

CICS V5.1

Operational Efficiency and Service
Agility with Cloud Enablement



CICS Version 5.1

Liberty meets CICS

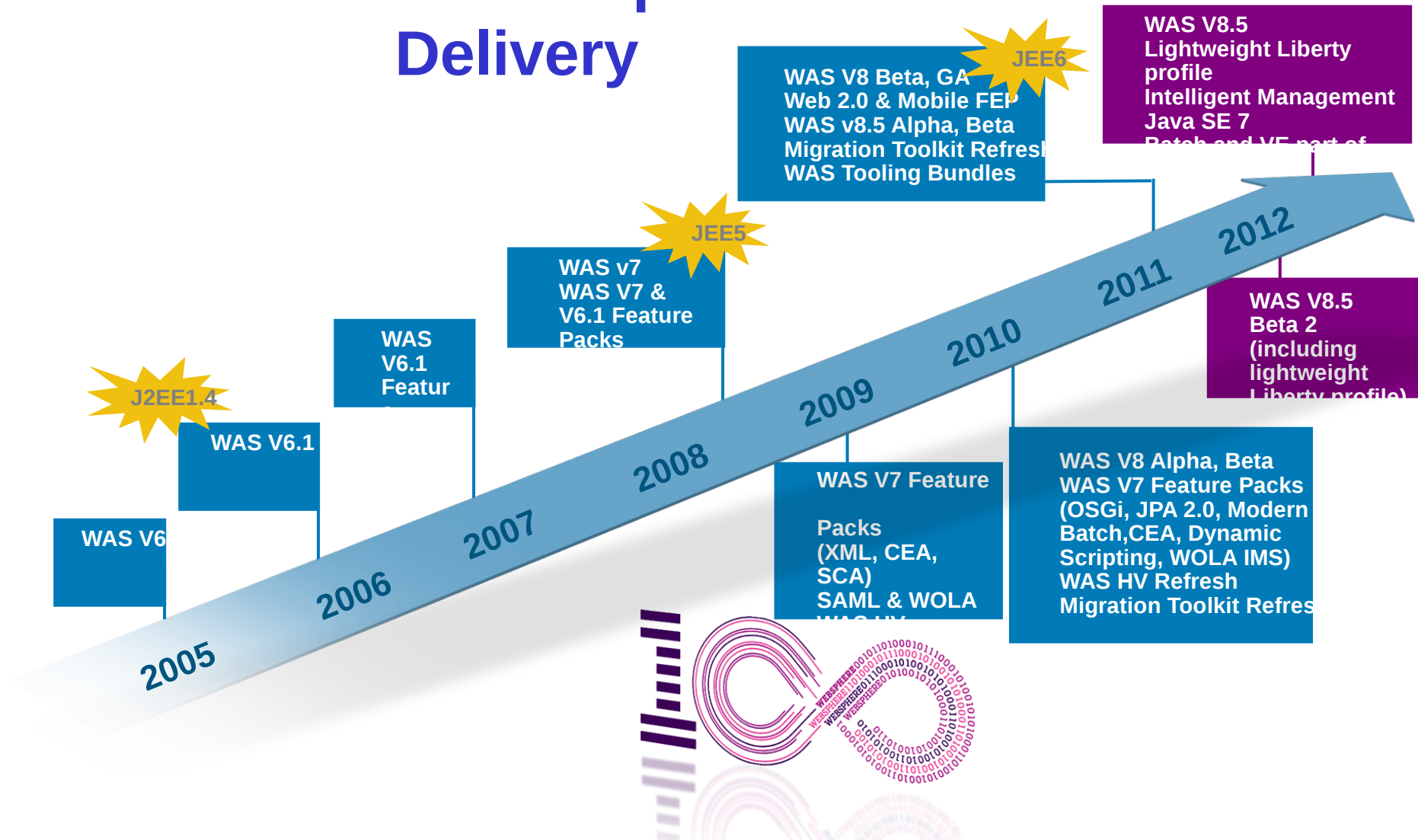


Tobias Leicher | CICS Client Technical Professional | tobias.leicher@de.ibm.com

© 2012 IBM Corporation

Part 1 – Introduction to the CICS Java Web Container based on WAS Liberty technology

WebSphere Application Server: 14 Years of Leadership and Trusted Delivery



What's Liberty?



Liberty is...



A LIGHTWEIGHT

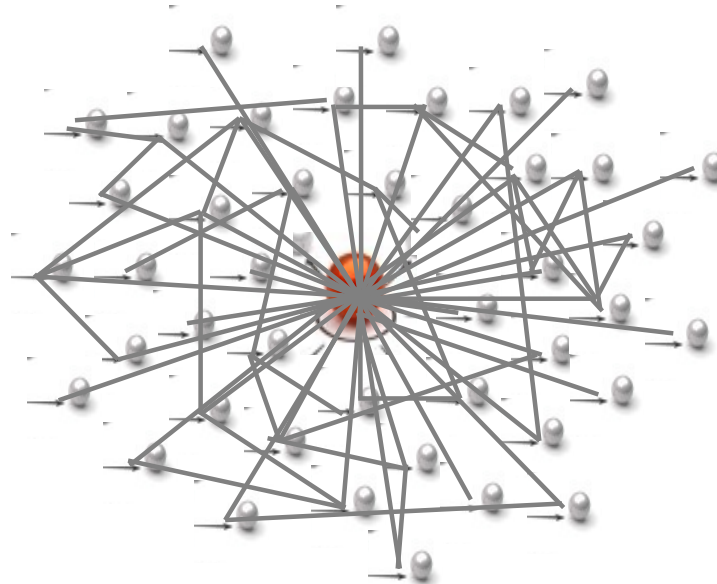


COMPOSABLE

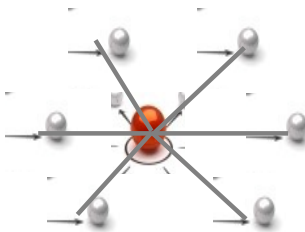


... 'Profile'
of WebSphere Application Server

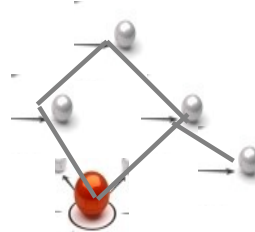
If this is tWAS...



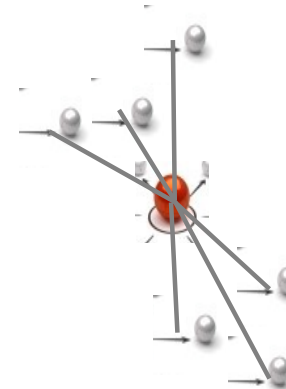
...this is Liberty (WAS)



...so is this



...or even this!



