	The server should be republished

J2C Connection Factory

Deployed application

# Going into Production

- Deploying on **WebSphere Application Server Enterprise Edition v4.1**
  - ▶ Export application out of WSAD IE as an Enterprise Archive (EAR) file
  - ▶ Install application (EAR) into WebSphere Application Server
  - ▶ Install the IMS Resource Adapter
    - Resource Archive (RAR) file is included in IMS Connector executable
  - ▶ Configure a J2C IMS Connection Factory
    - Specify JNDI name
    - Specify properties (host name, port, max connections, etc.)
  - ▶ Start the Server
  - ▶ Run

# Common Client Interface (CCI)

- J2EE Connector Architecture defines a set of interfaces and classes used by the client to interact with an EIS

```
// Establish a JNDI context
Context ic = new InitialContext();

// Use JNDI to lookup a ConnectionFactory for the EIS system
ConnectionFactory cf = (ConnectionFactory) ic.lookup("java:comp/env/eis/MyIMS");

// Obtain a connection to the EIS
Connection conn = cf.getConnection();

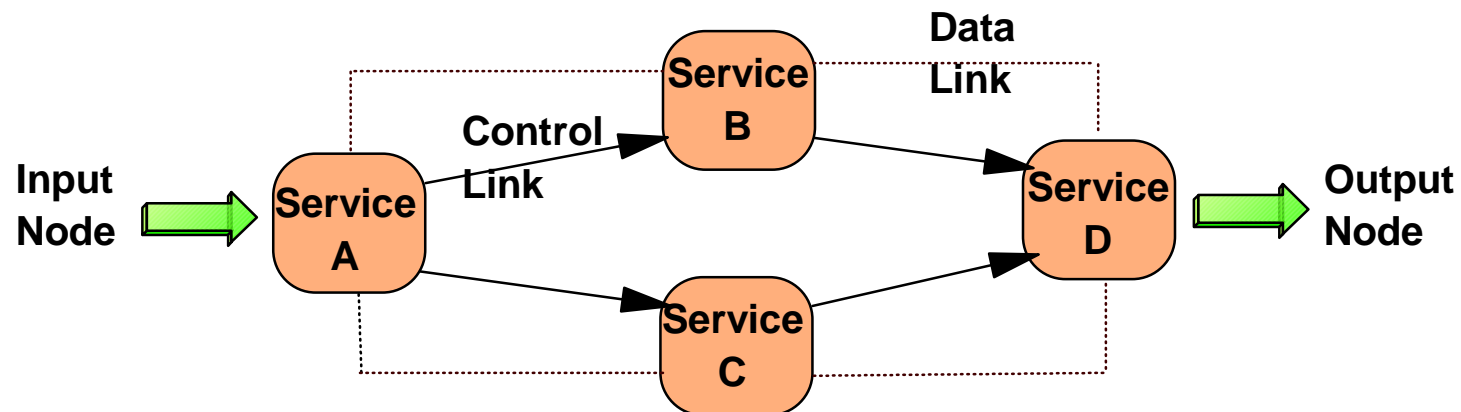
// Create an interaction to execute an EIS function
Interaction interaction = conn.getInteraction();

// Specify the properties for the interaction
IMSInteractionSpec iSpec = new IMSInteractionSpec();
iSpec.setInteractionVerb(SYNC_SEND_RECEIVE);
...
// Create Input Record and set the input values
...
// Create Output Record
...
// Invoke an EIS function (e.g. run an IMS transaction)
interaction.execute(iSpec, input, output);

// Close the interaction and connection
interaction.close();
conn.close();
```

# Service Flows

- WSAD IE 4.1 supports **micro flows**
  - ▶ Analogous to VisualAge for Java's Navigators
  - ▶ Short lived and synchronous
- Multiple services (activities) combined to form a single service (business process)
  - ▶ Activity nodes (services) are combined in specific order
    - Control Links
  - ▶ Data can be mapped/converted between activity nodes
    - Data Links



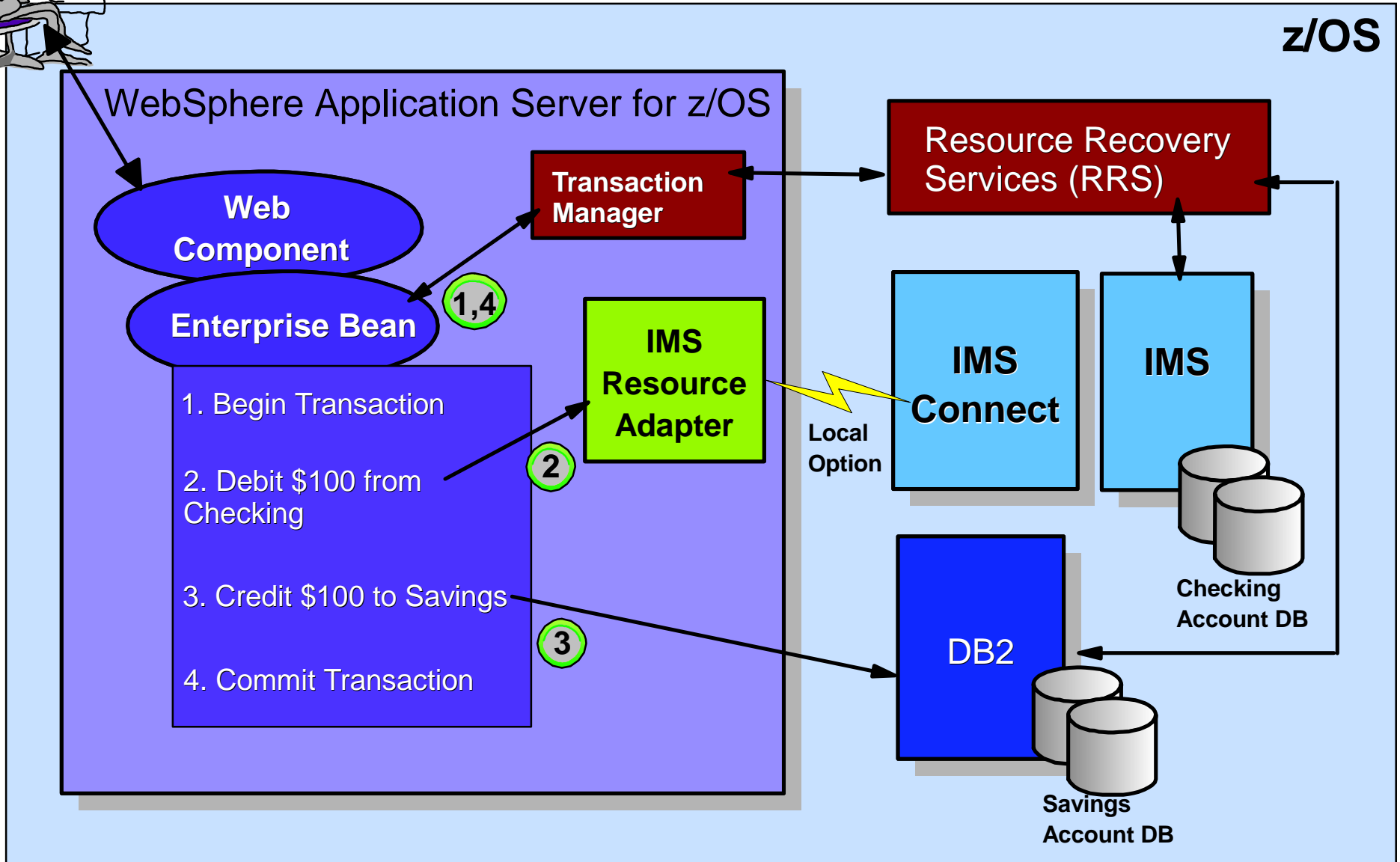
# IMS Conversations

- Programming Models
  - ▶ Stateful session bean
    - See IMS Connector for Java User's Guide
    - VisualAge for Java conversational applications run in Integration Edition in coexistence mode
  - ▶ WSAD IE 4.1 micro-flows
    - For simple IMS conversations
    - Presented in session [H03](#)
  - ▶ Strategic direction for IMS conversations is work-flows
    - Available in a future version of Integration Edition

