

Web Apps using Liberty Profile Technology in CICS

Ian J Mitchell,
IBM Distinguished Engineer, CICS Portfolio Architect
IBM Hursley

Thursday 15th August 2013
Session Number : 13380



Abstract

CICS TS V5.1 offers a fast and lightweight Java™ web container, providing developers with the rich features of Java Servlet and JavaServer Pages (JSP) specifications, and fast local access to your existing CICS applications and data. Built on WebSphere® Application Server Liberty profile technology, this web container runs in the CICS JVM server environment. A wide range of Java development tools can be used to develop web applications, such as WebSphere Application Server Developer Tools for Eclipse (WDT), and Rational® Developer for System z. This session will demonstrate these features and show the integration between the web container and CICS resources.

Agenda



- Java Update for CICS TS
- What is Liberty?
- Liberty in CICS
- Deeper look at Liberty Technology in CICS
- Future Direction and Summary



Java Update



- Java 7 (64-bit) JVMServer
- Equinox 3.7 as the OSGi framework.
 - Implements the OSGi R4.3 specification
- WAS Liberty Profile 8.5.0 based Web Container
- IBM CICS SDK for WebSphere Application Server Liberty profile v5.1
- Eclipse 3.6.2

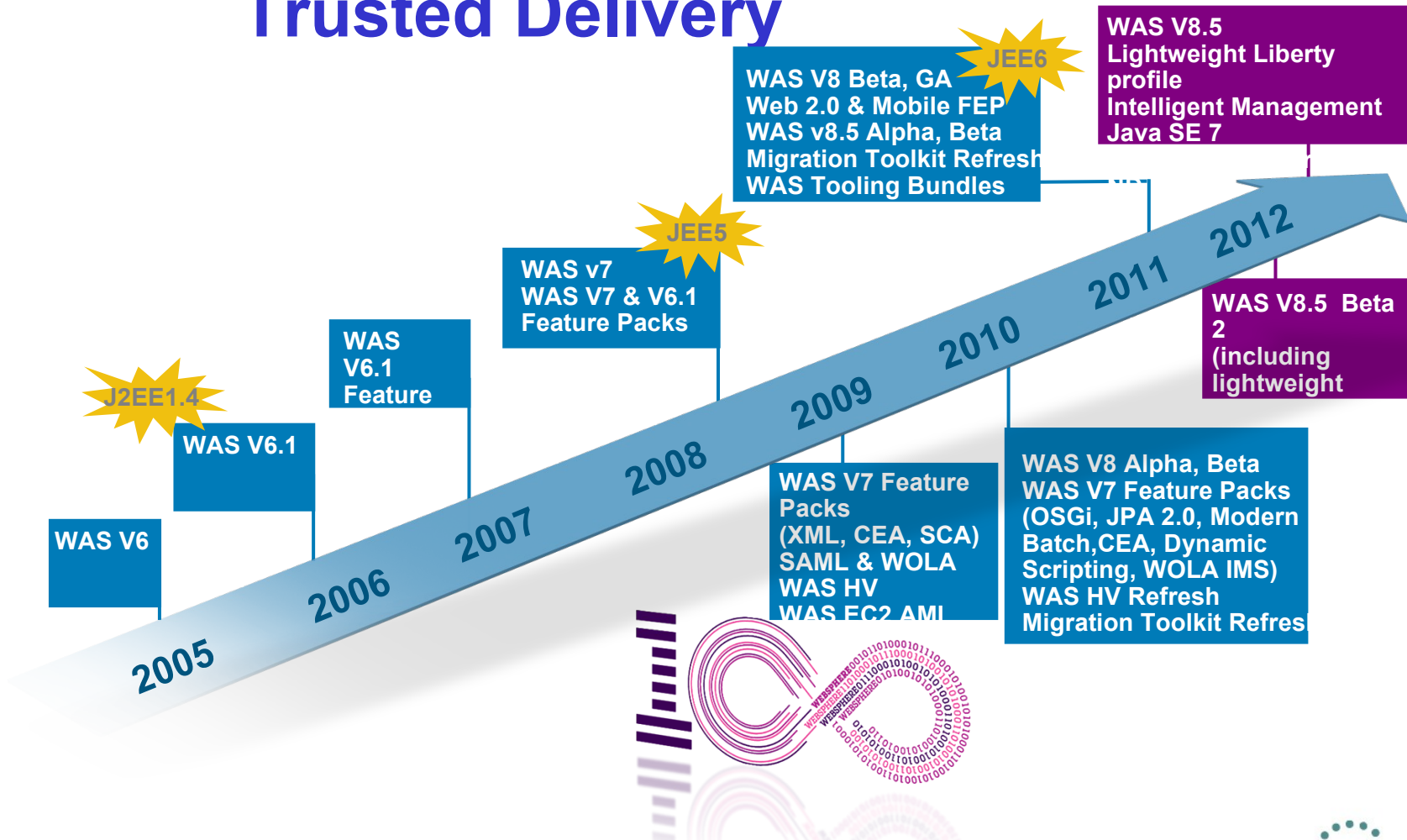


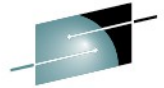
Introduction to the CICS Java Web Container based on WAS Liberty technology

What's Liberty?

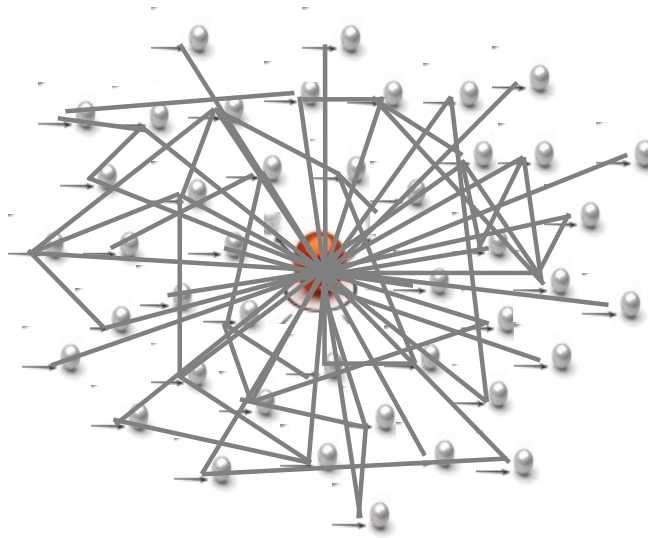


WebSphere Application Server: 15 Years of Leadership and Trusted Delivery

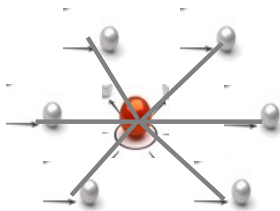




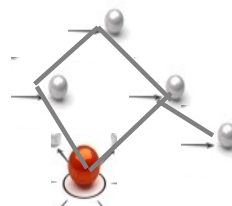
If this is tWAS...



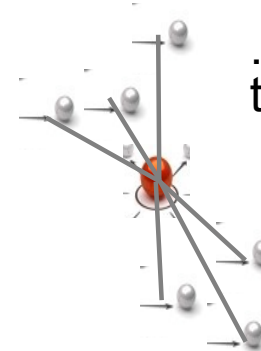
...this is Liberty (WAS)



...so is this



...or even this!



What is the 'Liberty' Profile?

A lightweight, dynamic, composable runtime

Lightweight

Server install is only about 55 MB

Extremely fast server starts – typically well under 5 seconds

Dynamic

Available features are user selected and can change at runtime

Restarts are not required for server configuration changes

Composable

Features are implemented as loosely coupled components with lazily resolved optional and mandatory dependencies

The availability of features and components determines what Liberty *can do and what's available to applications*

Configuration by Exception

- This is the entire configuration needed to run Liberty as a Web-container with Servlet support.

```
<server description="new server">  
  <featureManager>  
    <feature>servlet-3.0</feature>  
  </featureManager>  
  
  <application id="BasicWeb" location="BasicWeb.war"  
    name="BasicWeb" type="war"/>  
  
</server>
```

Lightweight Configuration

Features control what's available in the runtime.

```
<server description="tradeLiteServer">
  <featureManager>
    <feature>jsp-2.2</feature>
    <feature>jdbc-4.0</feature>
  </featureManager>
```

Singleton configurations specify properties for runtime services when there's only one instance

```
<logging consoleLogLevel="INFO" />
```

Instance configurations allow multiple instances of resources and applications to be declared

```
<application type="war"
  id="tradelite"
  name="tradelite"
  location="${shared.app.dir}/webcontainer">
```

Includes can be used to implement an extensible configuration model

```
<include location="jdbc-drivers.xml" />
<include location="${user.home}/custom.xml" optional="true" />
```

References can be used in multiple elements to point to and share a common definition

```
<dataSource id="jdbc/DerbyTradeDataSource"
  jndiName="jdbc/TradeDataSource"
  jdbcDriverRef="DerbyEmbedded">
  <properties databaseName="${shared.resource.dir}/data/tradedb" />
</dataSource>
</server>
```

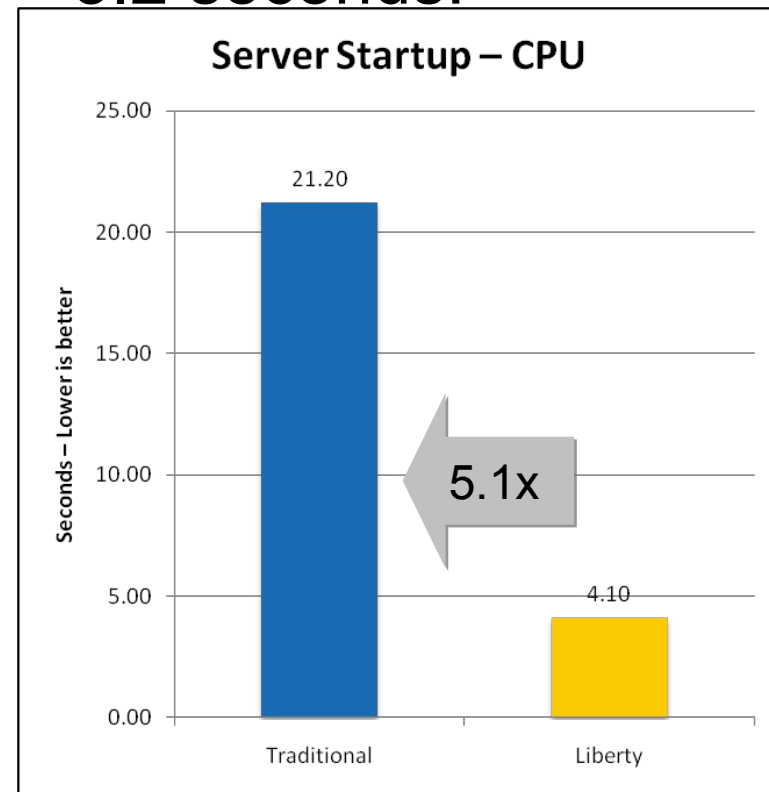
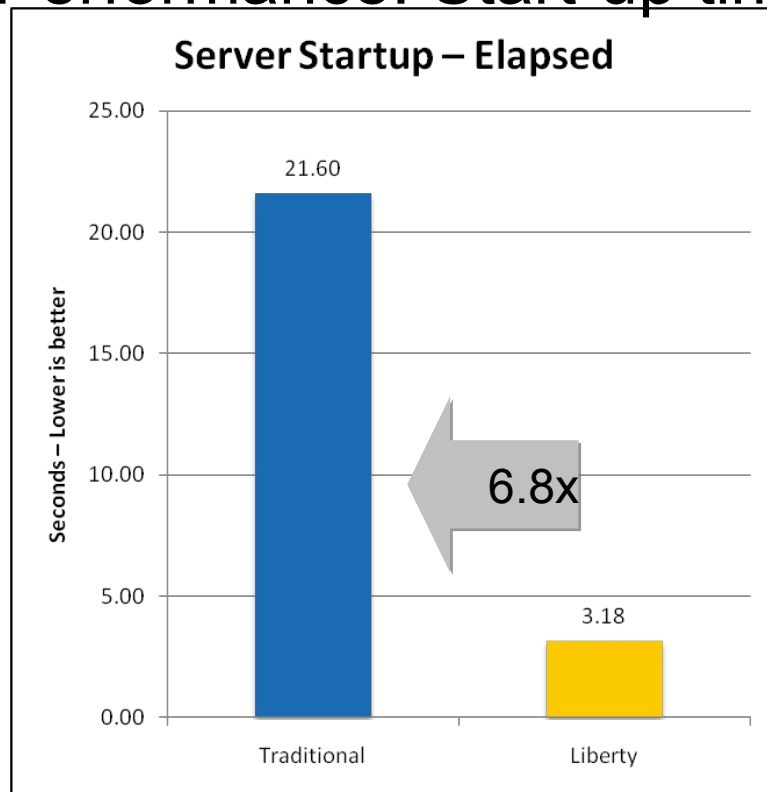
- “CICS TS V5.1 offers a fast and lightweight Java web container, providing developers with the rich features of the Java Servlet and JavaServer Pages (JSP) specifications, and fast local access to your existing CICS applications and data. Built on WebSphere Application Server Liberty technology, this web container runs in the CICS JVM server environment. A wide range of Java development tools can be used to develop web applications, such as WebSphere Application Server Developer Tools for Eclipse (WDT), and Rational Developer for System z. “