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>
```

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*

What is the 'Liberty' Profile?

- A transportable runtime for your applications
 - » Use “server package” to generate an archive that contains a tested, self-contained, pre-configured server instance that includes your application
- Enables an application-centric deployment model that allows for easy scale-out
- Light-touch admin builds on the ND job manager infrastructure to manage Liberty server instances
- A runtime environment with fidelity to full WAS
 - » Liberty is WebSphere
 - » Applications that are developed and tested on Liberty will run on the full profile

... YOU compose the runtime from the 'Features' you want.
... not the ones you don't.



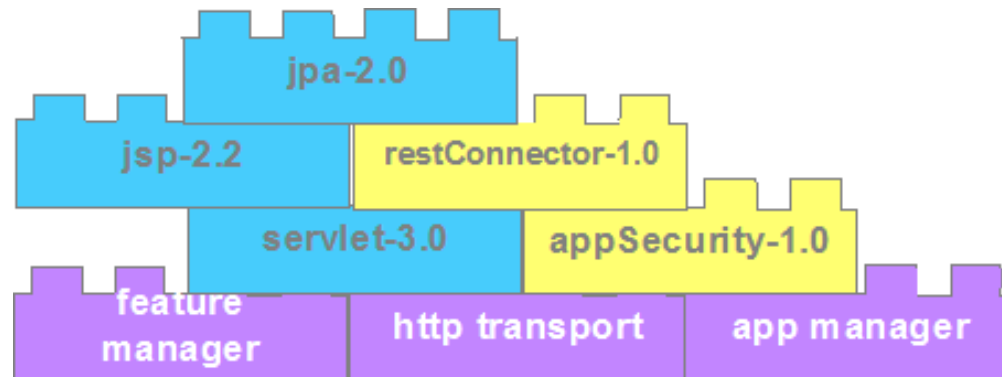
- “CICS TS V5.1 open beta offers a fast and lightweight Java web container, providing developers with the rich features of the Java Servlet and JavaServer Pages (JSP) specifications, and fast local access to your existing CICS applications and data. Built on WebSphere Application Server Liberty technology, 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.

Composability – Based on features