# IMS Connector for Java Transaction Support



Transfer \$100 from  
Checking to Savings



# What is a Transaction ?

- A transaction is a group of changes that are either fully completed or fully rolled back
- J2EE offers two options on how to set the transaction boundaries in your application
  - ▶ Bean-Managed
  - ▶ Container-Managed

# Bean-Managed Transaction

- Can be used in EJB and Servlet applications
- The application code sets the transaction boundaries
  - ▶ Uses Java Transaction API (JTA) interface to demarcate the transaction

```
void methodA() {  
  
    java.transaction.UserTransaction transaction =  
        ejbContext.getUserTransaction();  
  
    // Indicates the beginning of the group of changes  
    transaction.begin();  
    :  
    // Make changes to the protected resource(s)  
    interaction.execute(interactionSpec, input, output);  
    :  
    // Commit the changes  
    transaction.commit();  
  
}
```

Transaction begins →

Transaction ends →



# Container-Managed Transaction

- Can be used in EJB applications
- EJB container sets the transaction boundaries
  - ▶ Simplifies application coding
  - ▶ Begins a transaction before an enterprise method starts
  - ▶ Ends a transaction before the method exits
- **Transaction attributes** are specified in the deployment descriptor of the EJB

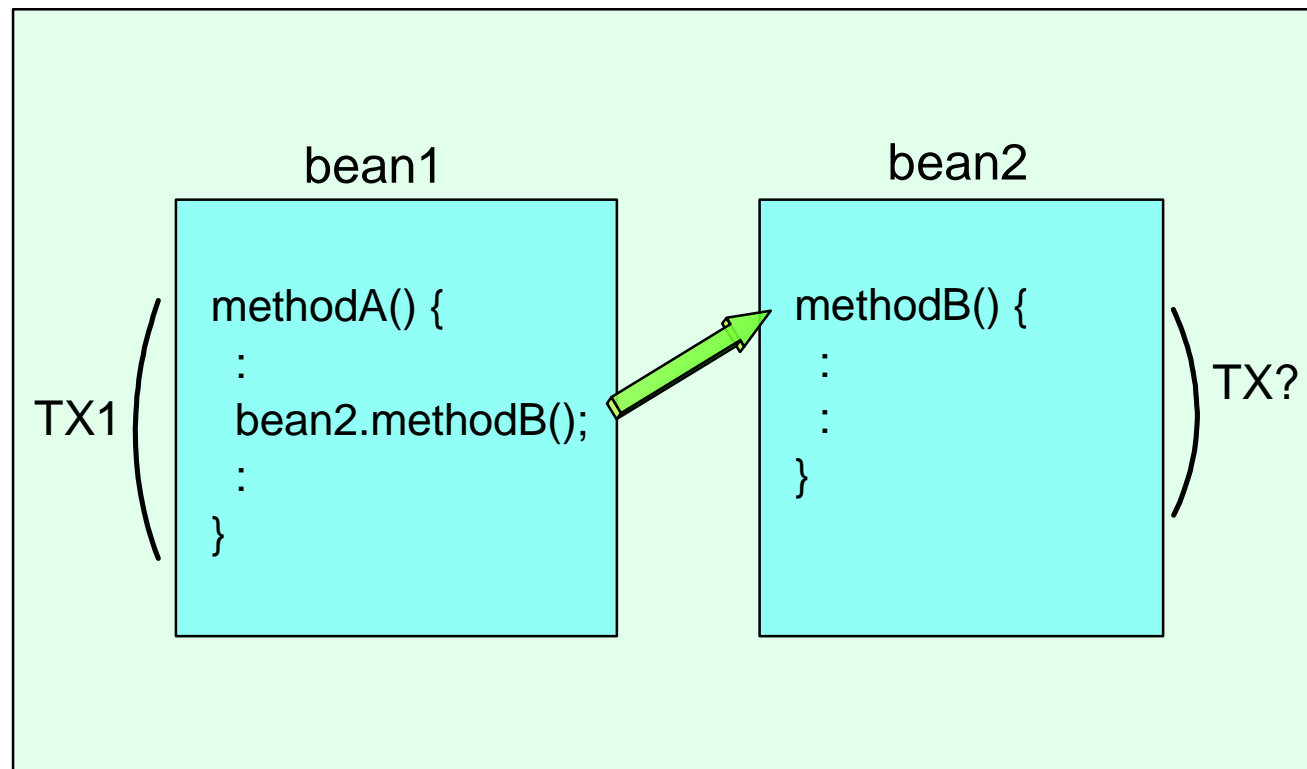
Transaction  
begins

Transaction  
ends

```
void methodA() {  
    :  
    // Make changes to the protected resource(s)  
    interaction.execute(interactionSpec, input, output);  
    :  
}
```

# Transaction Attributes

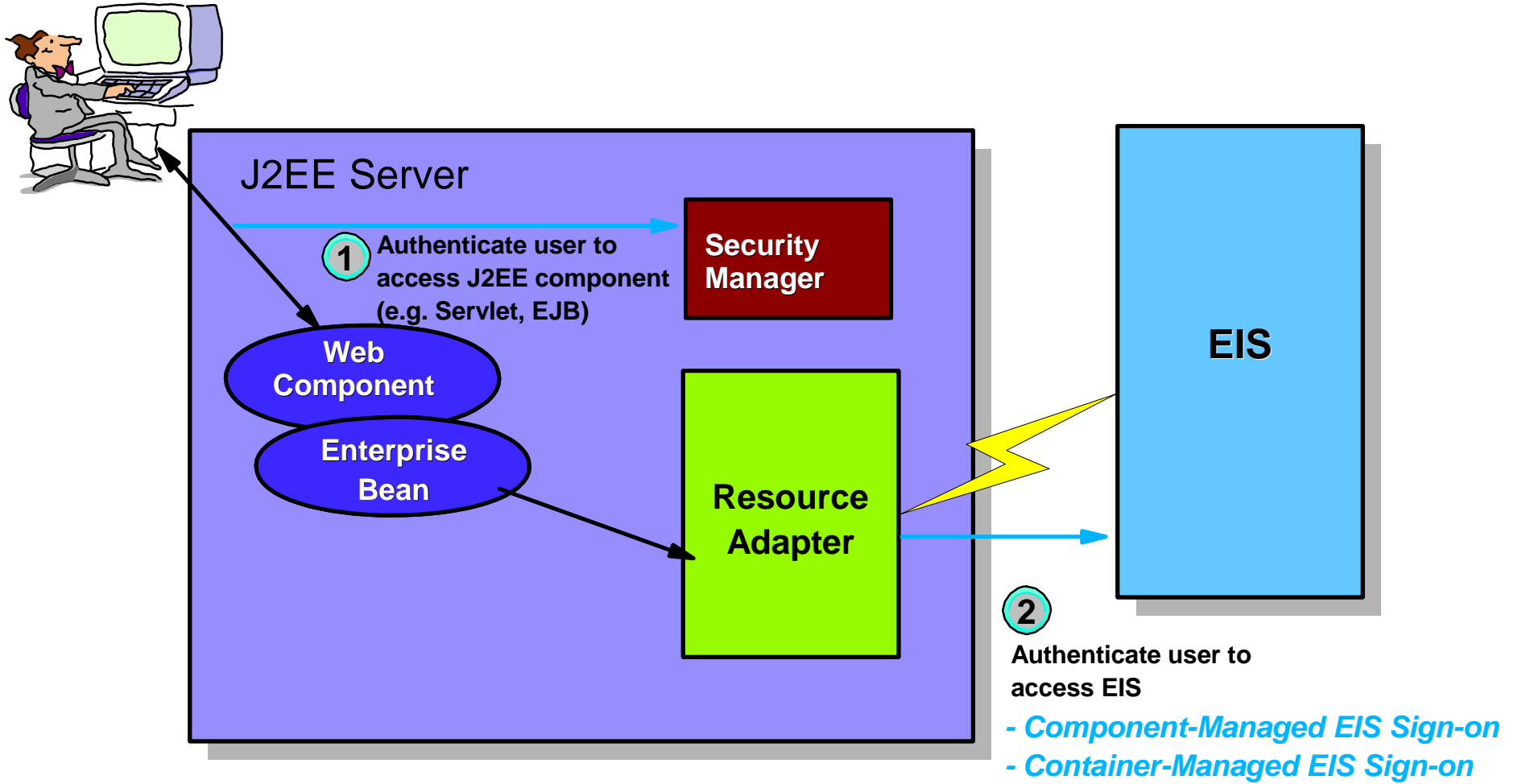
- Transaction attributes control the transaction scope
  - ▶ Required
  - ▶ RequiresNew
  - ▶ Mandatory
  - ▶ NotSupported
  - ▶ Supports
  - ▶ Never