FAST. LIGHTWEIGHT. LOCAL.

Liberty on z/OS – start-up time

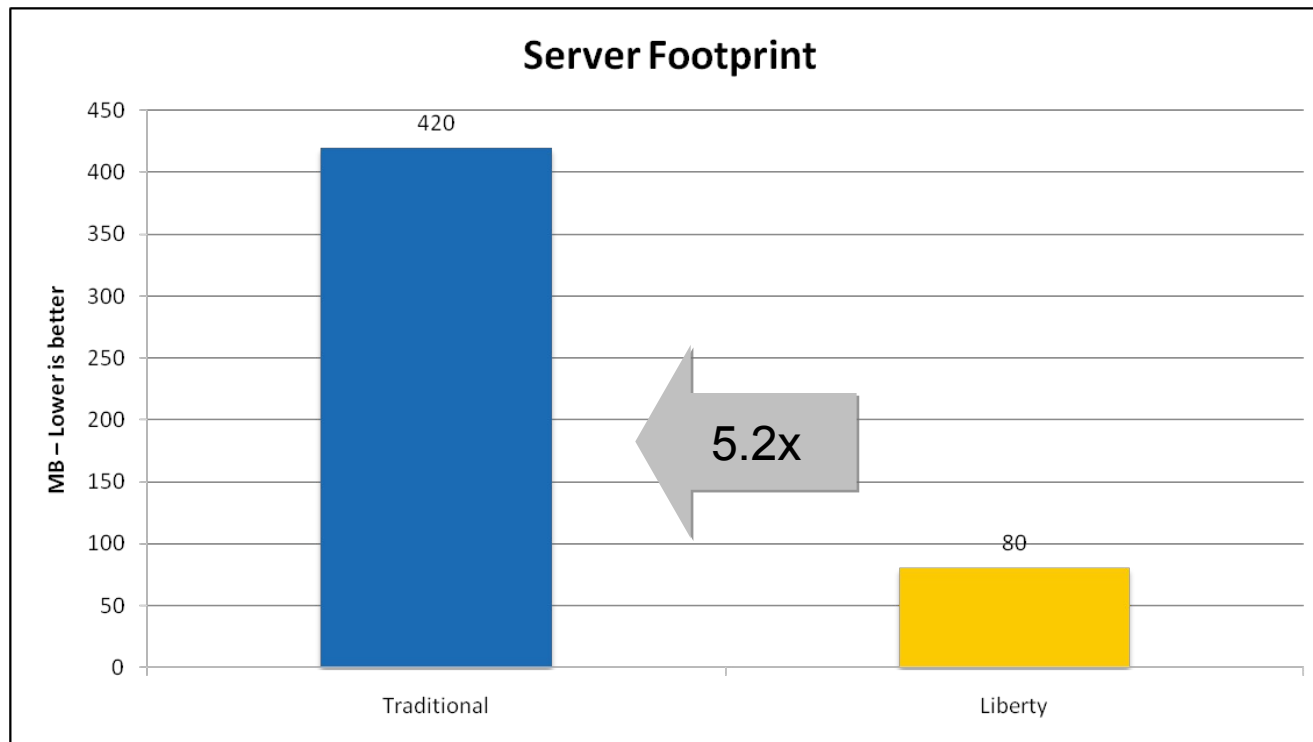
Performance: Start-up time – 3.2 seconds!



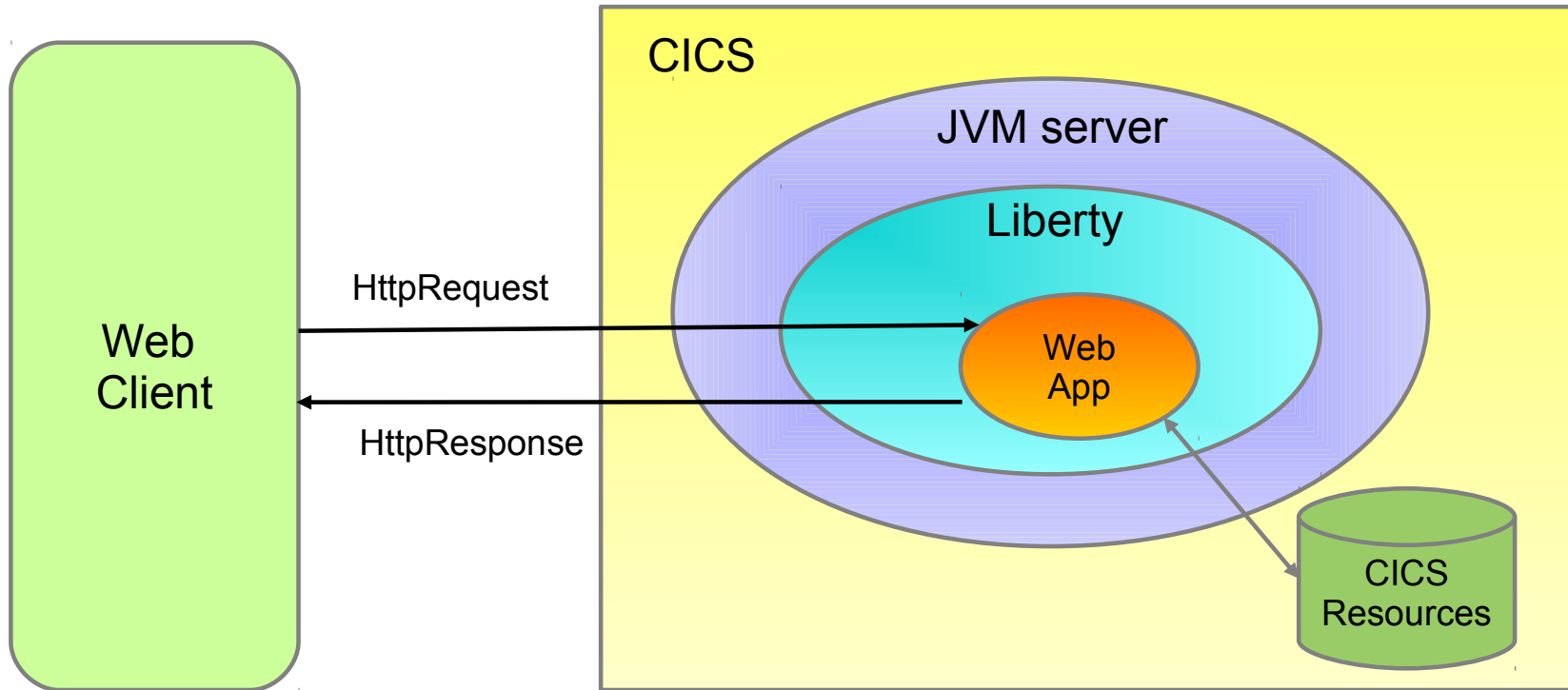
- Liberty – 64bit IBM Java 6.0.1, 64/64MB min/max heap, 60MB shared class cache, TradeLite installed
- Traditional – 64bit IBM Java 6.0.1, 1SR, 128/256MB min/max CR heap, 256/512MB min/max SR heap, 75MB CR shared class cache, 75MB SR shared class cache, no applications installed

Liberty on z/OS – memory footprint

Performance: Memory footprint – 80% reduction



- Liberty – 64bit IBM Java 6.0.1, 64/64MB min/max heap, 60MB shared class cache, TradeLite installed
- Traditional – 64bit IBM Java 6.0.1, 1SR, 128/256MB min/max CR heap, 256/512MB min/max SR heap, 75MB CR shared class cache, 75MB SR shared class cache, no applications installed



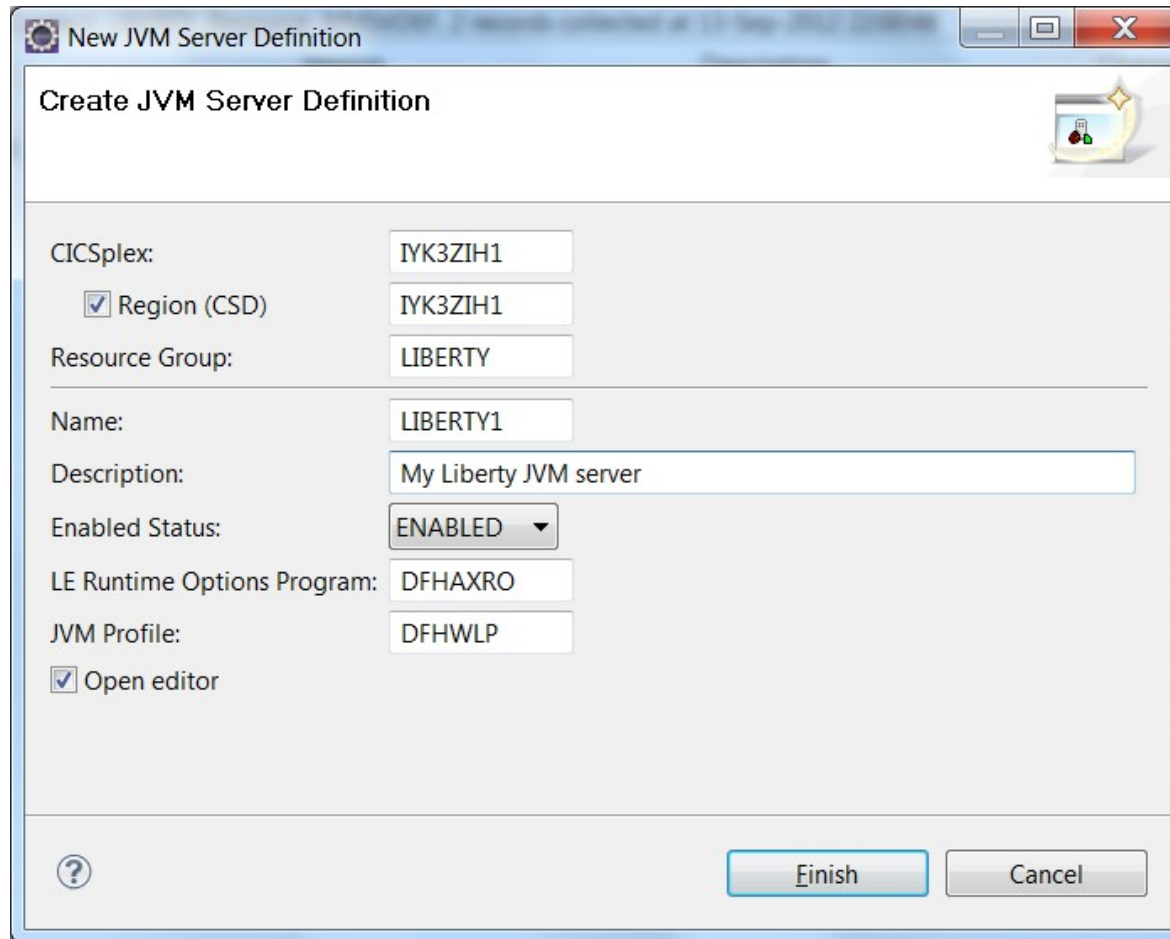
Benefits for CICS



- ✓ Provides “off the shelf” Web-server capabilities (JSPs and Servlets)
- ✓ Potential to re-use even more WebSphere technology in CICS.
- ✓ JSP and Web servlets have direct, local, access to CICS data and resources.
- ✓ Servlets can take advantage of existing CICS OSGi applications to provide a Dynamic Web front end.