```
<server description="composabilityIsKey">  
  <featureManager>  
    <feature>appSecurity-1.0</feature>  
    <feature>jsp-2.2</feature>  
    <feature>restConnector-1.0</feature>  
    <feature>jpa-2.0</feature>  
  </featureManager>  
</server>
```



Liberty and traditional/full profile

There are functional differences between traditional WAS and the Liberty profile – Liberty provides a useful subset of traditional WAS

Liberty Profile

- Bean validation
- Blueprint
- Java API for RESTful Web Services
- Java Database Connectivity (JDBC)
- Java Naming and Directory Interface (JNDI)
- Java Persistence API (JPA)
- Java Server Faces (JSF)
- Java Server Pages (JSP)
- JMX
- Monitoring
- OSGi JPA
- Remote connector
- Secure Sockets Layer (SSL)
- Security
- Servlet
- Session Persistence
- Transaction
- Web application bundle (WAB)
- z/OS Security (SAF)
- z/OS Transactions (RRS)
- z/OS Workload Management

Traditional WAS Profile

Everything Liberty has...

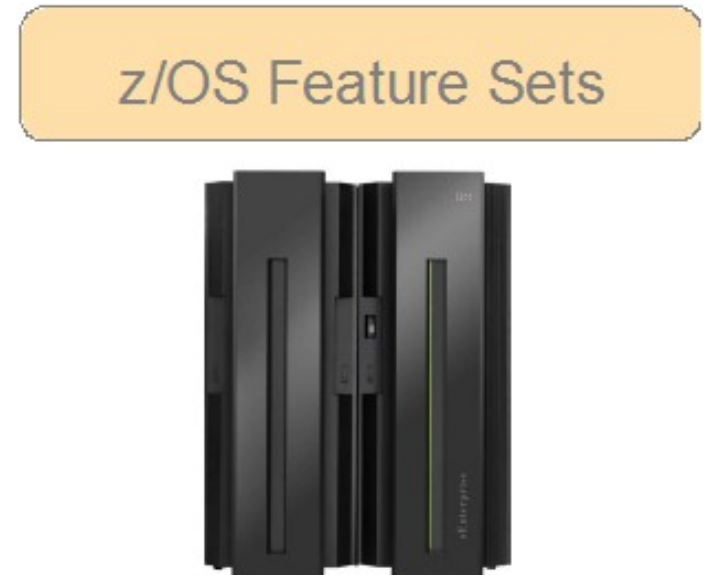
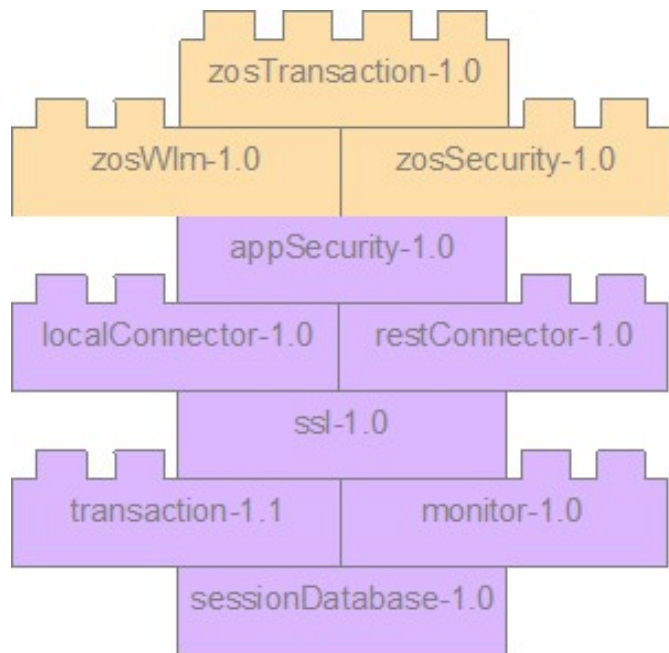


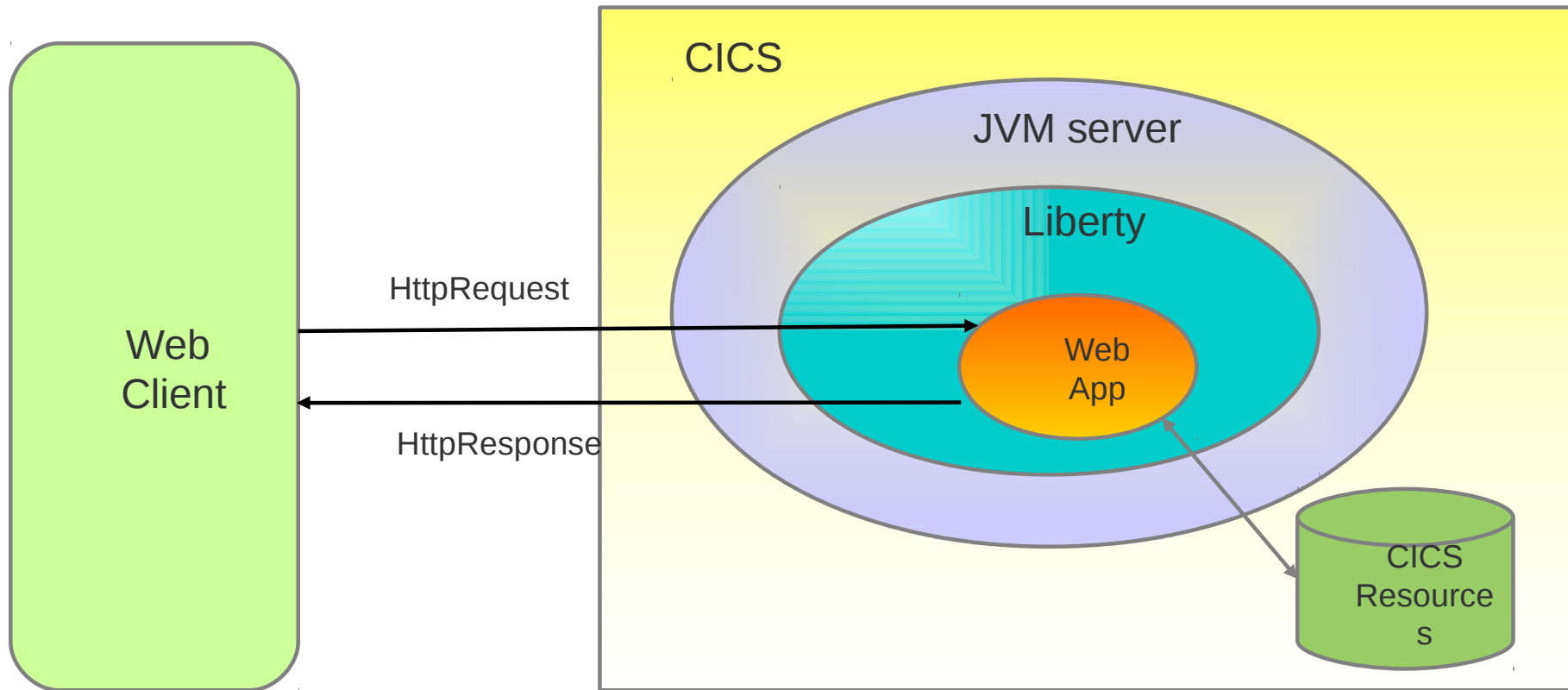
- Enterprise Java Beans (EJBs)
- Messaging (JMS)
- Web Services
- Service Component Arch (SCA)
- Java Connector Architecture (JCA)
- Clustering
- WebSphere Optimized Local Adapters
- Administrative Console
- WSADMIN scripting
- Multi-JVM Server Model**

And much more ...

What is the WAS for z/OS Liberty Profile?

- The WAS for z/OS Liberty profile is Liberty with optional, independently enabled extensions that exploit z/OS facilities
- Only enable exploitation of z/OS features you need
- Only configure the z/OS functions you use
- Focus of v8.5 is basic integration and exploitation





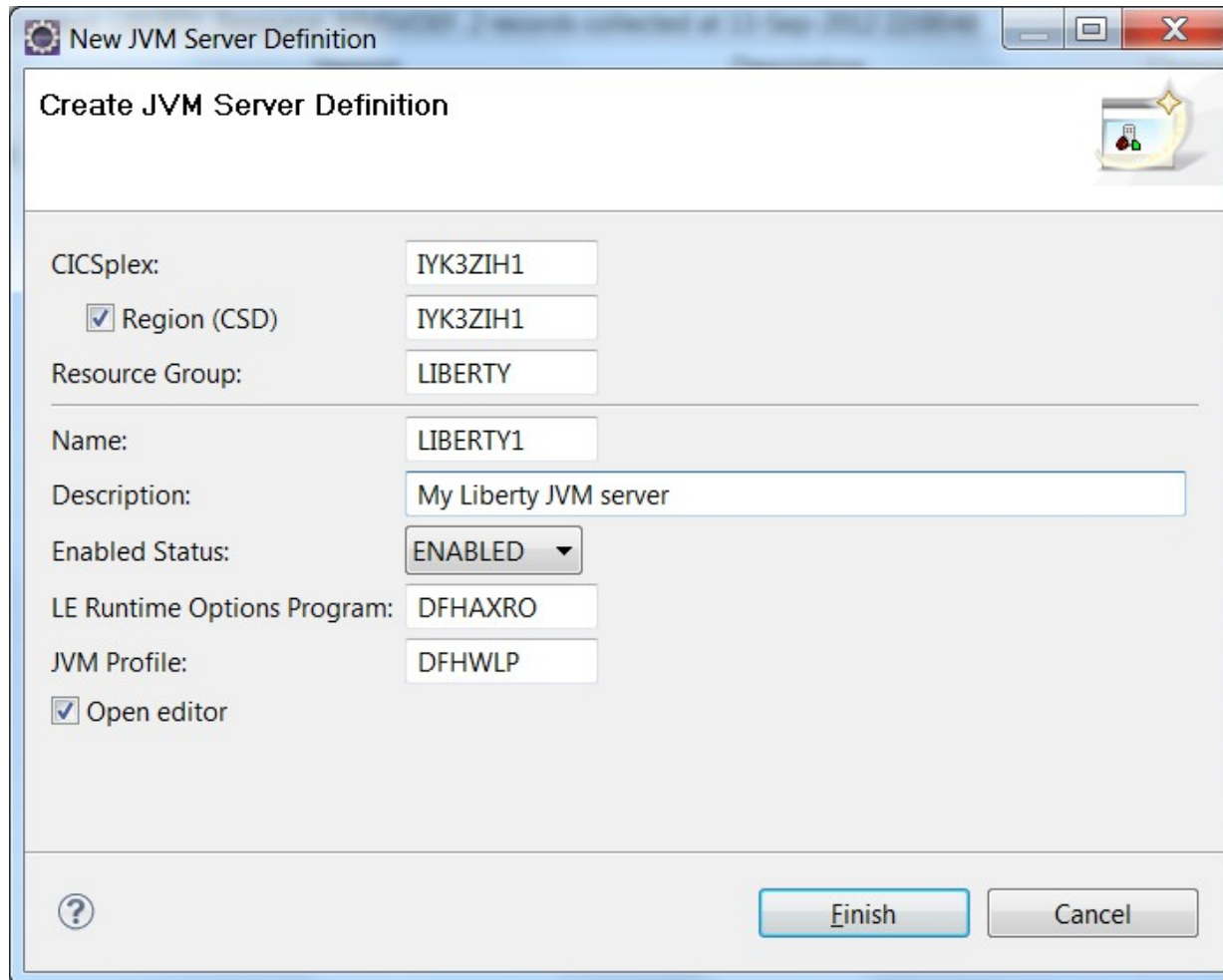
Benefits for CICS



- ✓ 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.

Part 2 – Nought to Web-App