# Considerations When Using Transaction Support

- Currently, IMS Connector for Java transaction support can only be used with Local Option
- Currently, WebSphere Application Server on z/OS, IMS Connect, and IMS must reside in the same MVS image
- MPPs involved in 2-phase commit transactions may require longer wait times
  - ▶ Additional MPPs may be needed for workload

# J2EE Security



# Component-Managed Sign-on

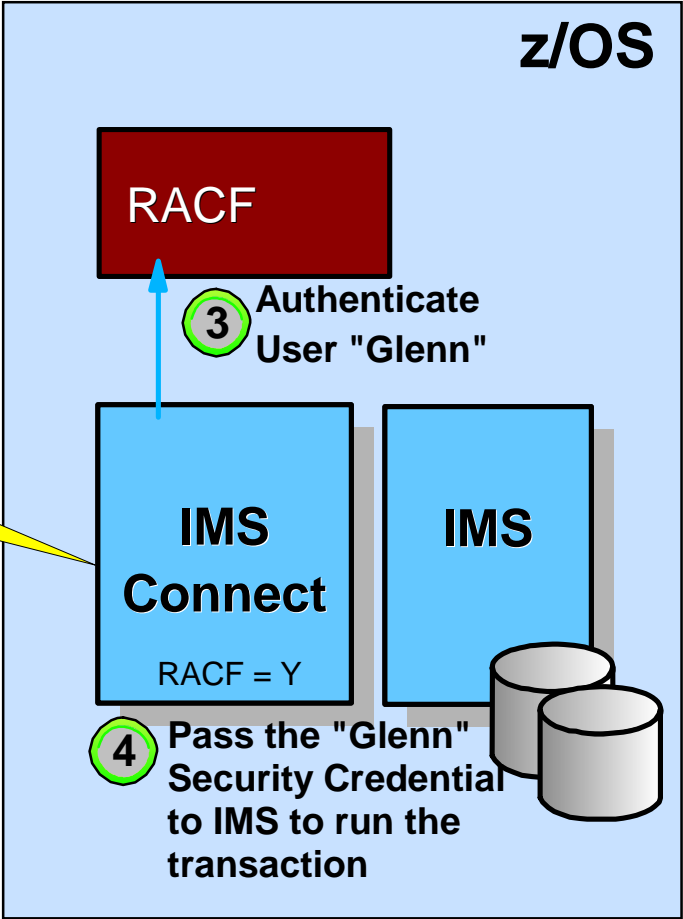
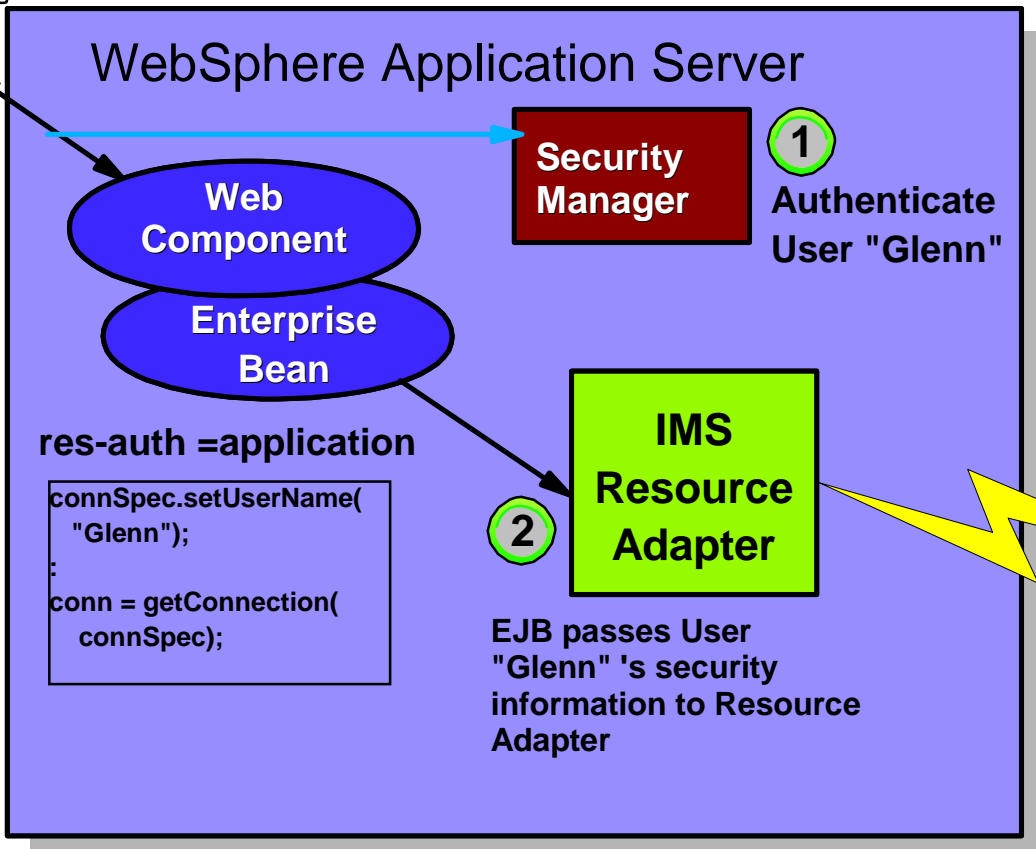
- The application component provides the security information to the resource adapter for EIS Sign-on
  - ▶ Set "res-auth = application" in the application's deployment descriptor
  - ▶ Provide the security information via the application code

```
:  
IMSCConnectionSpec connSpec = new IMSCConnectionSpec();  
connSpec.setUsername("user1");  
connSpec.setPassword("pwd1");  
  
Connection connection =  
    ConnectionFactory.getConnection(connSpec);  
:
```