Nought to Web-App

Create a JVM server resource in Explorer, CEDA, or CPSM.



Create JVM Server Definition

CICSplex: IYK3ZIH1

Region (CSD) IYK3ZIH1

Resource Group: LIBERTY

Name: LIBERTY1

Description: My Liberty JVM server

Enabled Status: ENABLED

LE Runtime Options Program: DFHAXRO

JVM Profile: DFHWLP

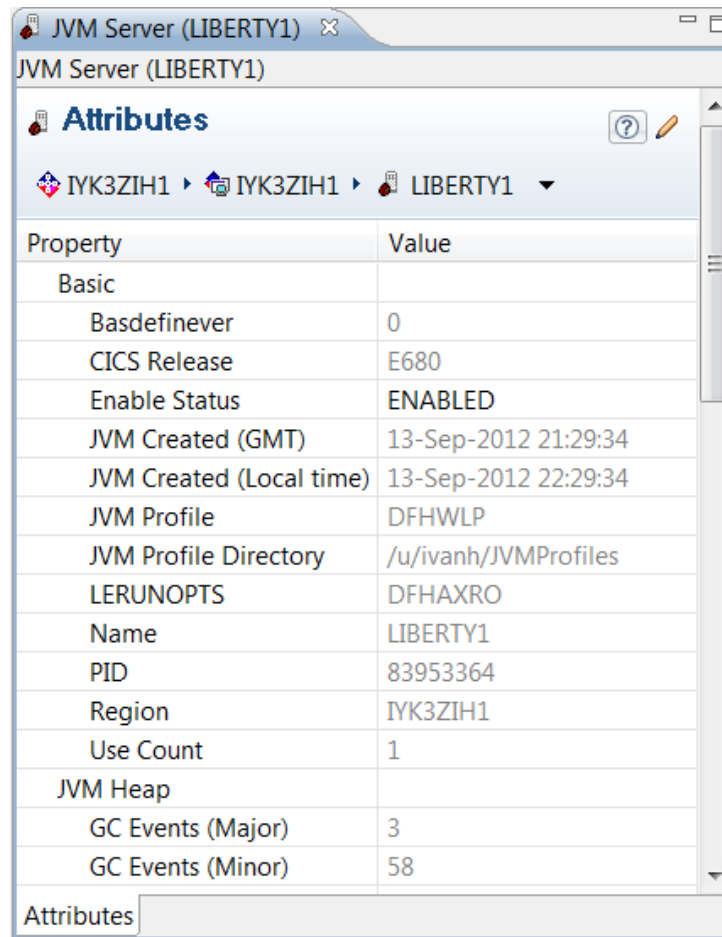
Open editor

? Finish Cancel

Configure the JVMProfile

- Copy the sample DFHWLP
- Check JAVA_HOME is correct.
- Uncomment the WLP_SERVER_HTTP_PORT and choose a unique port number.
- Point your JVM server definition at the new JVMProfile

Enable the JVM server



JVM Server (LIBERTY1)

Attributes

IYK3ZIH1 ▸ IYK3ZIH1 ▸ LIBERTY1 ▾

Property	Value
Basic	
Basdefinever	0
CICS Release	E680
Enable Status	ENABLED
JVM Created (GMT)	13-Sep-2012 21:29:34
JVM Created (Local time)	13-Sep-2012 22:29:34
JVM Profile	DFHWLP
JVM Profile Directory	/u/ivanh/JVMProfiles
LERUNOPTS	DFHAXRO
Name	LIBERTY1
PID	83953364
Region	IYK3ZIH1
Use Count	1
JVM Heap	
GC Events (Major)	3
GC Events (Minor)	58

Attributes

Liberty is running! (check the logs).

Server defaultServer created.

Launching defaultServer (wlp-1.0.0.20120428-1251/websphere-kernel_1.0.0) on
IBM J9 VM, version pmz6470sr1-20120302_01 (SR1) (en_US)

[AUDIT] CWWKE0001I: The server defaultServer has been launched.

[AUDIT] CWWKG0028A: Processing included configuration resource:
file:/u/ivanh/IYK3ZIH1/LIBERTY1/wlp/usr/servers/defaultServer/installedApp
s.xml

[AUDIT] CWWKG0028A: Processing included configuration resource:
file:/u/ivanh/IYK3ZIH1/LIBERTY1/wlp/usr/servers/defaultServer/cicsSecurity
.xml

[AUDIT] CWWKZ0058I: Monitoring dropins for applications.

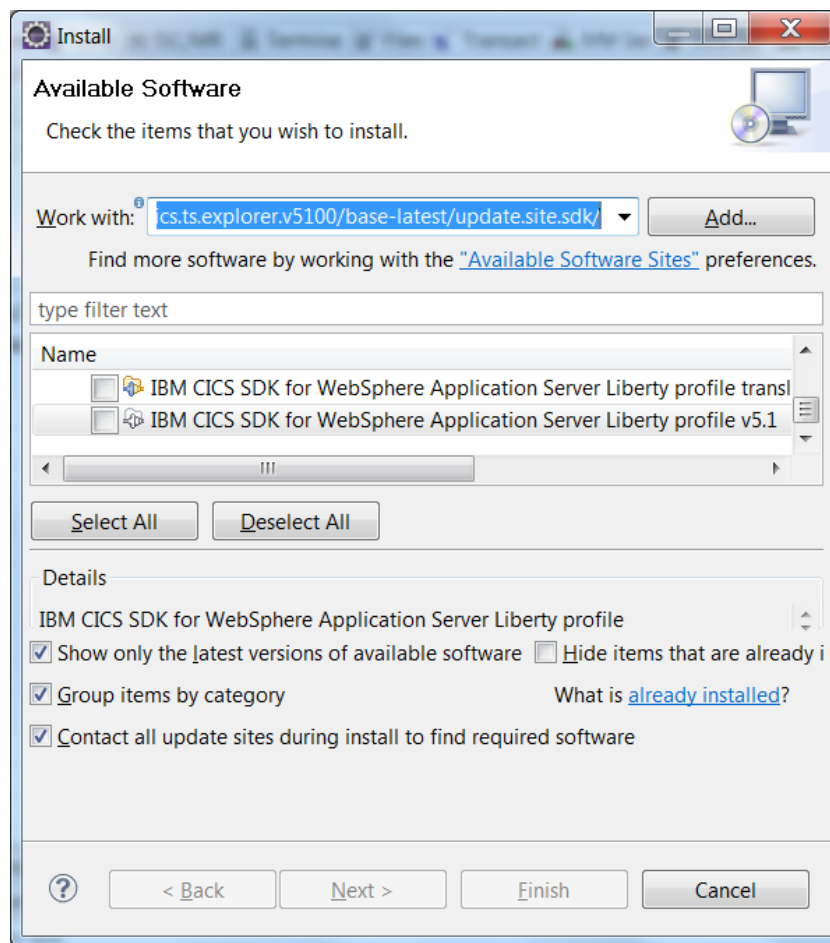
[AUDIT] CWWKF0011I: **The server defaultServer is ready to run a smarter
planet.**

Install Eclipse 3.6.2 preferably JEE version, but Classic will suffice.

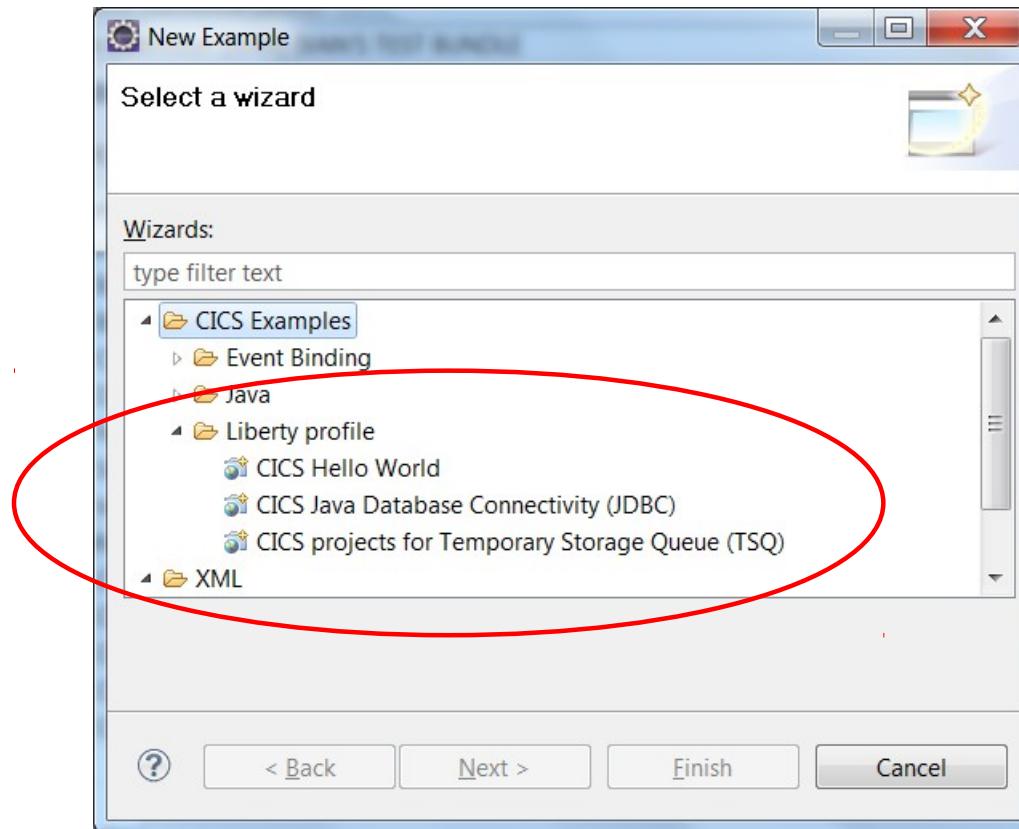
<http://www.eclipse.org/downloads/packages/release/helios/sr2>