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



The image shows a 'New JVM Server Definition' dialog box. The title bar includes a gear icon and the text 'New JVM Server Definition'. The main title is 'Create JVM Server Definition'. The dialog contains several input fields and checkboxes. The 'CICSplex:' field is set to 'IYK3ZIH1'. The 'Region (CSD)' checkbox is checked, and its field is also set to 'IYK3ZIH1'. The 'Resource Group:' field is set to 'LIBERTY'. The 'Name:' field is set to 'LIBERTY1'. The 'Description:' field contains the text 'My Liberty JVM server'. The 'Enabled Status:' dropdown menu is set to 'ENABLED'. The 'LE Runtime Options Program:' field is set to 'DFHAXRO'. The 'JVM Profile:' field is set to 'DFHWLP'. The 'Open editor' checkbox is checked. At the bottom left is a help icon (question mark). At the bottom right are 'Finish' and 'Cancel' buttons.

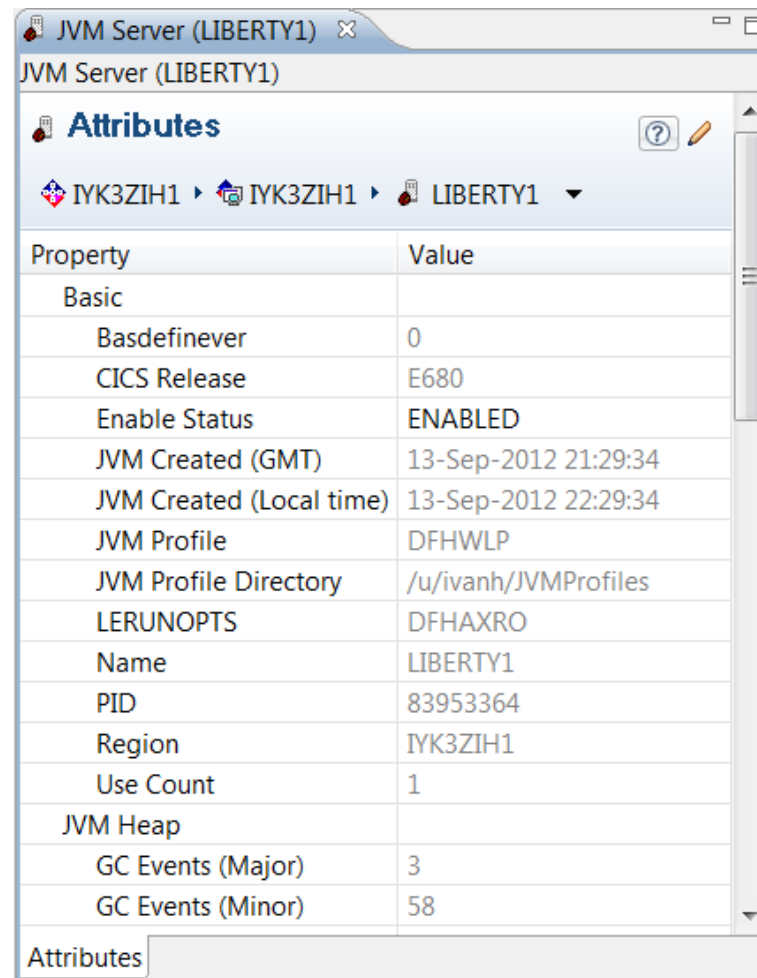