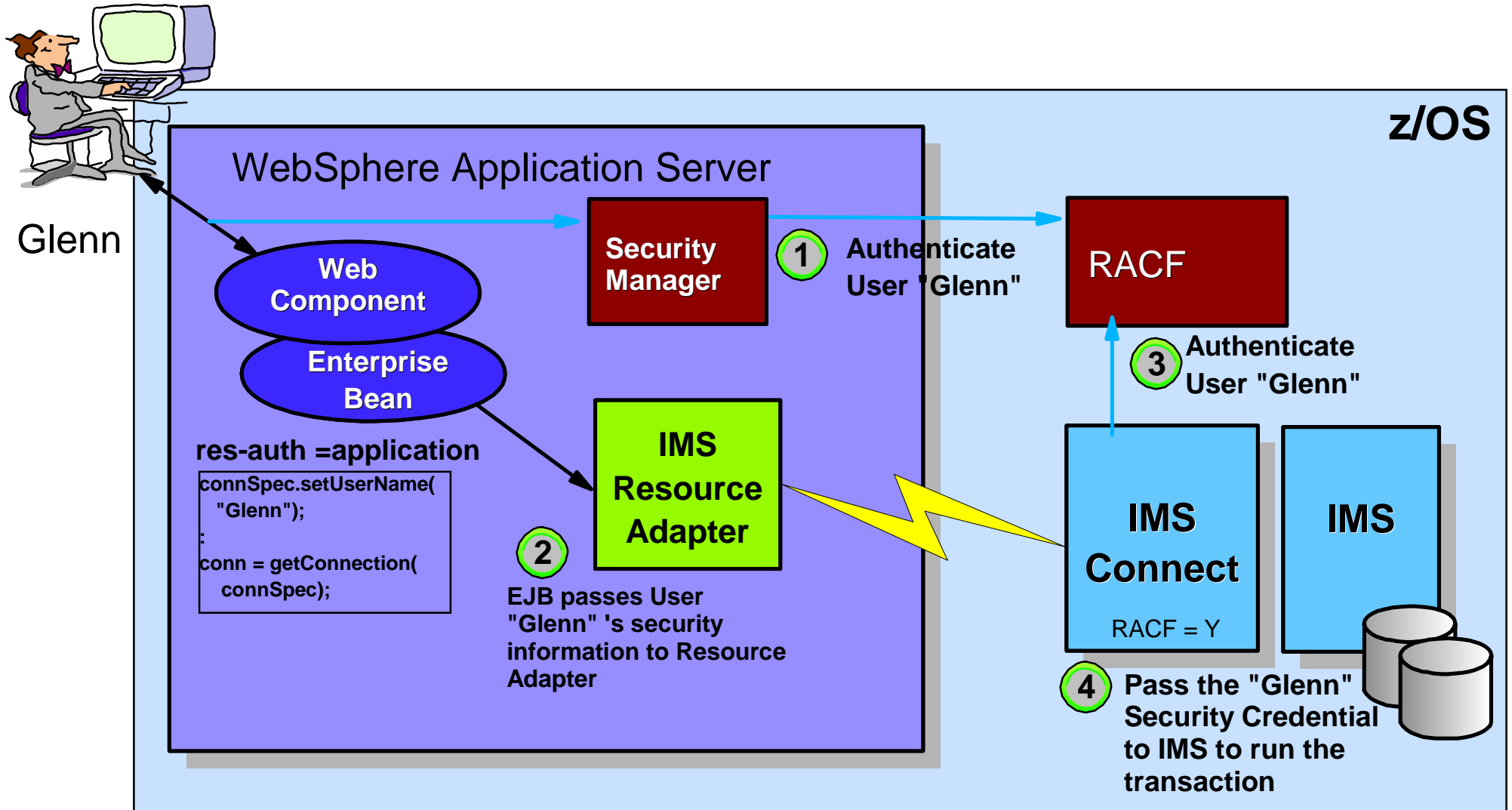
# Component-Managed EIS Sign-on Distributed



Glenn



# Component-Managed EIS Sign-on z/OS



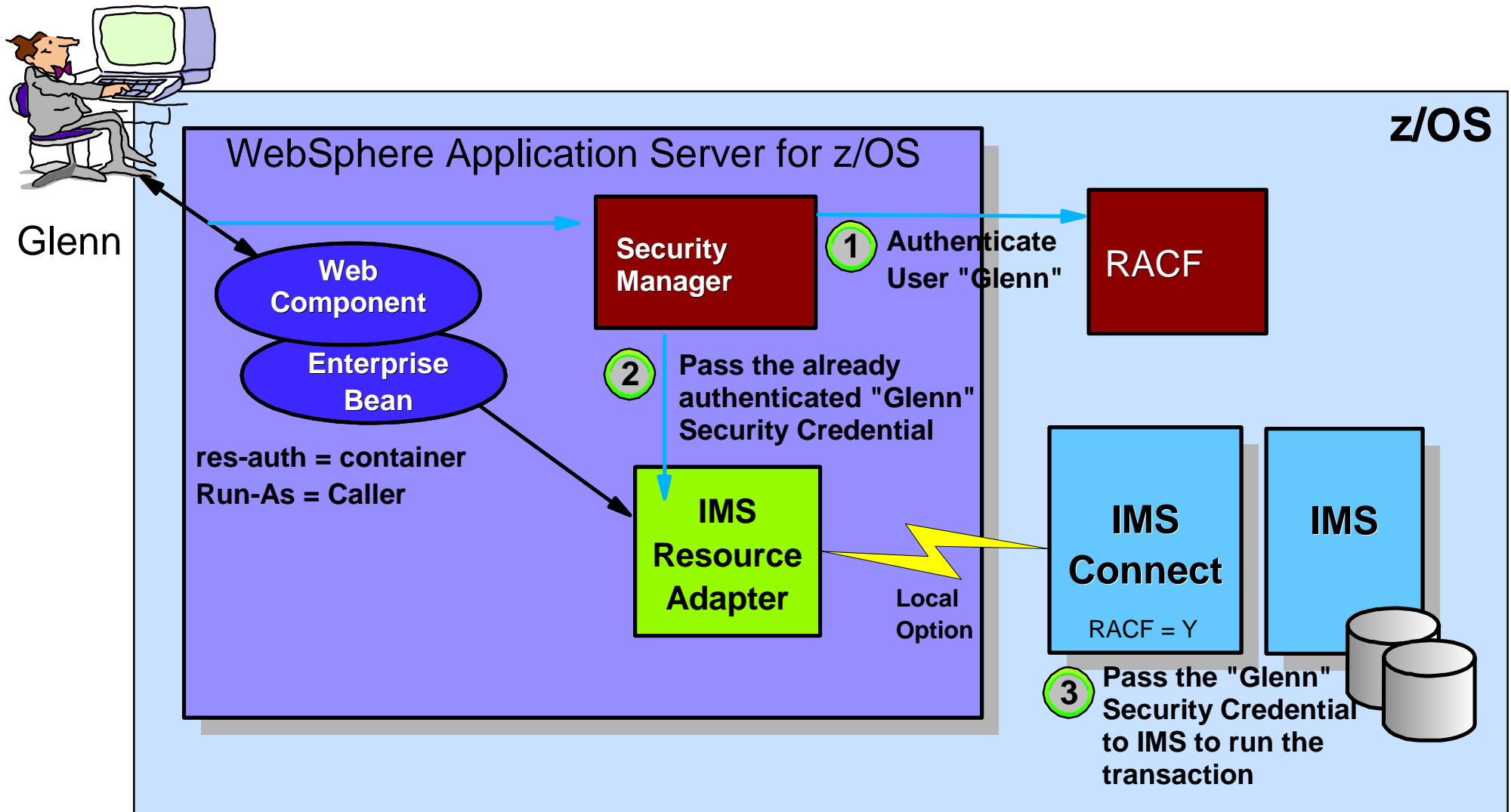
# Container-Managed Sign-on

- The container (J2EE Server) provides the security information to the resource adapter for EIS Sign-on
  - ▶ Set "res-auth = container" in the application's deployment descriptor
  - ▶ The security information is provided to the J2EE server using the J2EE server's administrative/deployment tool
  - ▶ Easy to maintain security information
  - ▶ Simplifies application coding

```
:  
Connection connection =  
    ConnectionFactory.getConnection();  
:
```