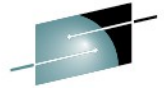
Install IBM CICS SDK for WebSphere Application Server Liberty profile v5.1



Create a Dynamic Web Project, or choose one of the Examples



JSP/Servlets plus JCICS/JDBC/Cobol



File Edit Source Refactor Navigate Search Project Run Window Help

Crash Reporter Java EE

Team My W Feed

HelloWorld.java TsqInfo.java index.jsp

All Project and Team Areas (60 of 91 areas selected)

- Repository Connections
- CICS Development and Service Environment [jazz1]
- CICS RFE Project [jazz104.hursley.ibm.com]
- CICS TS for zOS [jazz104.hursley.ibm.com]
- COBOL Container for WAS [jazz104.hursley.ibm.com]
- Eclipse Tooling Development [jazz104.hursley.ibm.com]
- SCM CICS TS for zOS [jazz104.hursley.ibm.com]
- Debug
- Favorites
- Feeds
- My Repository Workspaces
- My Team Areas
- Work Item History

```
22  */
23  public TsqInfo() {
24      super();
25      // TODO Auto-generated constructor stub
26  }
27
28  /**
29   * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
30
31  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
32
33      // obtain the input values from the request
34      String tsq = request.getParameter("tsq");
35
36      System.out.println("Tsq INFO is: " + tsq);
37
38
39      TSQ tsqQ = new TSQ();
40      tsqQ.setName(tsq);
41
42      int length = 0;
43      try{
44          length = tsqQ.readItem(1, new ItemHolder());
45      } catch(Exception e){
46          e.printStackTrace();
47      }
48
49      String name = "<name>" + tsqQ.getName() + "</name>";
50      String type = "<type>" + tsqQ.getType().toString() + "</type>";
51      //String sysID = sysID + tsqQ.getSysId() + tsqQ.getSysId();
52      String lenStr = "<length>" + length + "</length>";
53
54      //System.out.println("Tsq SYSID is: " + tsqQ.getSysId());
55
56
57      response.getOutputStream().write("<info>".getBytes());
58
59      response.getOutputStream().write(name.getBytes());
60      response.getOutputStream().write(type.getBytes());
61      //response.getOutputStream().write(sysID.getBytes());
62      response.getOutputStream().write(lenStr.getBytes());
63
```

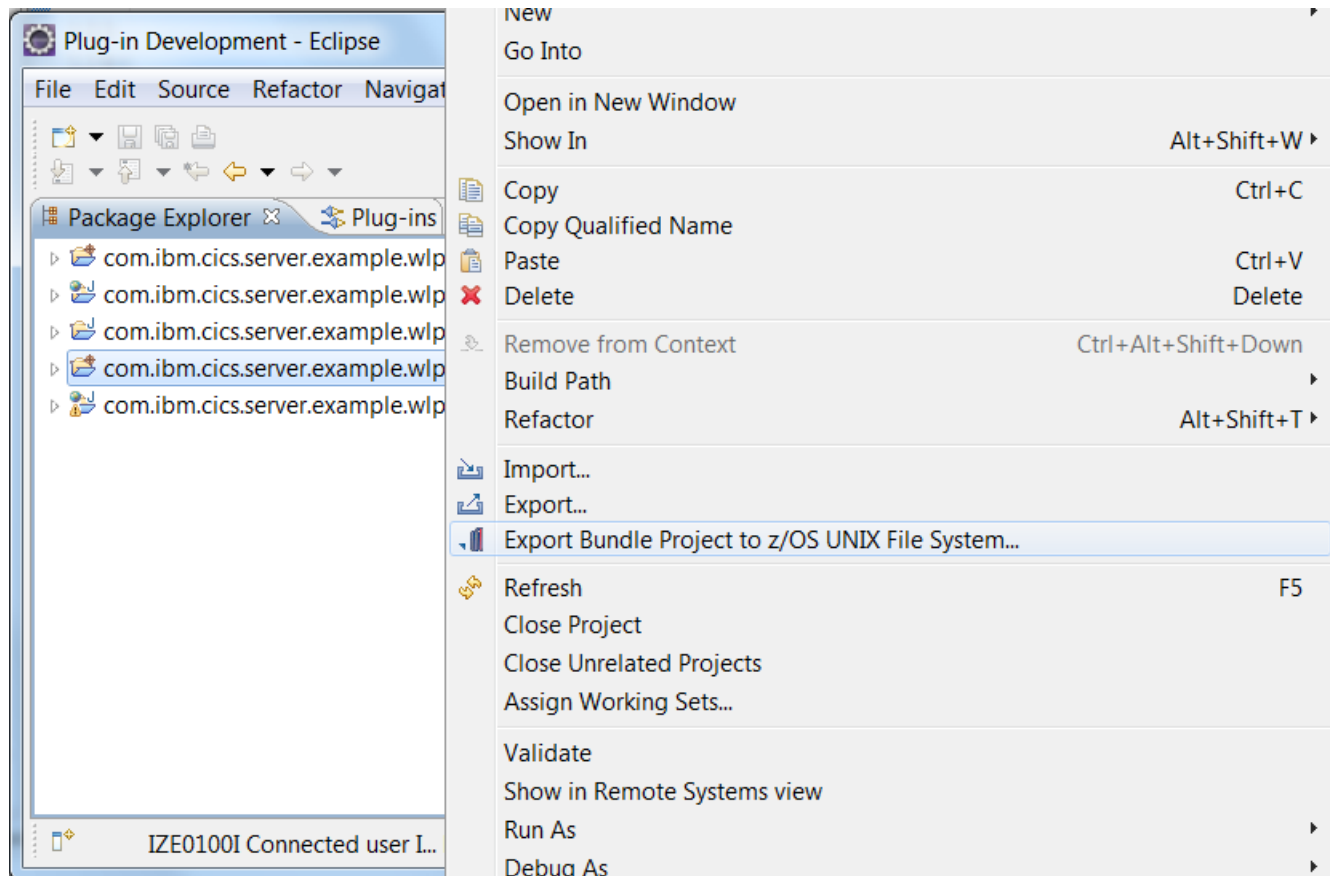
Project Explorer Remote System

- WebContent
 - images
 - META-INF
 - WEB-INF
 - index.jsp
 - response.jsp
 - splat.jsp
- HostConnectProjectFiles
 - org.example.mysql
 - org.example.mysql.ui
 - RTC-LP
 - Tivoli CDM

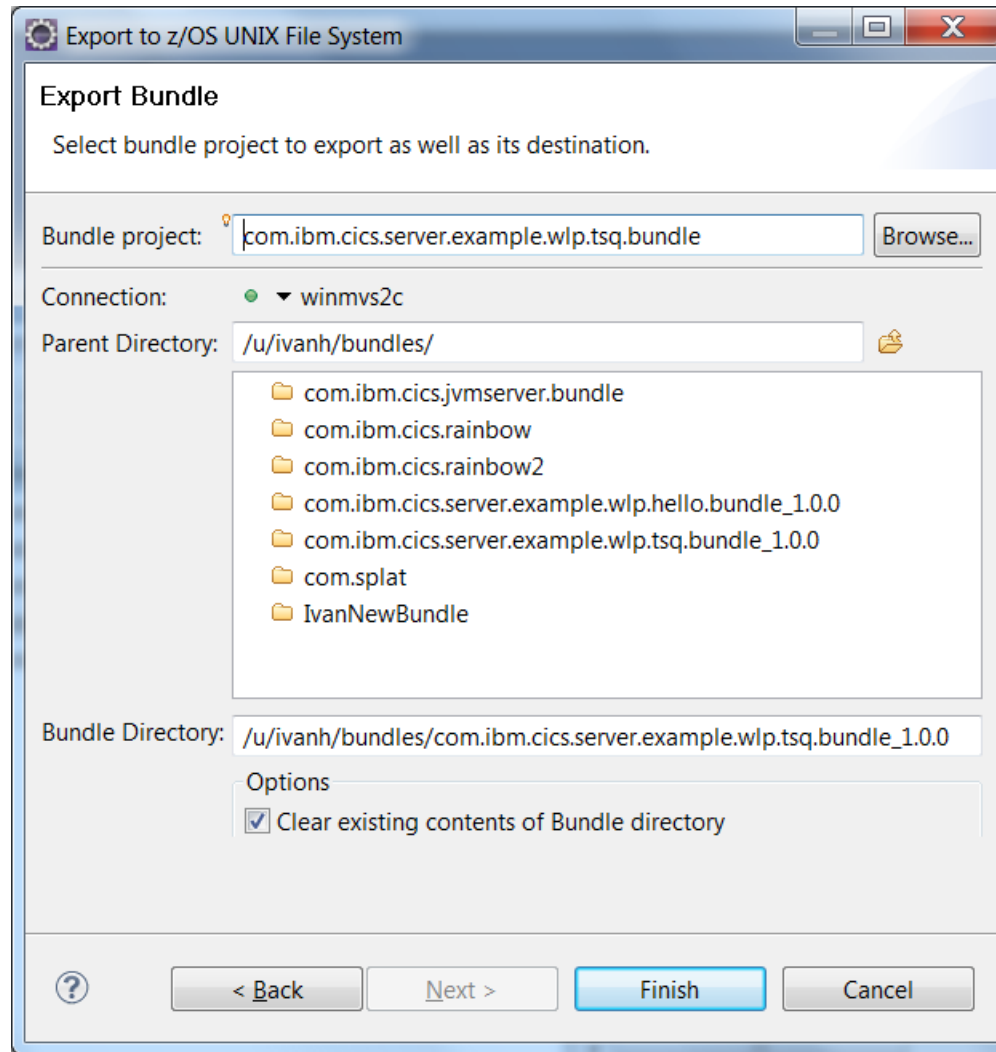
Chang Tag Cl Work I Probl Builds Repo Antz Histor Prope Remo Progr Pendi Navig Cons Team Search

Search for ID or Text CICS TS for zOS Writable Smart Insert 53 : 13 <No Current Work>