CICSplex:	IYK3ZIH1
<input checked="" type="checkbox"/> Region (CSD)	IYK3ZIH1
Resource Group:	LIBERTY
Name:	LIBERTY1
Description:	My Liberty JVM server
Enabled Status:	ENABLED
LE Runtime Options Program:	DFHAXRO
JVM Profile:	DFHWLP
<input checked="" type="checkbox"/> Open editor	

Finish Cancel

2) 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

3) Enable the JVM server



JVM Server (LIBERTY1)

JVM Server (LIBERTY1)

Attributes

YK3ZIH1 ▶ YK3ZIH1 ▶ 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	YK3ZIH1
Use Count	1
JVM Heap	
GC Events (Major)	3
GC Events (Minor)	58

Attributes

4) 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/installedApps.xml

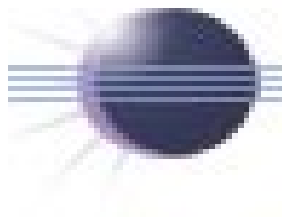
[AUDIT] CWWKG0028A: Processing included configuration resource:
file:/u/ivanh/IYK3ZIH1/LIBERTY1/wlp/usr/servers/defaultServer/cicsSecurity.xml

[AUDIT] CWWKZ0058I: Monitoring dropins for applications.