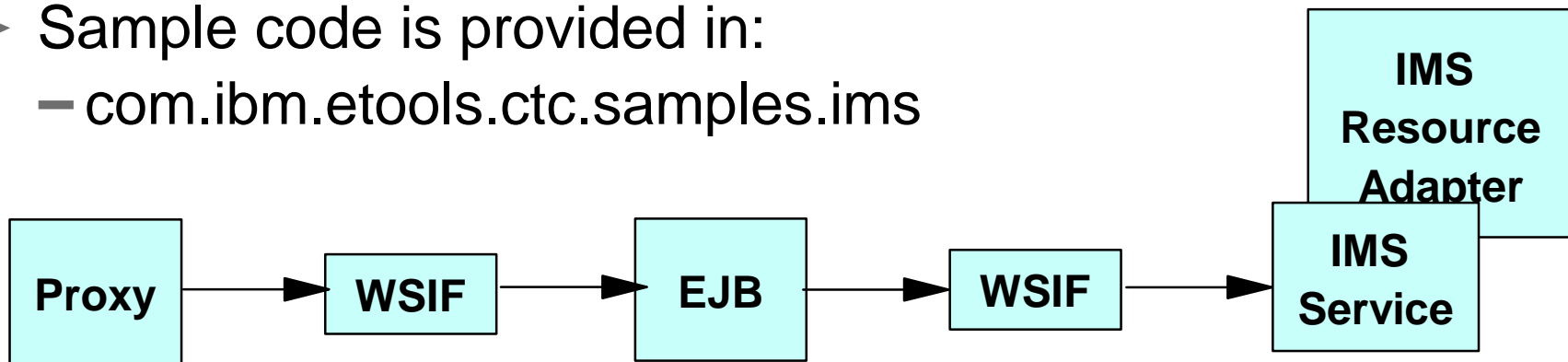
# Container-Managed EIS Sign-on

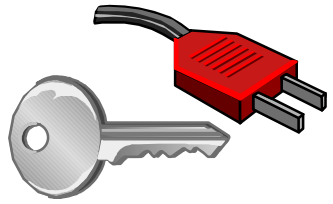


*Avoids additional RACF call in each interaction with IMS*

# Error Handling

- IMS Resource Adapter exceptions extend `javax.resource.ResourceException`
  - ▶ For example:
    - `javax.resource.spi.IllegalStateException`
    - `com.ibm.connector2.ims.ico.IMSDFSMessageException`
      - Contains "DFS" message from IMS
      - If `imsRequestType = 1`
- Your application has to "unwrap" the exception
  - ▶ WSIF "layers" wrap `ResourceException`
  - ▶ Sample code is provided in:
    - `com.ibm.etools.ctc.samples.ims`



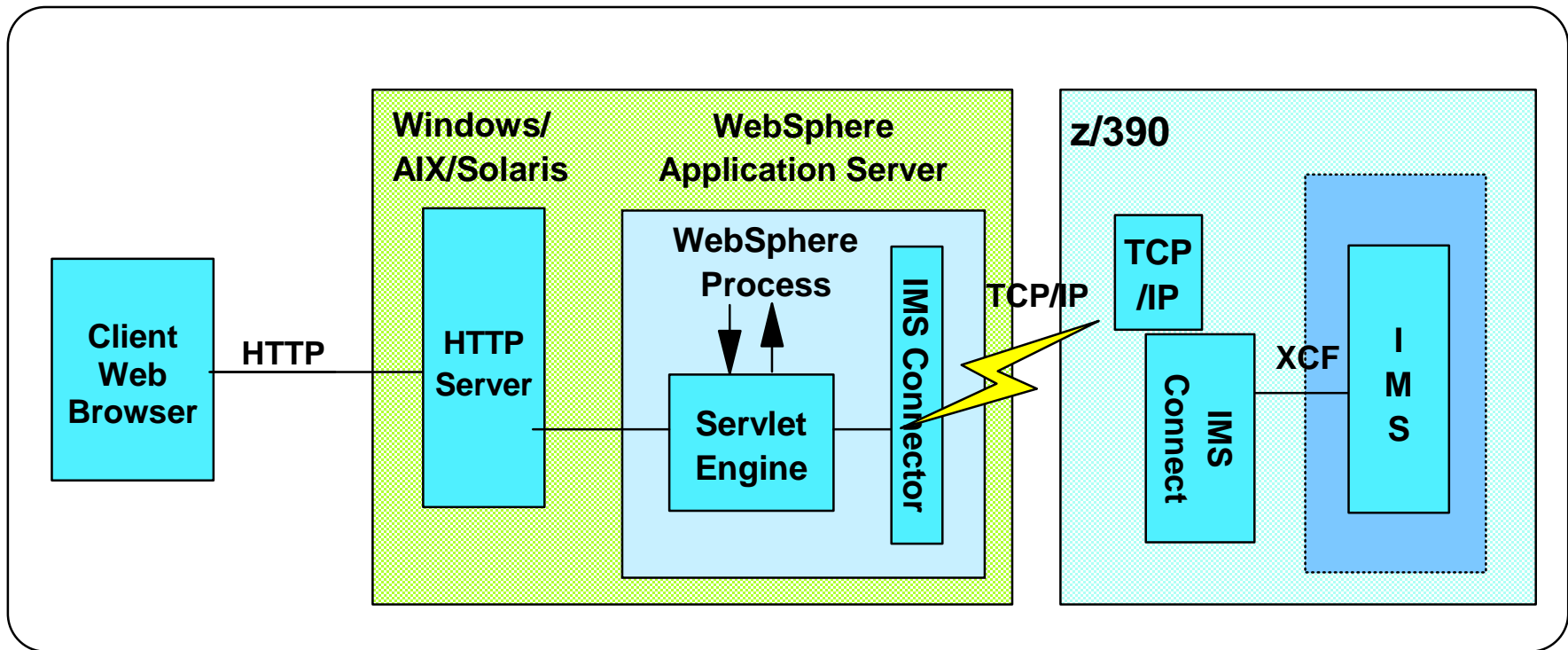


# Runtime Management

- Topology
- Connection management
- Setting Trace for Problem Analysis

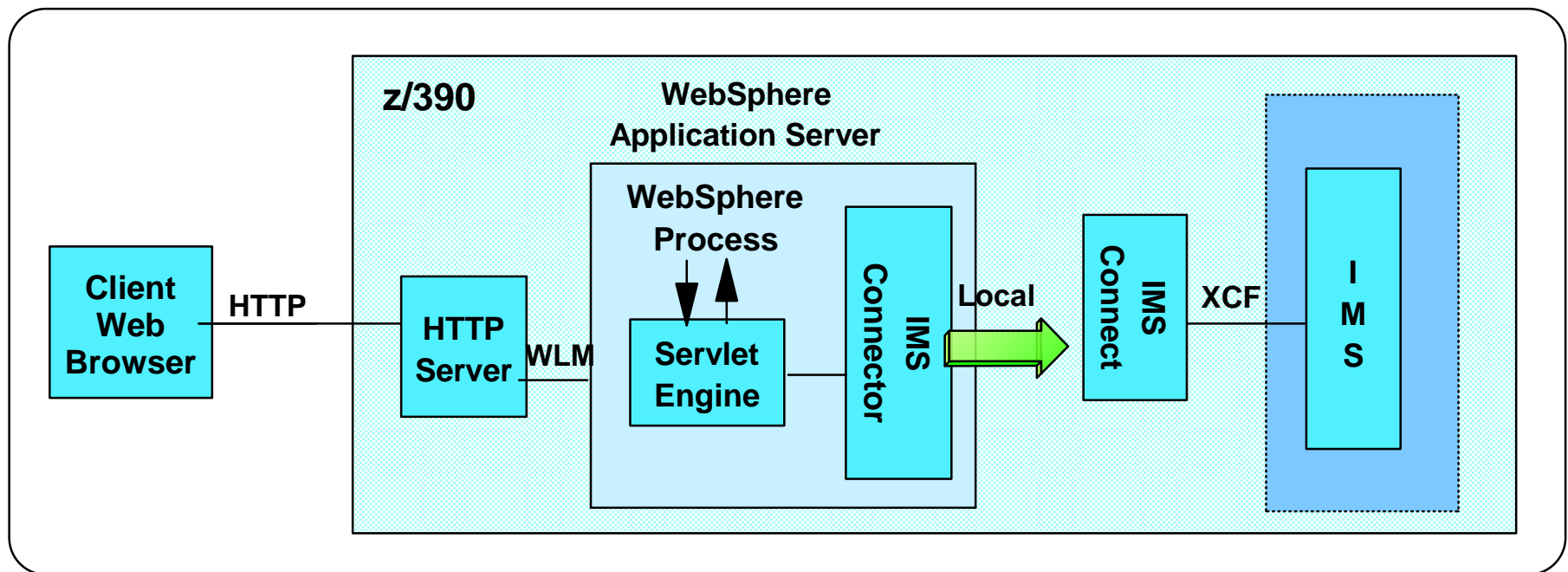
# Topology: Distributed with TCP/IP

- Provides TCP/IP communications between WebSphere Application Server on distributed platforms and IMS Connect on z/OS