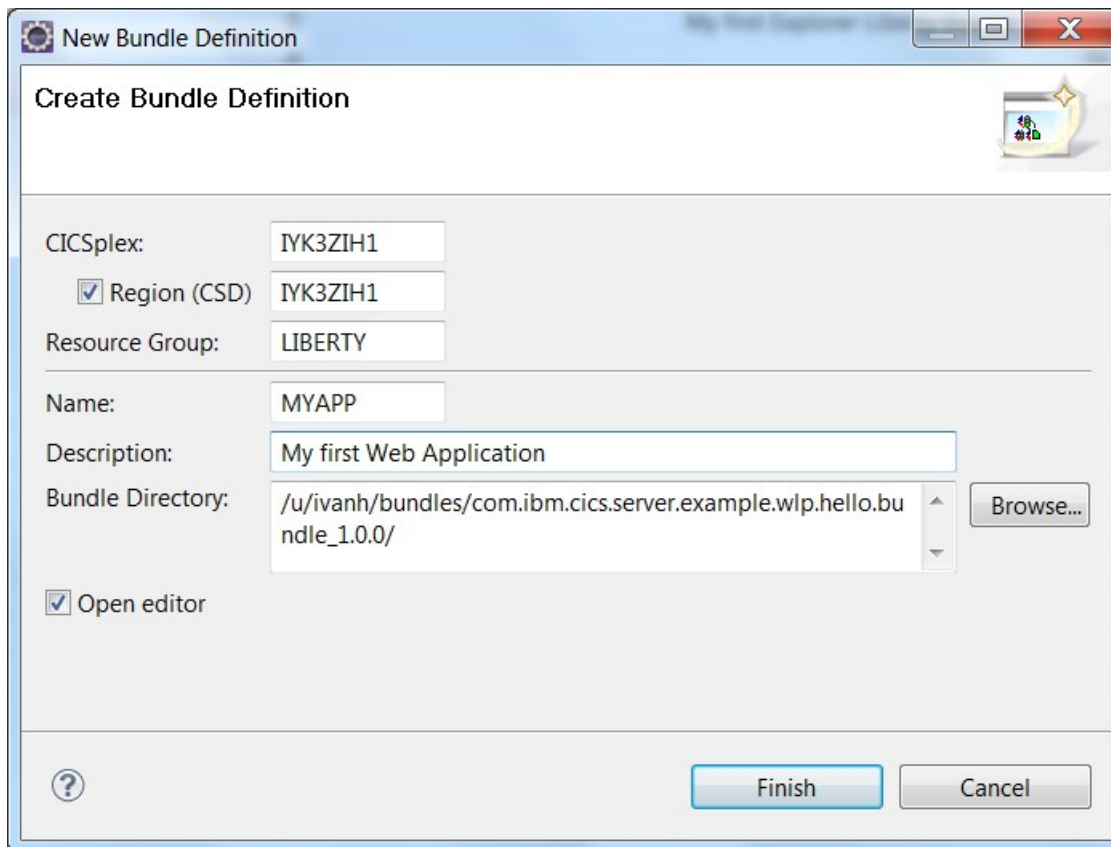
Export the CICS bundle project



Pick a zFS location for the CICS bundle project



Create a CICS bundle definition to control the life-cycle of the Application



New Bundle Definition

Create Bundle Definition

CICSplex: IYK3ZIH1

Region (CSD) IYK3ZIH1

Resource Group: LIBERTY

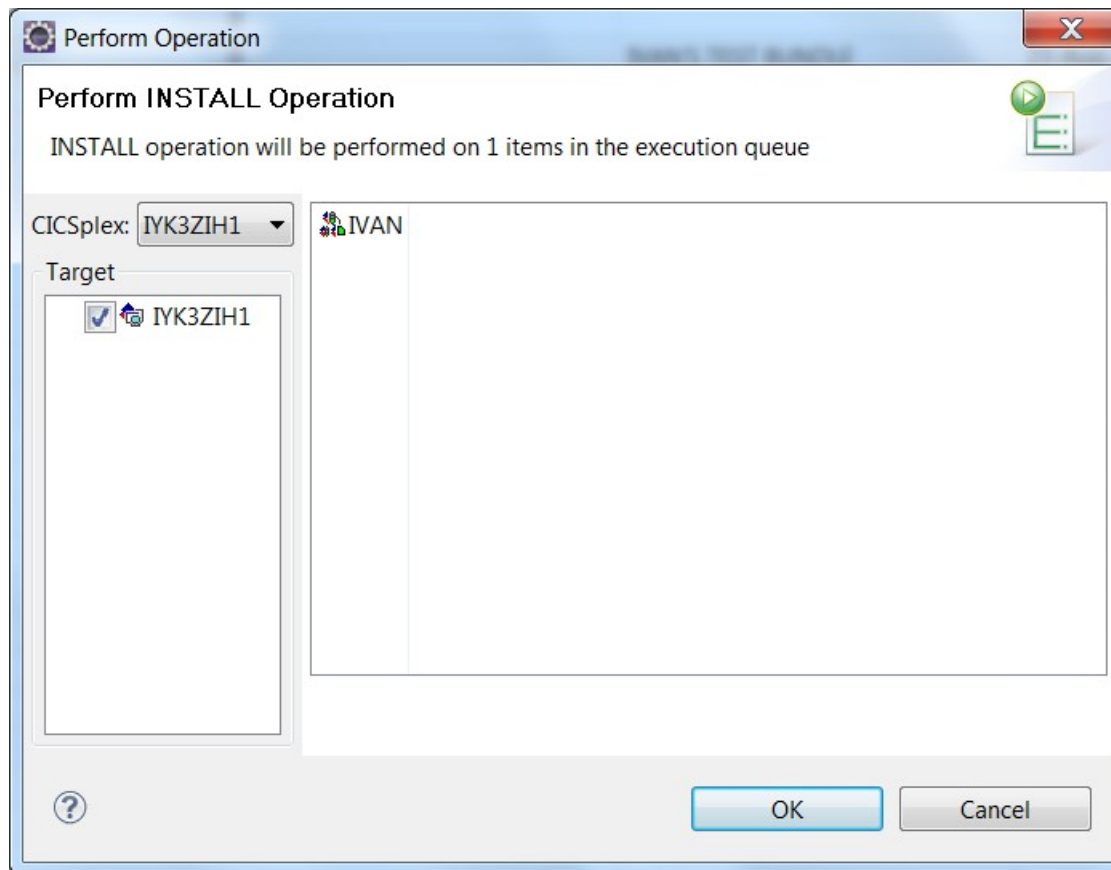
Name: MYAPP

Description: My first Web Application

Bundle Directory: /u/ivanh/bundles/com.ibm.cics.server.example.wlp.hello.bundle_1.0.0/

Open editor

Install the CICS bundle definition



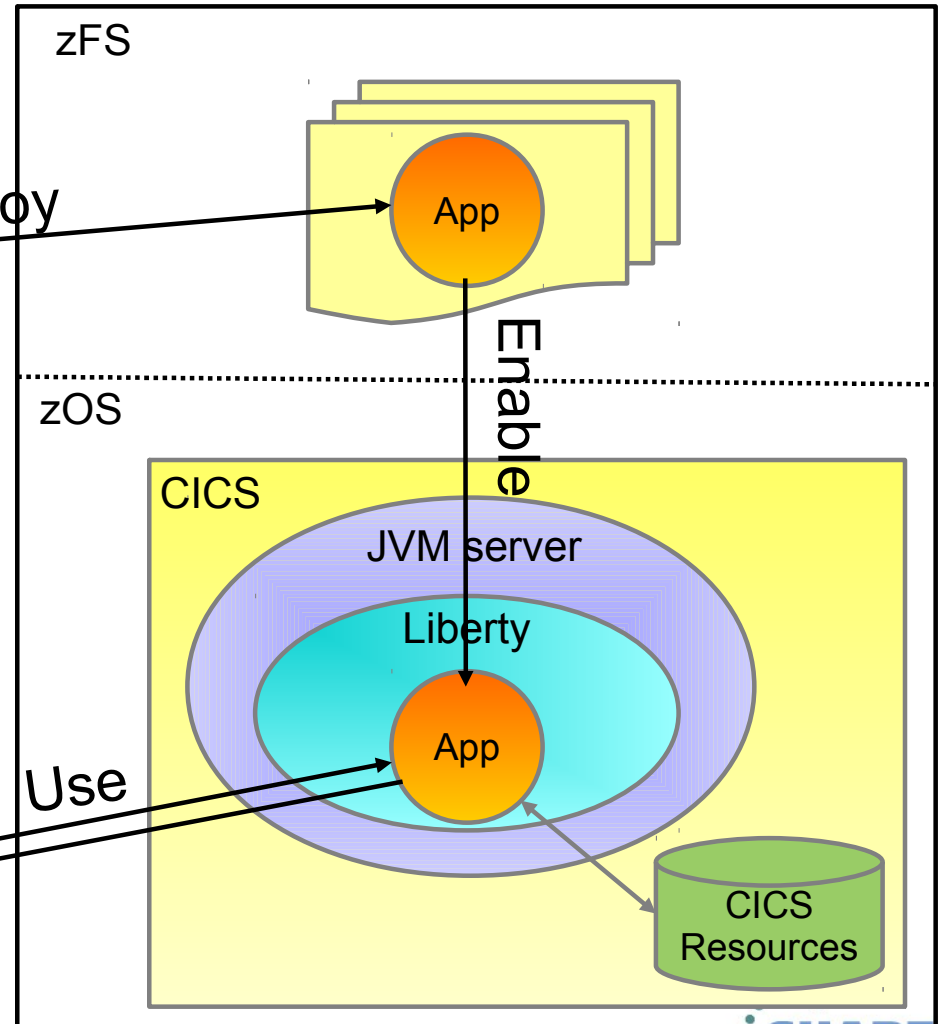
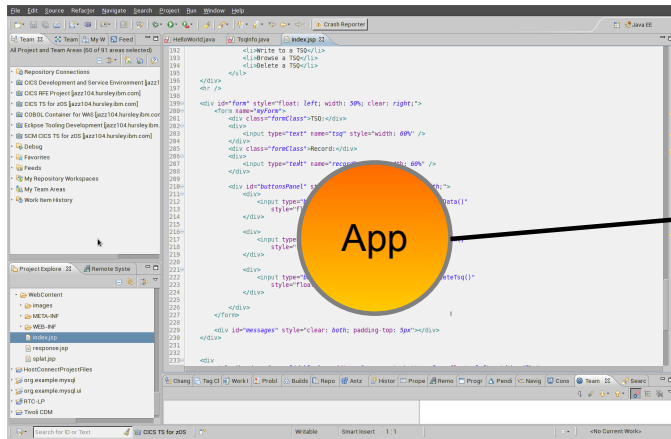
Run the application!

[AUDIT] CWWKT0016I: Web application available (default_host):
<http://winmvs2c.hursley.ibm.com:27245/com.ibm.cics.server.example.wlp.tsq.web/>



Putting it all together

Eclipse with Liberty Tools



The Technology

Principles

As little customization as we can get away with.

Do things the Liberty way first, and if appropriate, only the Liberty way.

Ensure Server.xml can be configured dynamically by the user.

Support Liberty monitored drop-ins directory for applications.

Provide CICS enhancements only where absolutely necessary (Security, Tasks, JDBC, MQ)

Provide End-to-end Development and Deployment experience to enable non-mainframe professionals to develop for CICS.

Fully compatible with existing CICS OSGi Java applications running within the same JVM server.

Specifications and Standards



- Java 7 (64-bit)

Equinox 3.7 as the OSGi framework.