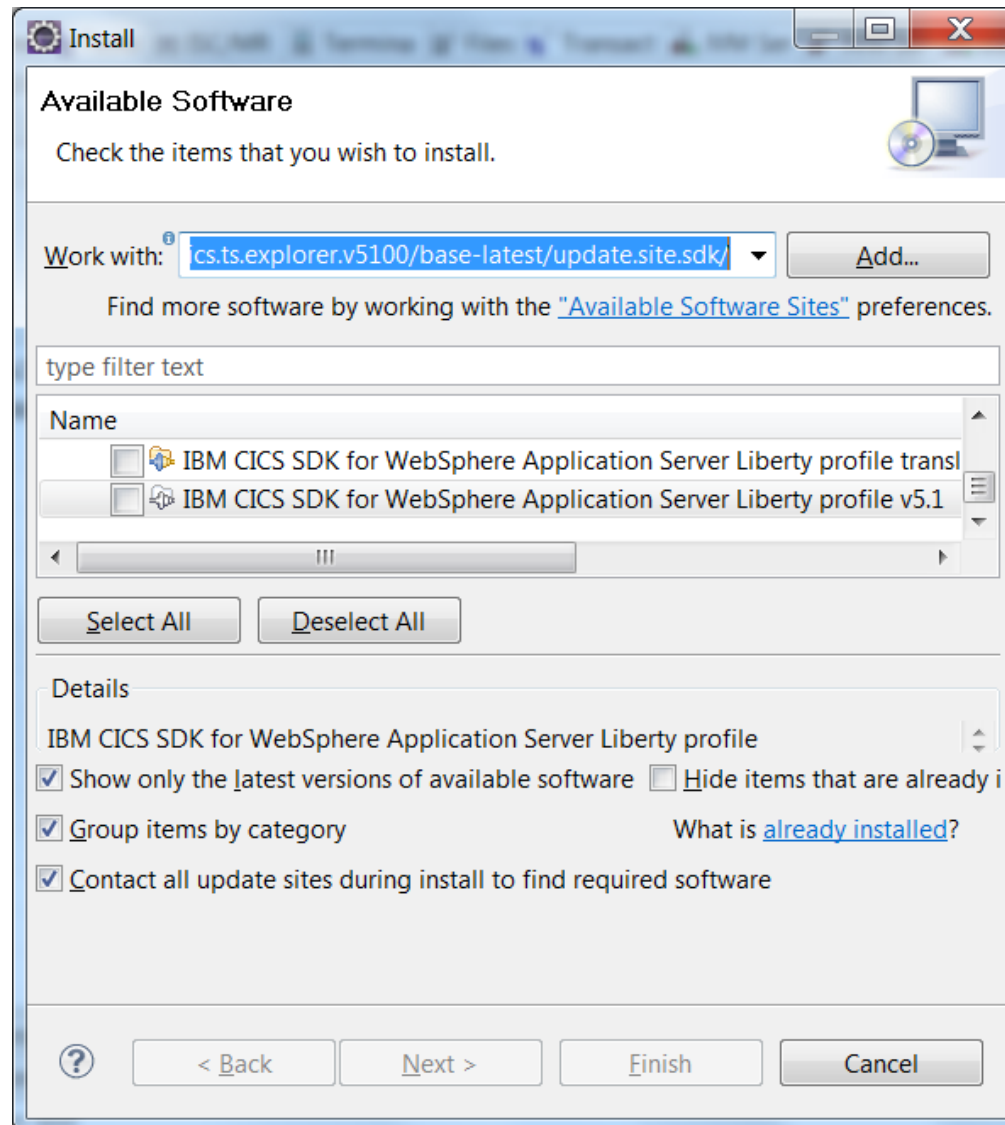
[AUDIT] CWWKF0011I: The server defaultServer is ready to run a smarter planet.

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

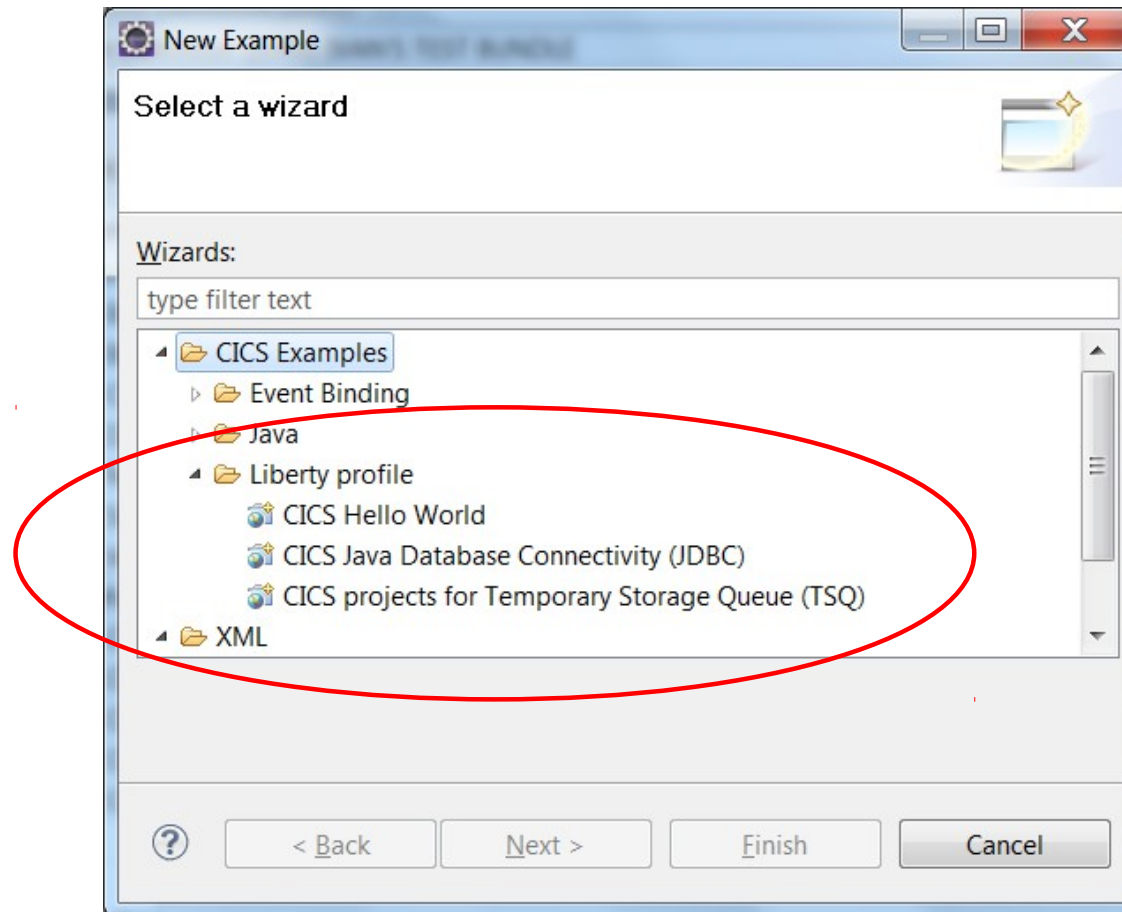
<http://www.eclipse.org/downloads/packages/release/helios/sr2>



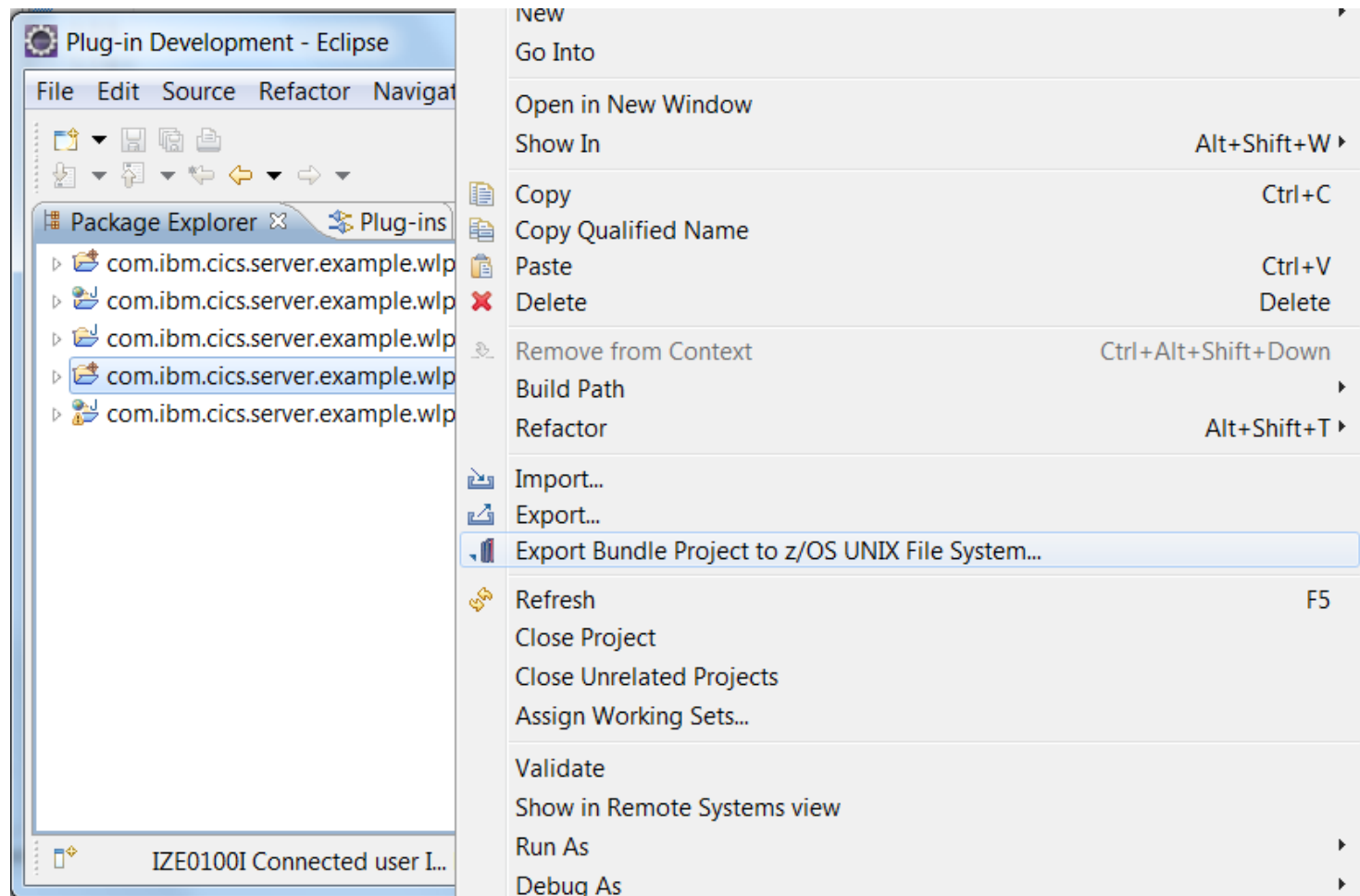
6) Install IBM CICS SDK for WebSphere Application Server Liberty profile v5.1



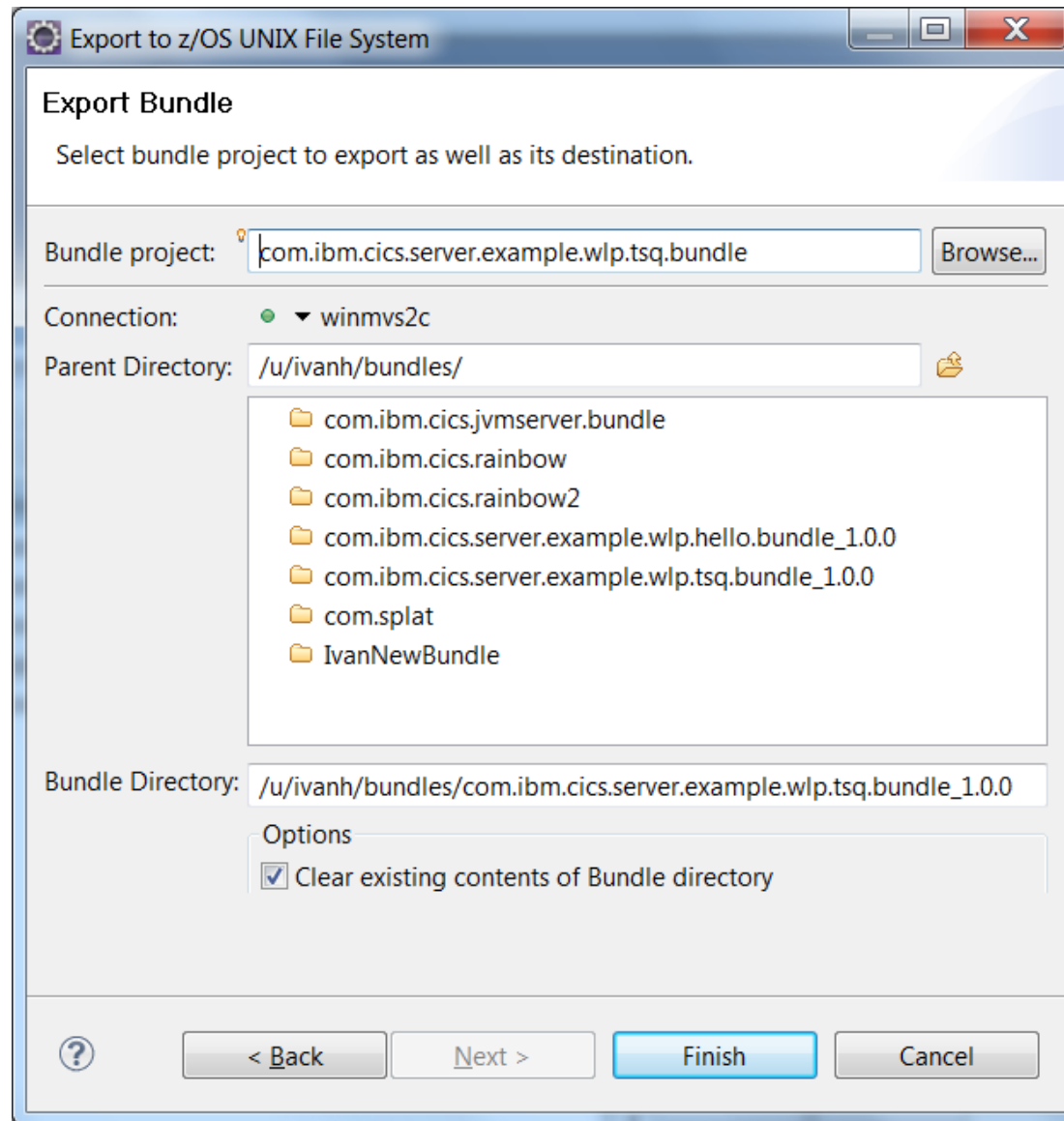
7) Create a Dynamic Web Project, or choose one of the Examples



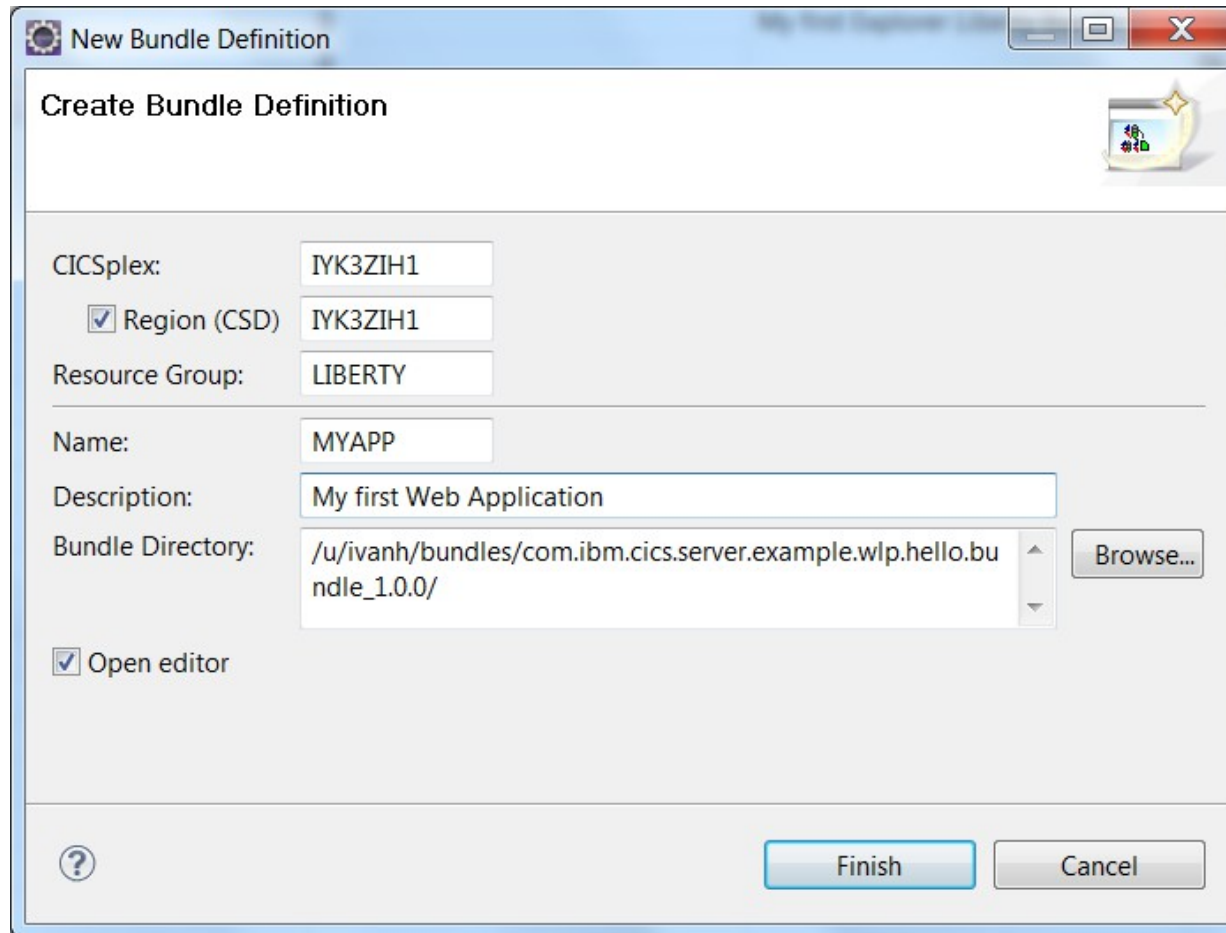
8) Export the CICS bundle project



9) Pick a zFS location for the CICS bundle project



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

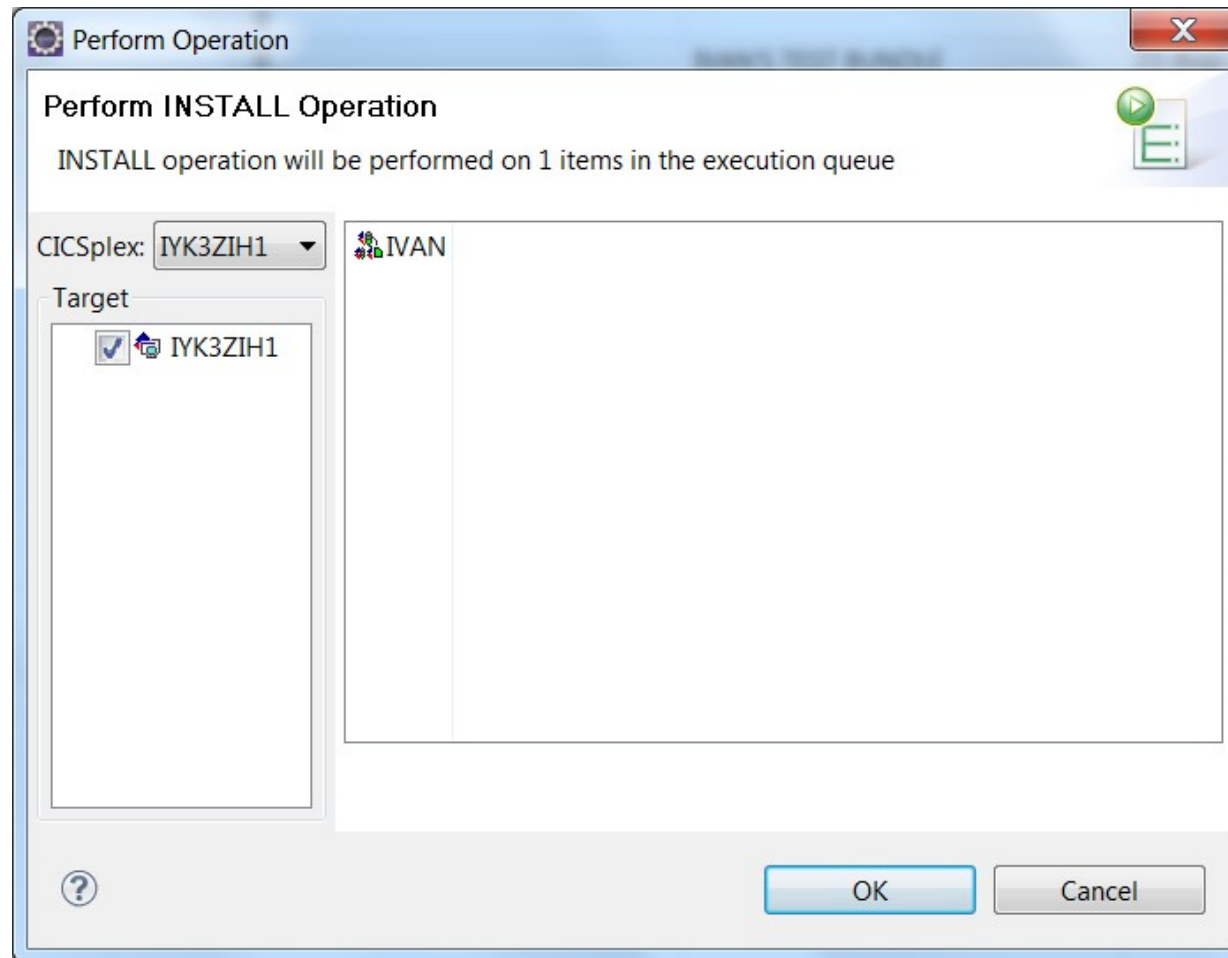


The image shows a 'New Bundle Definition' dialog box with the following fields and options:

- 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**