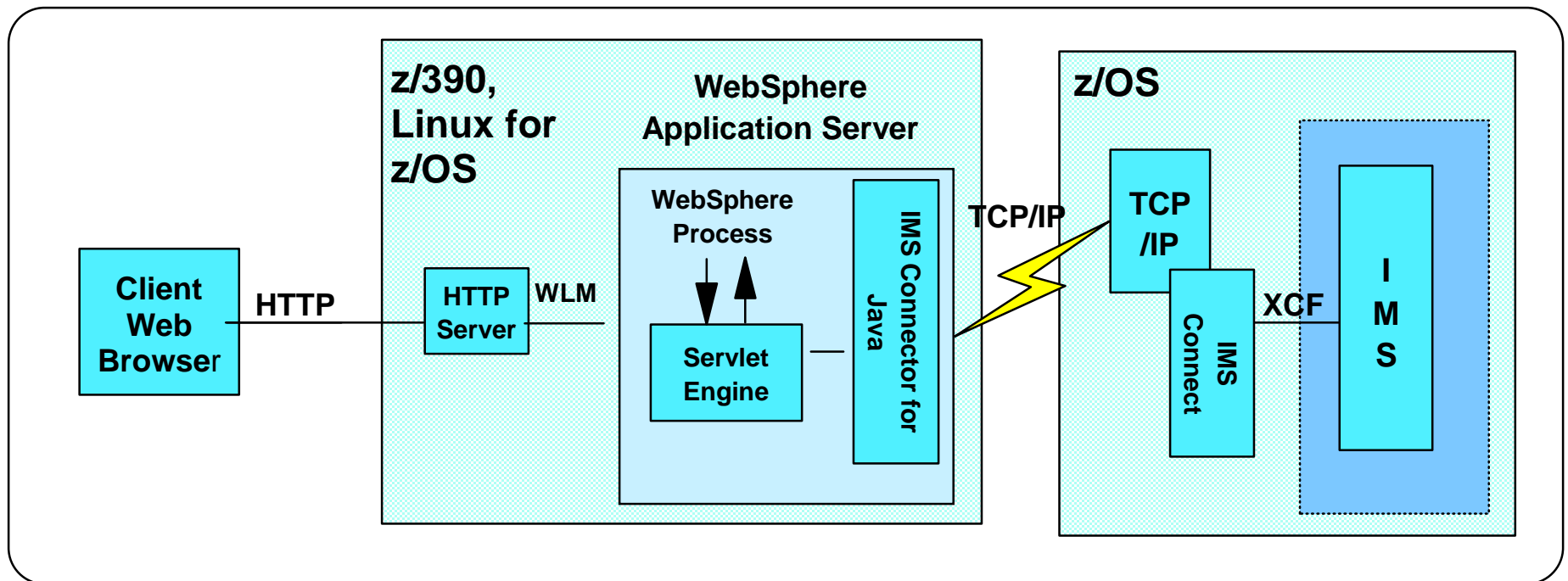
# Topology: z/OS with Local Option

- Provides non-TCP/IP communications between WebSphere Application Server and IMS Connect
- WebSphere Application Server and IMS Connect are in the same MVS image
- IMS can be on a different MVS image



# Topology: z/OS with TCP/IP

- WebSphere Application Server can be in a different MVS image than IMS Connect and IMS



# Current Functions Supported

Function	Distributed client with TCP/IP	z/OS client with Local Option	z/OS client with TCP/IP
Component-Managed Sign-on	Yes	Yes	Yes
Container-Managed Sign-on	No	Yes	No
Global Transaction	No	Yes (IMS must be in the same MVS image)	No

# Connection Management

- **Managed environment** for connections
  - ▶ WSAD IE and WAS EE both support the J2EE Connector Architecture
- Define J2C connection factories for an IMS resource adapter
  - ▶ To provide your application with connections to the EIS
  - ▶ Using WSAD IE and WAS EE tooling
- A connection factory is associated with a set of configuration properties
  - ▶ IMS resource adapter specific properties (host name, port, datastore name, etc.)
  - ▶ Standard properties used by connection manager (maximum connections, reap time, connection timeout, etc.)



# J2C Connection Factory Configuration

IMS Resource Adapter specific properties

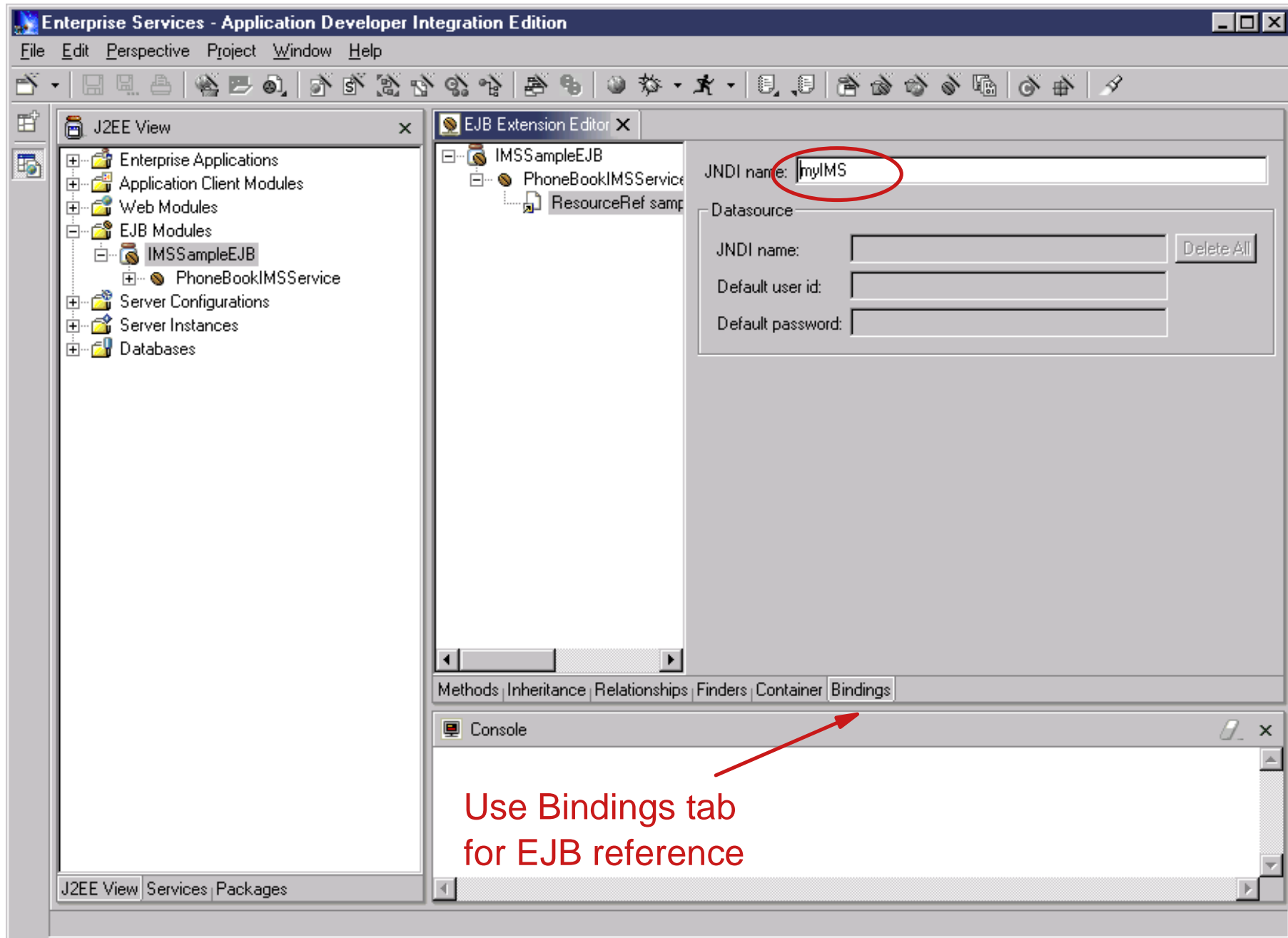
The screenshot shows the 'Server - Application Developer Integration Edition' interface. The 'J2C Resource Adapters' table lists the 'IMS' adapter. The 'J2C Connection Factories' table shows a factory named 'cf\_myIMS' with JNDI name 'myIMS'. The 'Resource Properties' table lists properties such as HostName, PortNumber, DataStoreName, Username, Password, and GroupName. The 'Edit Connection Factory' dialog box is open, showing the following properties:

Name	Value
Name	cf_myIMS
JNDI name	myIMS
Description	
Pool name	
Subpool name	
Min connections	0
Max connections	0
Connection timeout	0
Reap time	0
Unused timeout	0

The 'Server Configuration' window shows the 'myServer' instance is 'Stopped' with the message 'The server should be republished'.

Edit J2C Connection Factory for standard properties

# Using J2C Connection Factories



# Using J2C Connection Factories

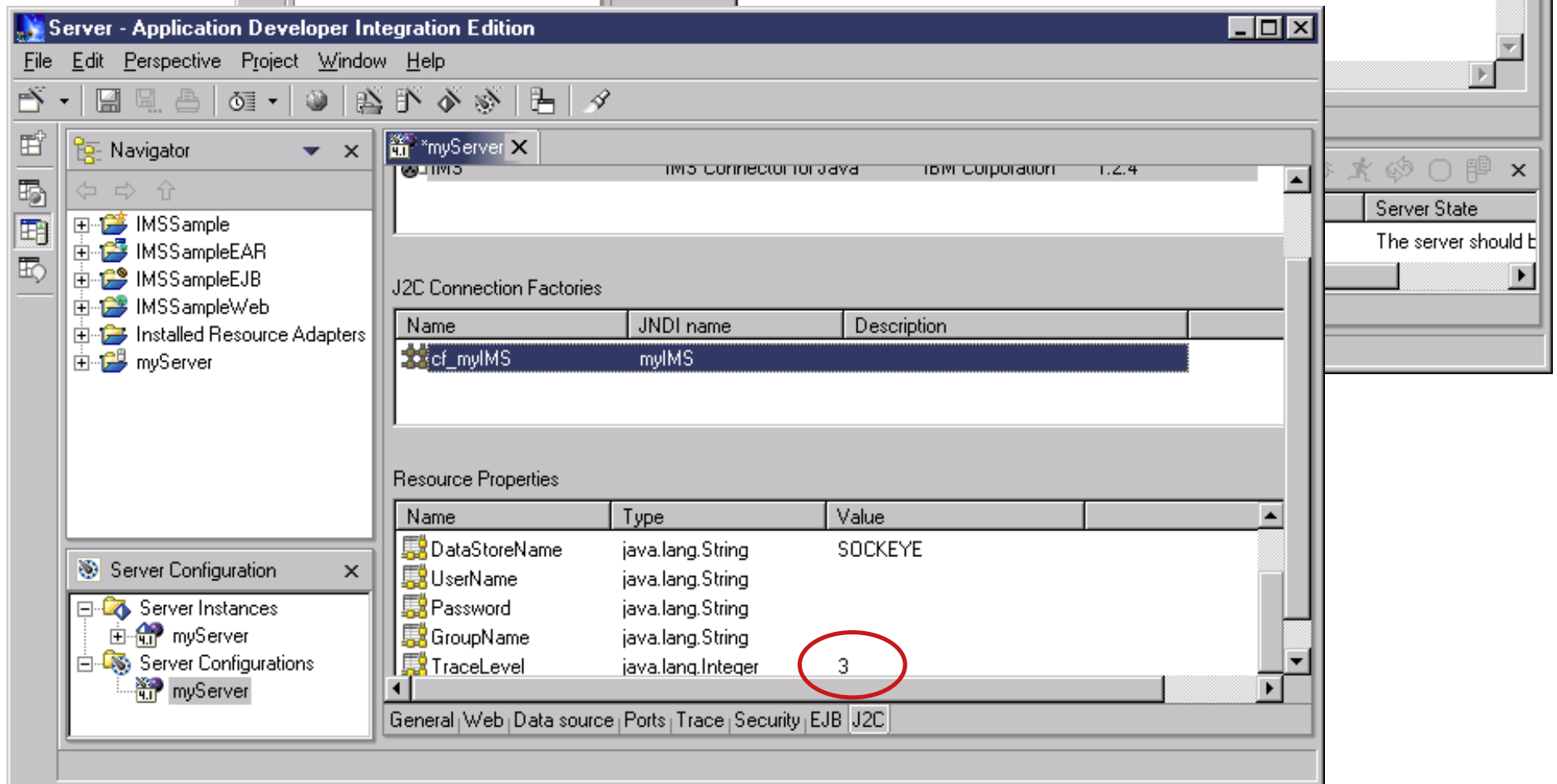
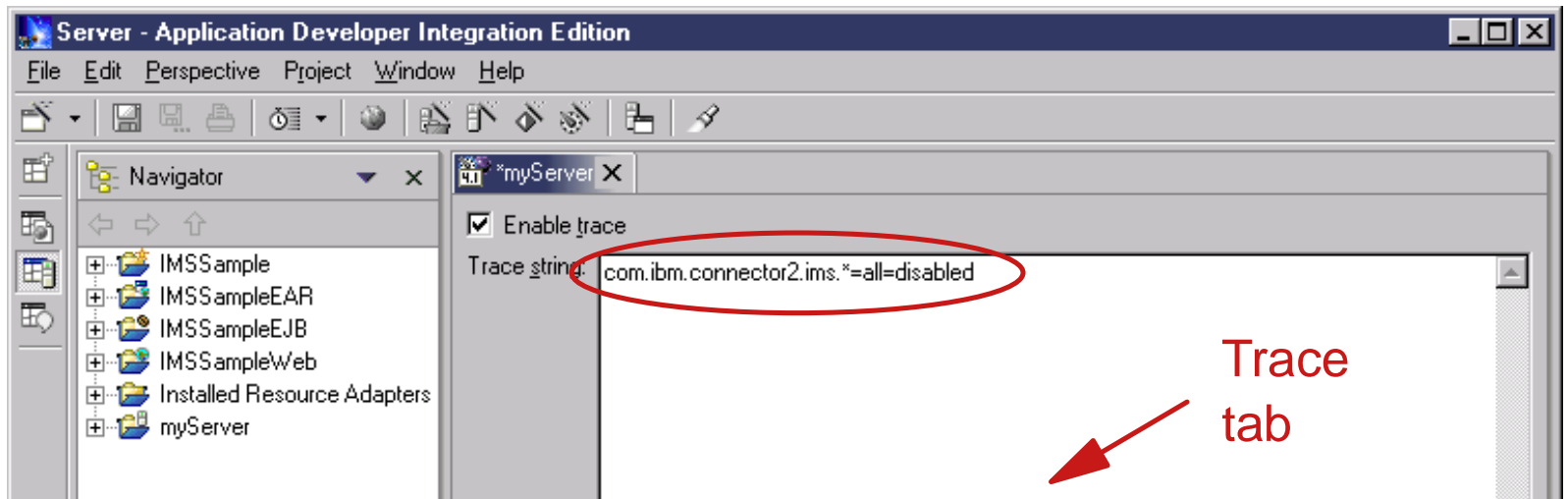
- Application components (e.g., EJBs) have resource-reference descriptors that refer to a specific connection factory
  - ▶ For example, to associate a J2C Connection Factory with the session EJB generated for an IMS service:
    - In J2EE view of Enterprise Services perspective, expand EJB Modules, right-click module, **Open with -> EJB Extension Editor**
    - Select **Bindings** tab and enter JNDI name of J2C Connection Factory; for example, "myIMS"