Implements the OSGi R4.3 specification

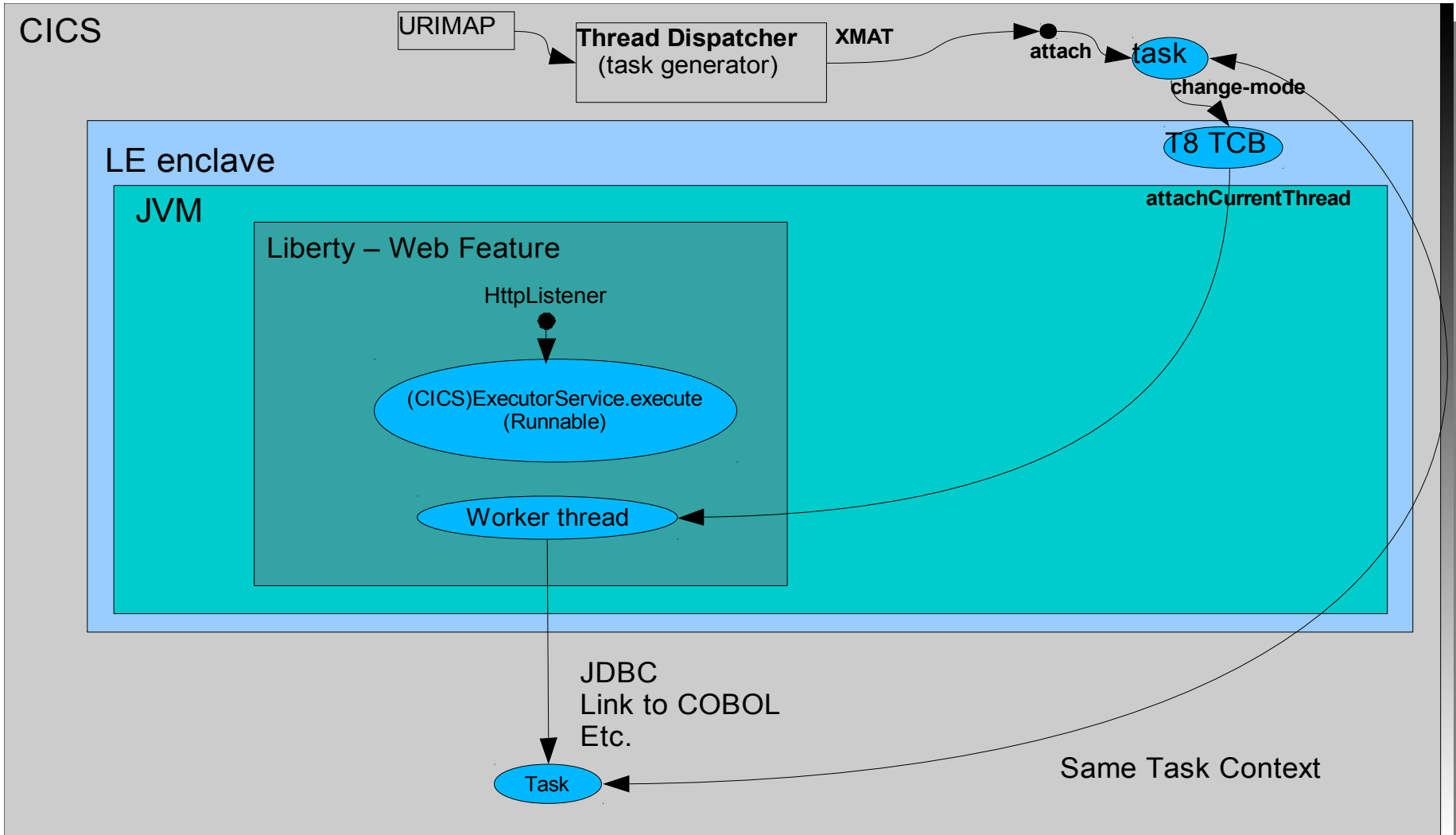
WAS Liberty Profile 8.5.0

IBM CICS SDK for WebSphere Application Server Liberty profile v5.1

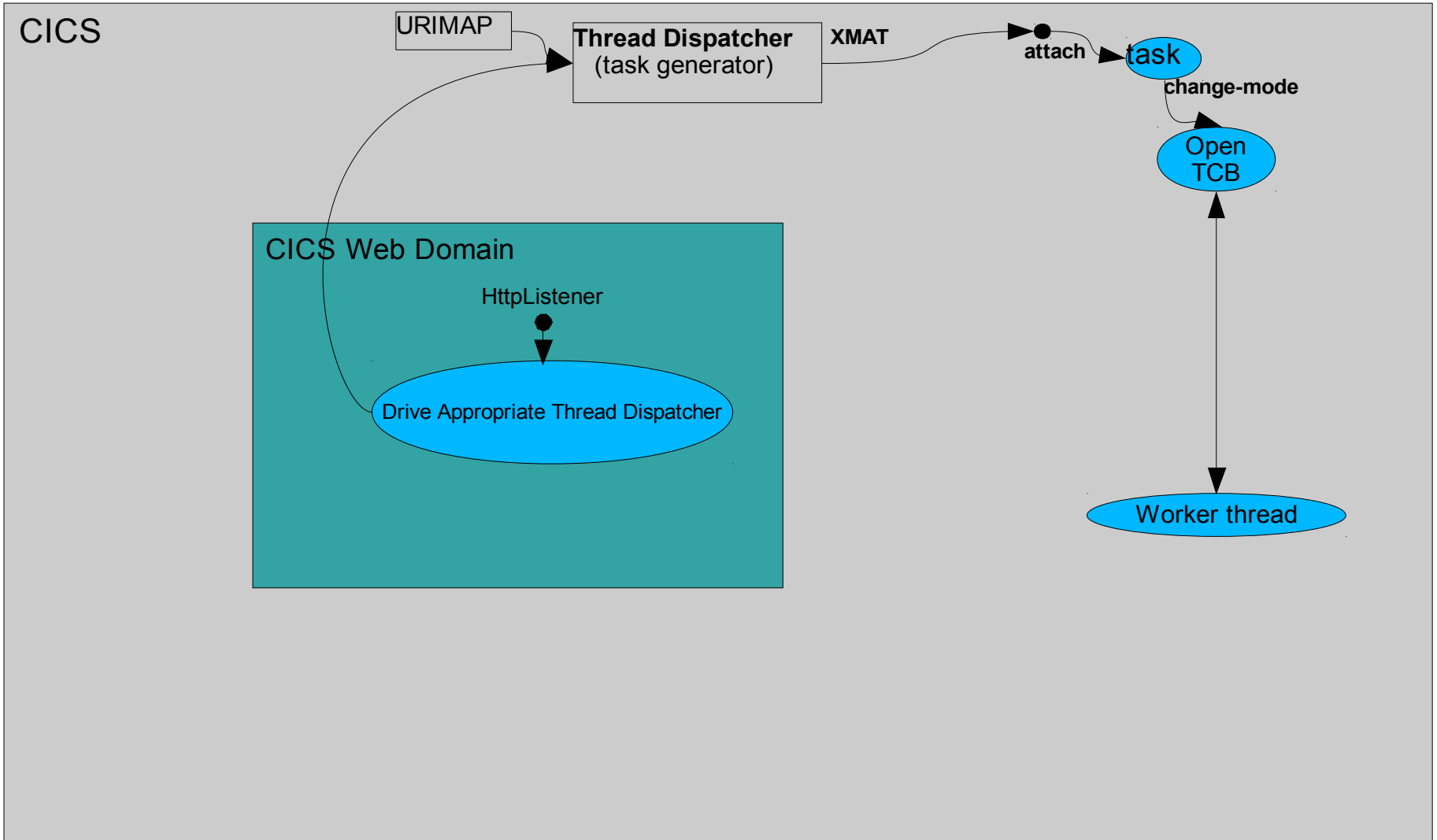
Eclipse 3.6.2



Hybrid Threads



'Standard' CICS Listener Pattern

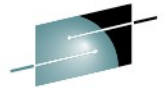


JCICS – ExecutorService

- Thread.start equivalent (from Java concurrency package)
- A standard Java pattern for dispatching runnable code to threads.
- CICS provides “CICSExecutorService” - to create CICS capable threads.
- CICSExecutorService registered with OSGi registry, can be obtained and used by 'vendor' products and applications.
- A convenience method provided called “CICSExecutorService.runAsCICS()”
- Liberty requests an ExecutorService from the OSGi service registry. When running in CICS JVM server, it is given the CICSExecutorService which produces JCICS enabled threads for Liberty to run servlets on.

Benefits of Hybrid Threads

- Each 'Invocation' (think Servlet Request) on a Hybrid Thread is also a CICS Transaction (Has a Trandid, Task Context etc).
- This gives you
 - A single common Transaction (UOW) and CICS Managed JDBC
 - Which can cross between Java and Cobol
 - Full JCICS API Access
 - In particular, LINK and access to VSAM
 - WLM (CICS WLM, Performance Classes etc).
 - Monitoring / Statistics
 - CICS Transaction Tracking / Association Data



CICS Security with Liberty



Servlets run under default transaction CJSAA with CICS

SEC=YES turns Security ON.

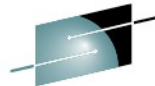
Basic-auth only (http or https) – Client cert not yet supported.

Client Application: Web.xml needs `<security_constraint>` to run with Security

Liberty: Server.xml will be updated by CICS automatically

- `<application-bnd>`

Role based Security not supported.

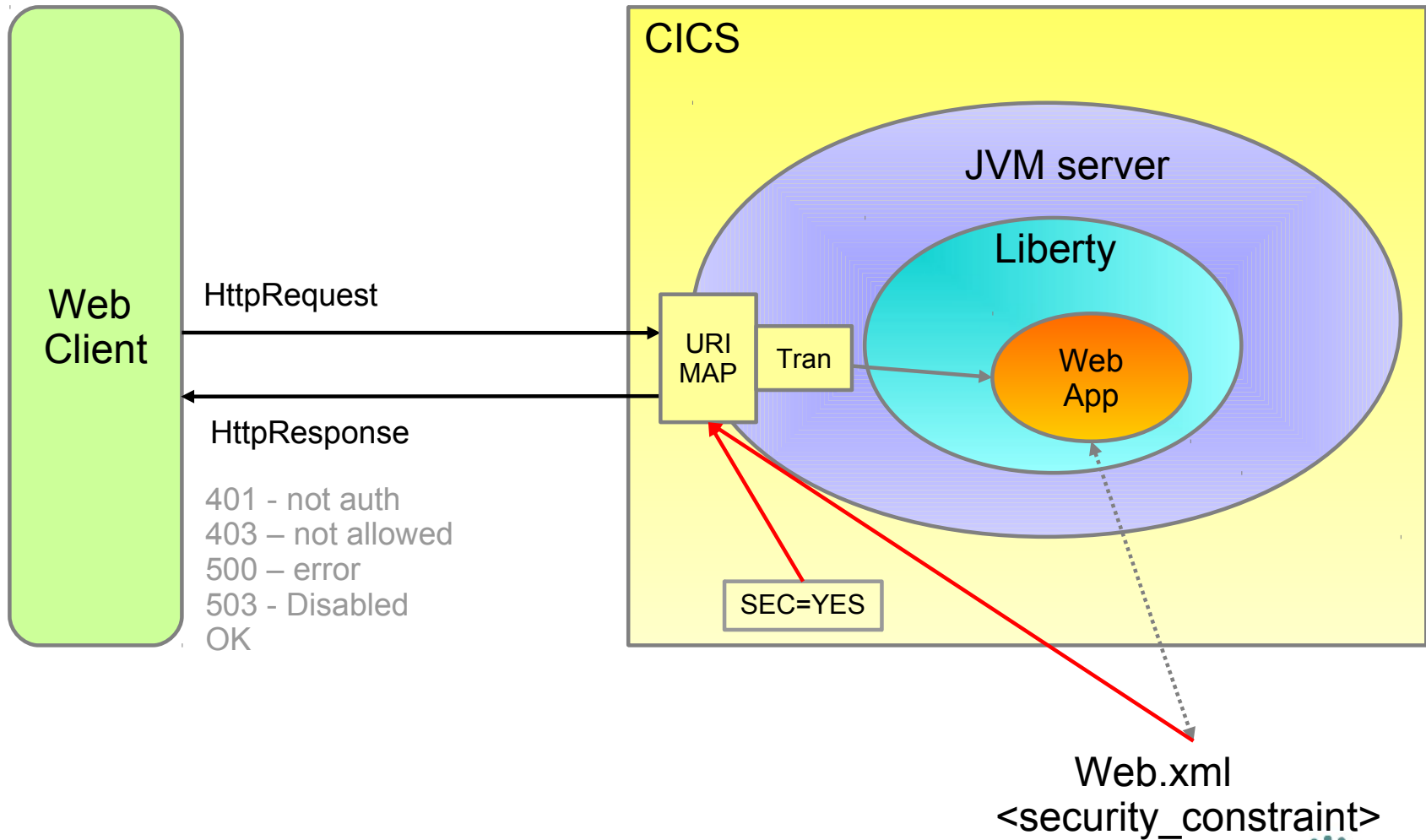


URIMAP enhancements for Liberty

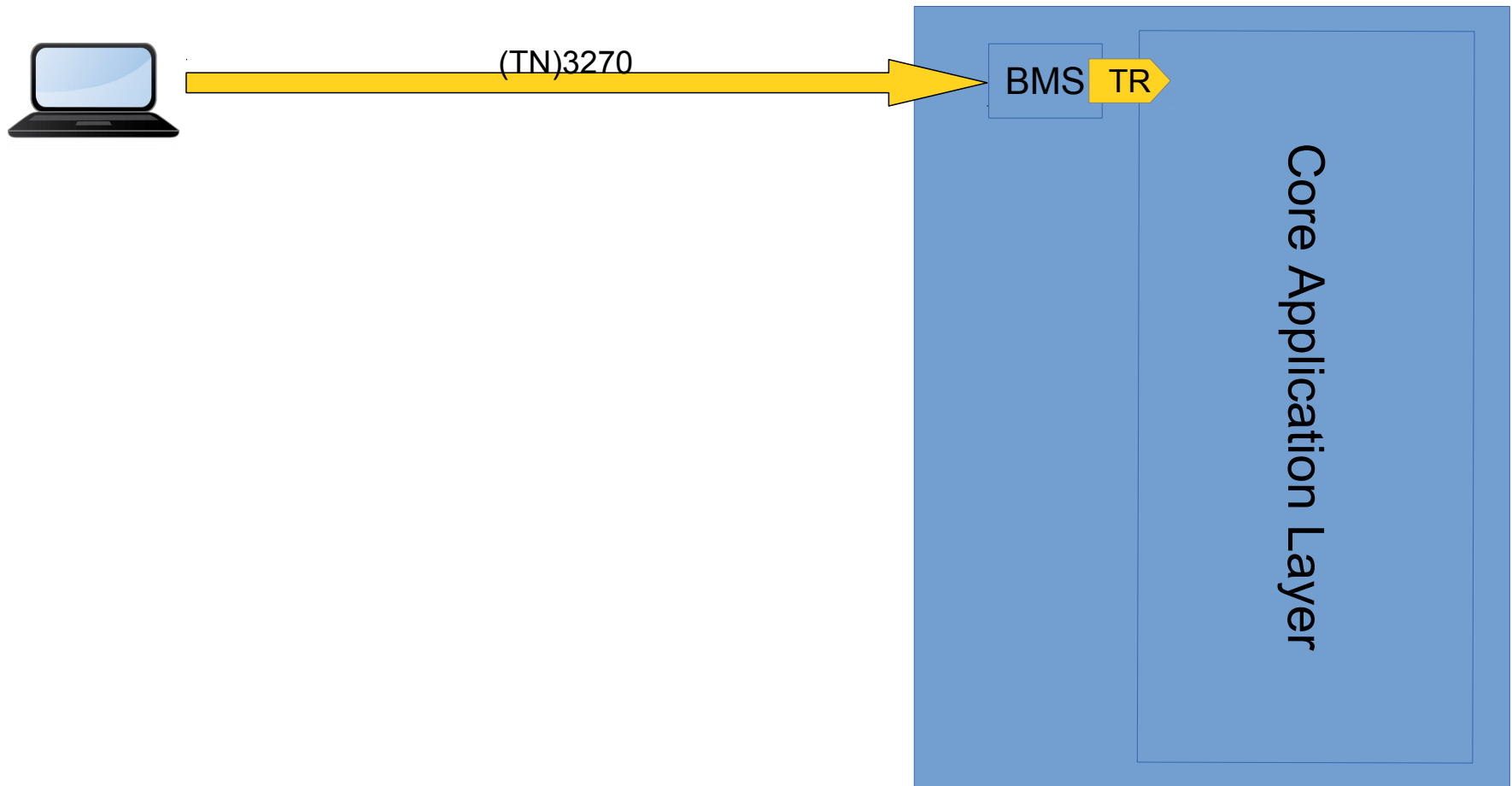


- URIMAP provides CICS authorisation via Transaction Security

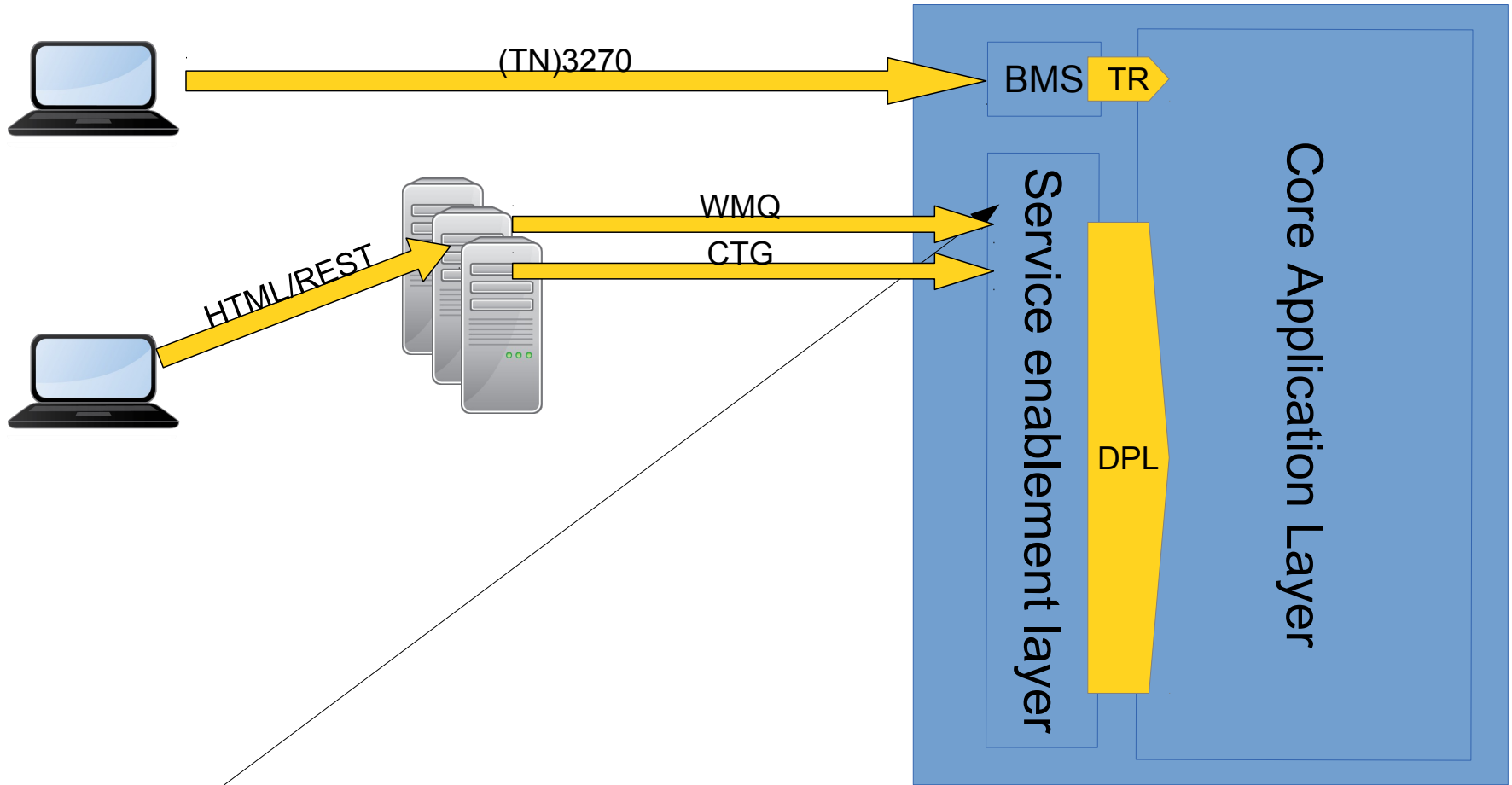
- URIMAP allows context switch to a 'user' transaction
 - Transaction Security (URL mapped to transaction)
 - monitoring and audit purposes.
 - “Transaction class” support



Adding the Web container to the Connectivity landscape



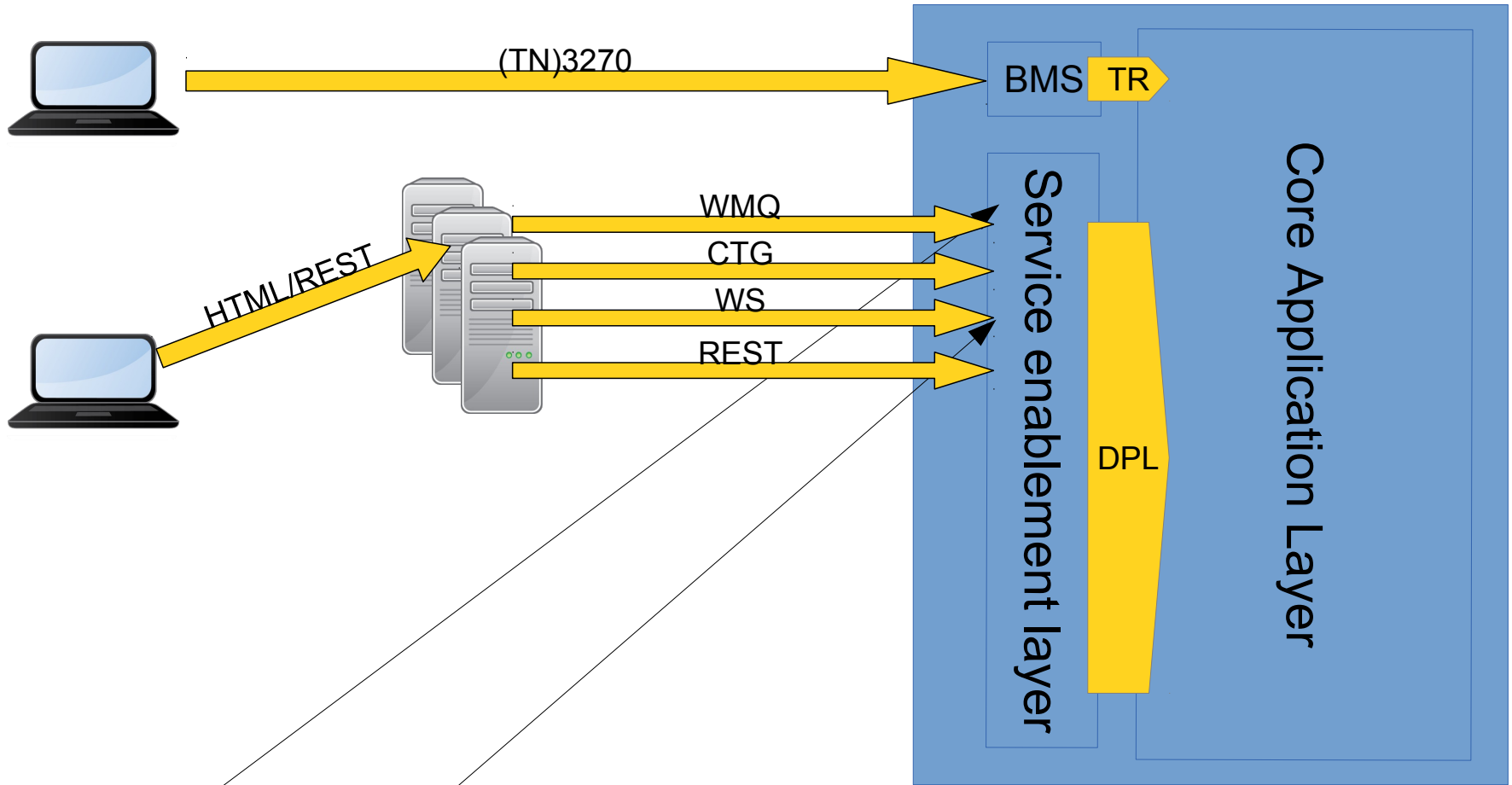
Adding the Web container to the Connectivity landscape