# Connection Management Considerations

- WAS maintains pools of connections
  - ▶ A pool for each hostName/port/dataStore combination
  - ▶ Unused connection objects are returned to the pool for re-use
    - The associated TCP/IP socket is not closed
- When a connection is closed on the IMS Connect side
  - ▶ Next attempt by an application to use the "in error" pooled connection results in an exception
  - ▶ If IMS Connect is recycled or TCP/IP failure
    - Subsequent attempts by applications to use "in error" pooled connections result in exceptions until all "in error" connections are used up

# Setting Trace for Problem analysis

- IMS Connector for Java trace:
  - ▶ Use with WSAD IE test environment or WAS
  - ▶ Use while only one iteration is executing
  - ▶ Set the trace level for the J2C Connection Factory
    - RAS\_TRACE\_INTERNAL (3)
      - Includes buffers sent to and received from IMS Connect
  - ▶ Enable tracing
    - com.ibm.connector2.ims.\*=all=enabled
      - Traces the IMS Resource Adapter
- Trace output
  - ▶ WSAD IE
    - [your\\_workspace\metadata\plugins\com.ibm.etools.websphere.tools\logs](#)
  - ▶ WAS
    - Location of trace file set using WAS tooling



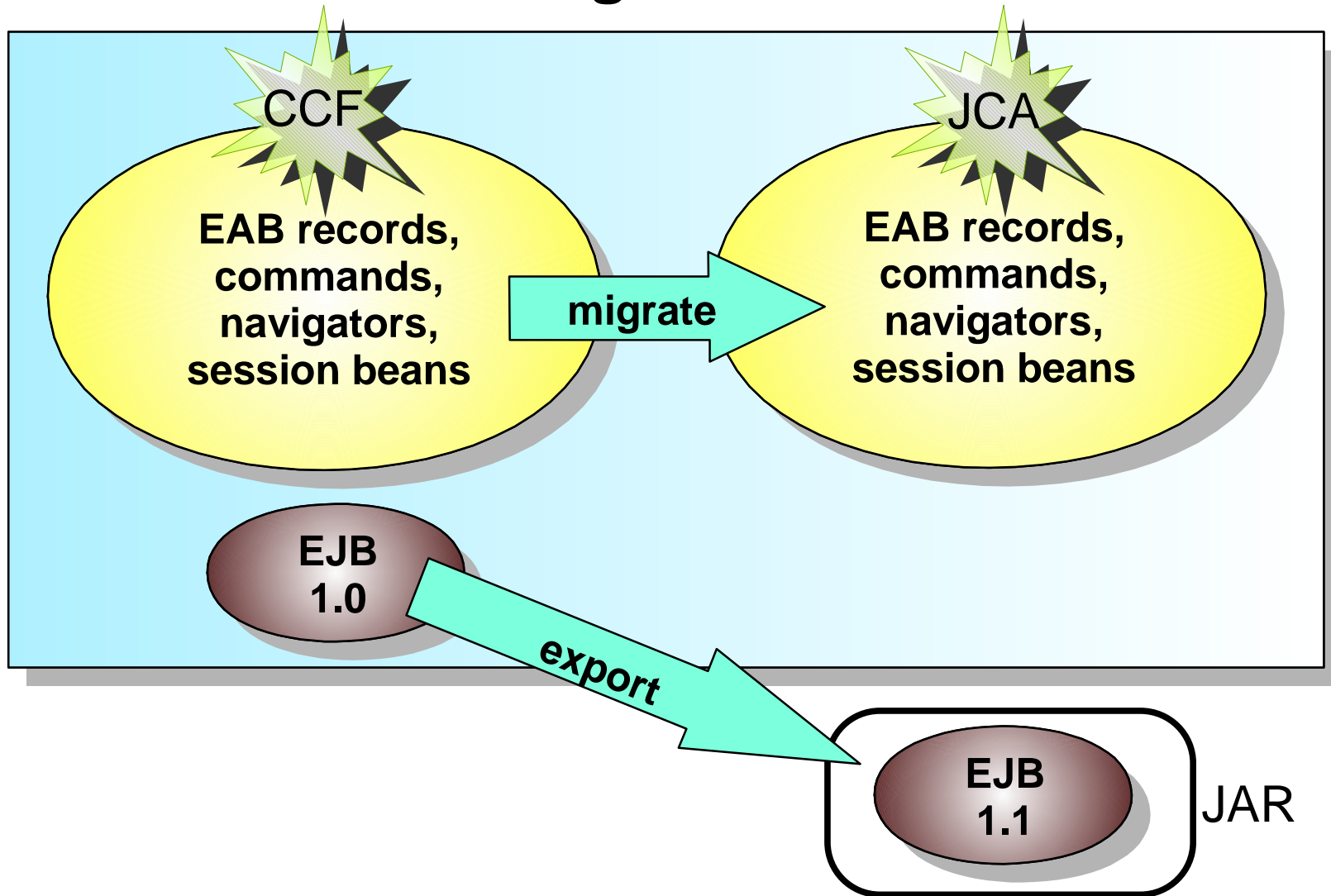


# Migration / Coexistence

- VisualAge for Java
- WebSphere Studio Integration Edition
- References

# Migration / Coexistence

## VisualAge for Java





# Migration / Coexistence

- Working with EAB applications in Integration Edition
  - ▶ Use standard **Application Developer** tools
    - Coexistence
    - Import commands, session beans and run in UTE
  - ▶ Use **Integration Edition** tools
    - Coexistence
    - Import EAB session bean and create a service
    - Create and deploy service as session bean
    - Run in UTE using SOAP proxy
  - ▶ Reengineer applications in Integration Edition
    - Create a service from imported COBOL or C file

# Migration / Coexistence

## ■ References

- ▶ WebSphere Studio Application Developer Integration Edition for Windows, V4.1 migration guide
  - *<IE\_install\_dir>\Application Developer Integration Edition\readme\migrate.html*
- ▶ WebSphere Studio Application Developer migration guide
  - *<IE\_install\_dir>\Application Developer Integration Edition\readme\wsad.readme\migrate.html*

# How to find our updates

- IMS Connector for Java **development**
  - ▶ Latest versions included in WSAD IE 4.1
- IMS Connector for Java **runtime** (aka IMS Resource Adapter, aka WebSphere Adapter for IMS)
  - ▶ For distributed platforms
    - 1.2.0/1.2.1 executables on CD with IMS Connect 1.2
      - Linux for z/OS executable downloadable from IMS Web site
    - Later versions downloadable from IMS Web site
      - [www.ibm.com/ims](http://www.ibm.com/ims)
  - ▶ For z/OS platform
    - SMPE installable APARs

# Related Web Sites



*e-business powered by IMS*

- **IMS, IMS Connect, IMS Connector for Java**
  - <http://www.ibm.com/ims>
- **WebSphere Application Server**
  - <http://www.ibm.com/software/webservers>
- **WebSphere Studio Application Developer, Integration Edition**
  - <http://www.ibm.com/software/ad/studioappdev/>