WMQ Triggering
& DPL Bridge

Complete your sessions evaluation online at SHARE.org/BostonEval

Adding the Web container to the Connectivity landscape

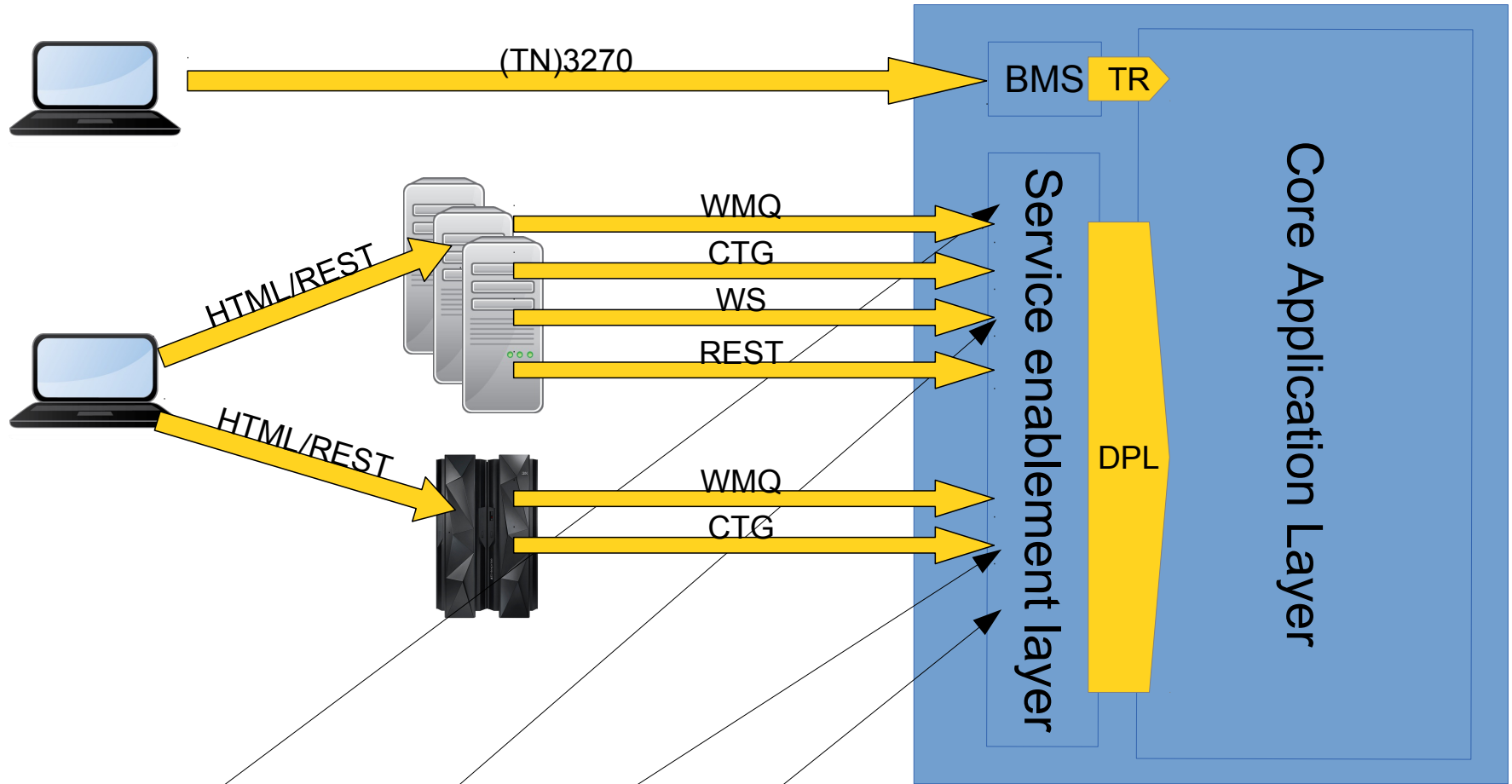


WMQ Triggering
& DPL Bridge

Native WS
& Axis2

Complete your sessions evaluation online at SHARE.org/BostonEval

Adding the Web container to the Connectivity landscape



WMQ Triggering
& DPL Bridge

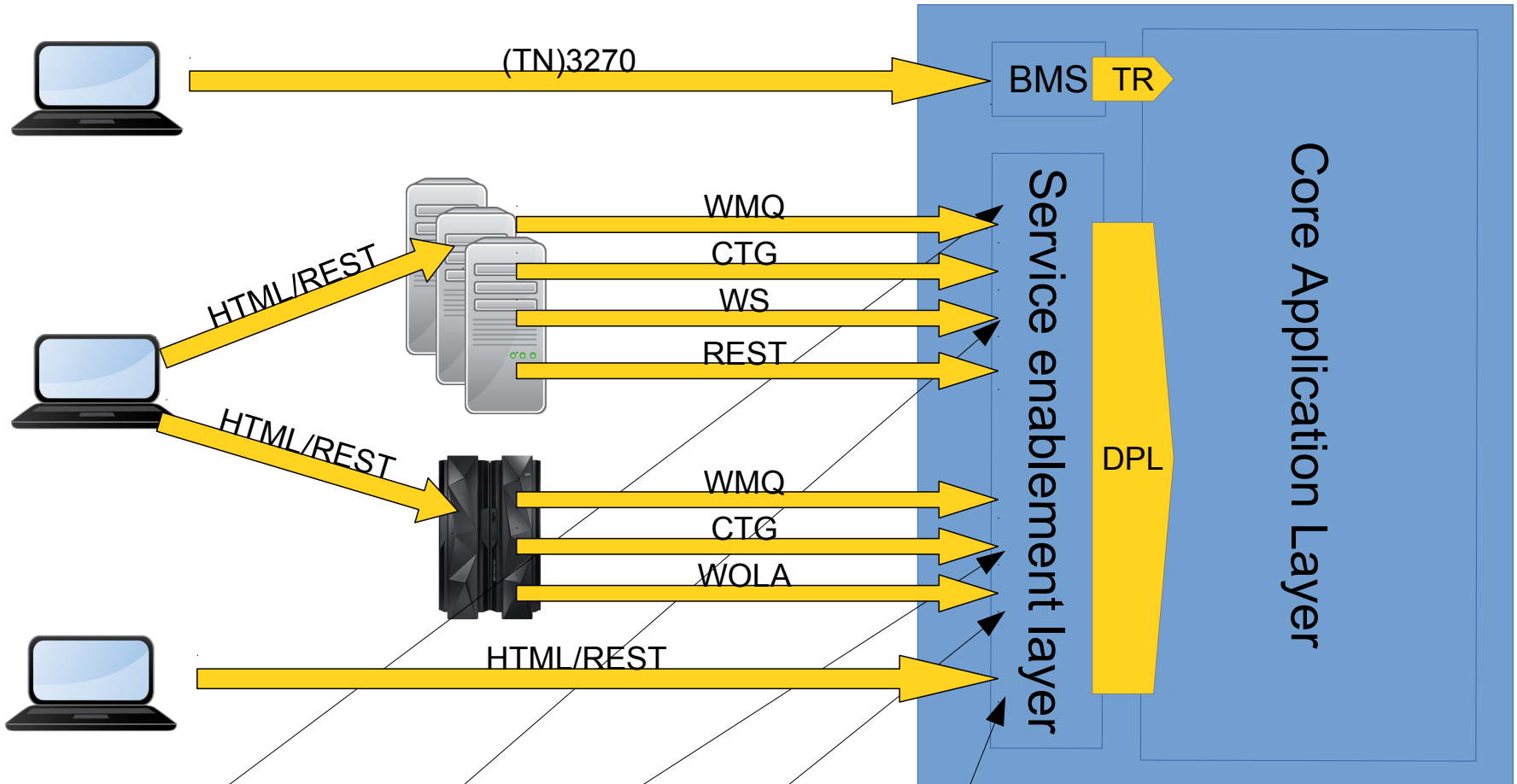
Native WS
& Axis2

IPIC &
EXCI

WOLA
TRUE

Complete your sessions evaluation online at SHARE.org/BostonEval

Adding the Web container to the Connectivity landscape



WMQ Triggering & DPL Bridge Native WS & Axis2 IPIC & EXCI WOLA TRUE Web Container (built on Liberty)

Complete your sessions evaluation online at SHARE.org/BostonEval

Part 4 – Summary and Future

Liberty Features (as of WAS 8.5.0)

- Bean validation
- Blueprint
- Java Database Connectivity (JDBC)
- Java Management Extensions (JMX)
- Java Persistence API (JPA)
- JavaServer Faces (JSF)
- JavaServer Pages (JSP)
- JAX-RS
- Secure Sockets Layer (SSL)
- Security, supported by either the basic user registry or a Lightweight Directory Access Protocol (LDAP) user registry
- Servlet
- Web application bundle (WAB)
- Web security
- zOS Security
- zOS Transactions

Liberty Features (for CICS TS V5.1 GA)

- Bean validation
- **Blueprint – via Service Stream**
- **Java Database Connectivity (JDBC)**
- Java Management Extensions (JMX)
- Java Persistence API (JPA)
- **JavaServer Faces (JSF)**
- **JavaServer Pages (JSP)**
- **JAX-RS, JSON – via Service Stream**
- **Secure Sockets Layer (SSL)**
- Security, supported by either the basic user registry or a Lightweight Directory Access Protocol (LDAP) user registry
- **Servlet**
- **Web application bundle (WAB) – via Service Stream**
- **Web security**

Summary of Key Benefits

Local. Lightweight. Fast. Web Applications run locally in CICS with direct access to CICS data and resources. No adapters, no converters, same address space.

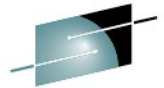
Standard tools for developers. Familiar, industry standard tools with Eclipse and Dynamic Web Projects. CICS Explorer SDK enhances the deployment experience.

Portable. Presentation logic in Servlets, business logic in OSGi bundles. Servlets are portable across runtimes. Bundles provide componentization.

Modular design. Architected in a modular way using OSGi, the server only enables and starts the features required by the applications and configuration. If you're not using a feature, it won't start in your server runtime

Dynamic runtime. Features can be added to the server dynamically, using the OSGi framework, while the server is running, with zero downtime and server restarts. Similarly server and application config can be updated without the need to restart.

Eclipse based tools. The eclipse tools for the Liberty Profile are small and very well integrated with the Liberty Profile environment



IARE
y - Connections - Results

धन्यवाद

Hindi

多謝

Traditional Chinese

ขอบคุณ

Thai

Спасибо

Russian

Thank You

English

Bedankt

Nederlands

شكراً

Arabic

Merci

French

Obrigado

Brazilian Portuguese

Gracias!

Spanish

多谢

Simplified Chinese

Danke

German

நன்றி

Tamil

ありがとうございました

Japanese

감사합니다