Buttons: **Finish**, **Cancel**, **Browse...**

11) Install the CICS bundle definition



12) Run the application!

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



Eclipse with Liberty Tools



Part 3 – The Technology

A white dove is shown in flight, its wings spread wide, against a background of a bright blue sky filled with soft, white clouds. The dove is positioned on the left side of the frame, flying towards the right.

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.

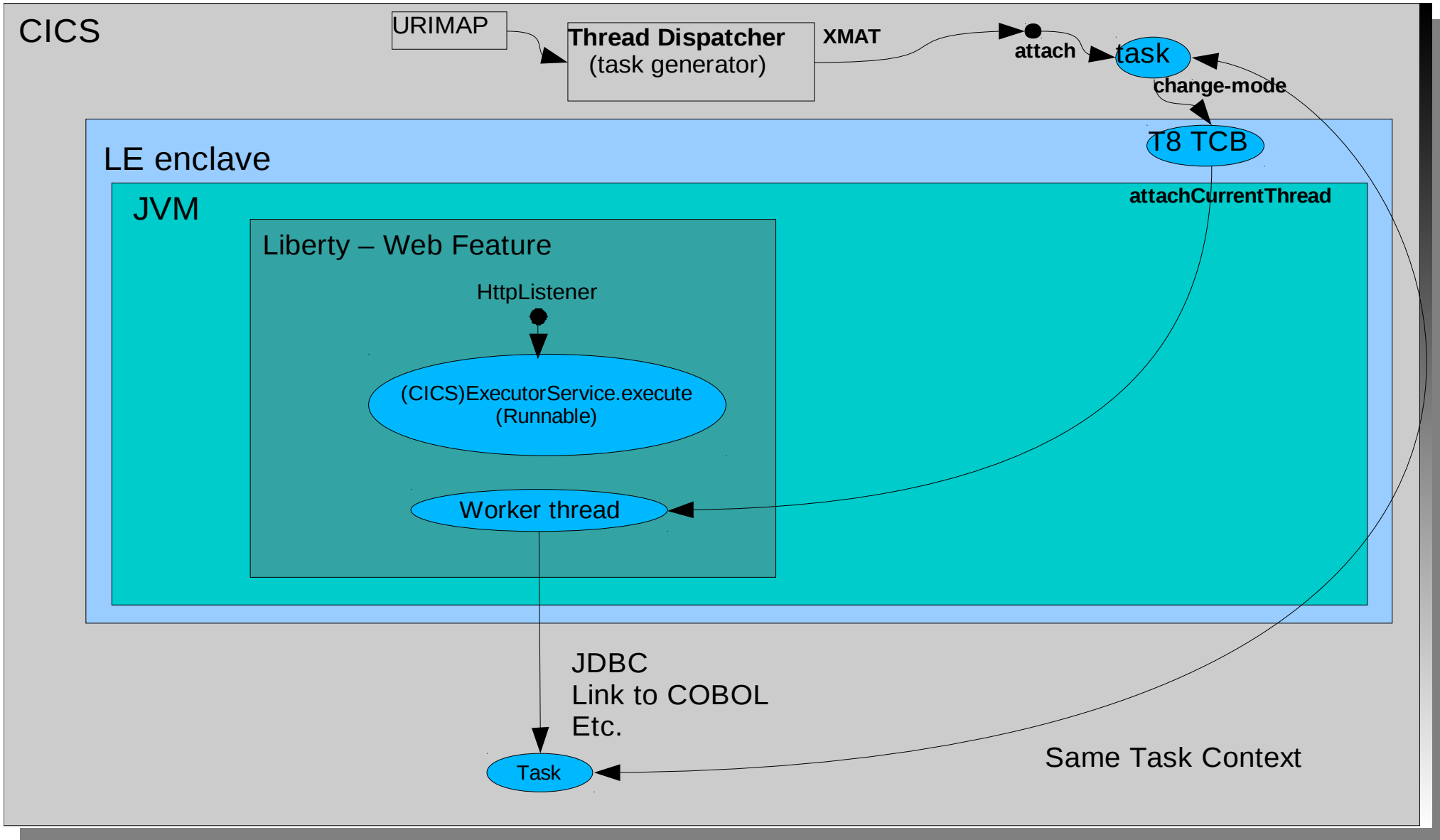
SPECIFICATIONS



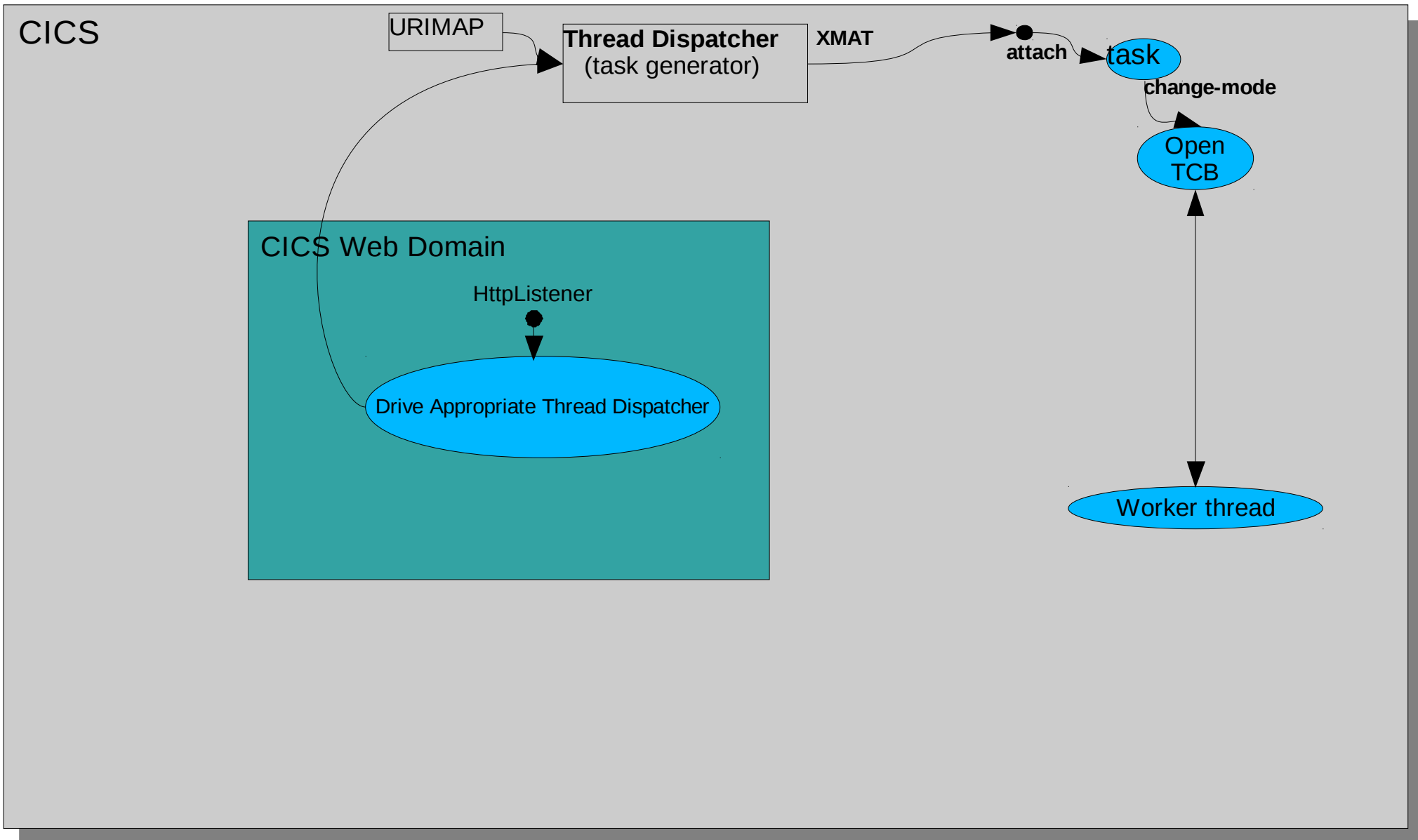
- Java 7 (64-bit)
- Equinox 3.7 as the OSGi framework.
 - Implements the OSGi R4.3 specification
- WAS Liberty Profile 8.5.0
- IBM CICS SDK for WebSphere Application Server Liberty profile v5.1
- Eclipse 3.6.2



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 Tranid, 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



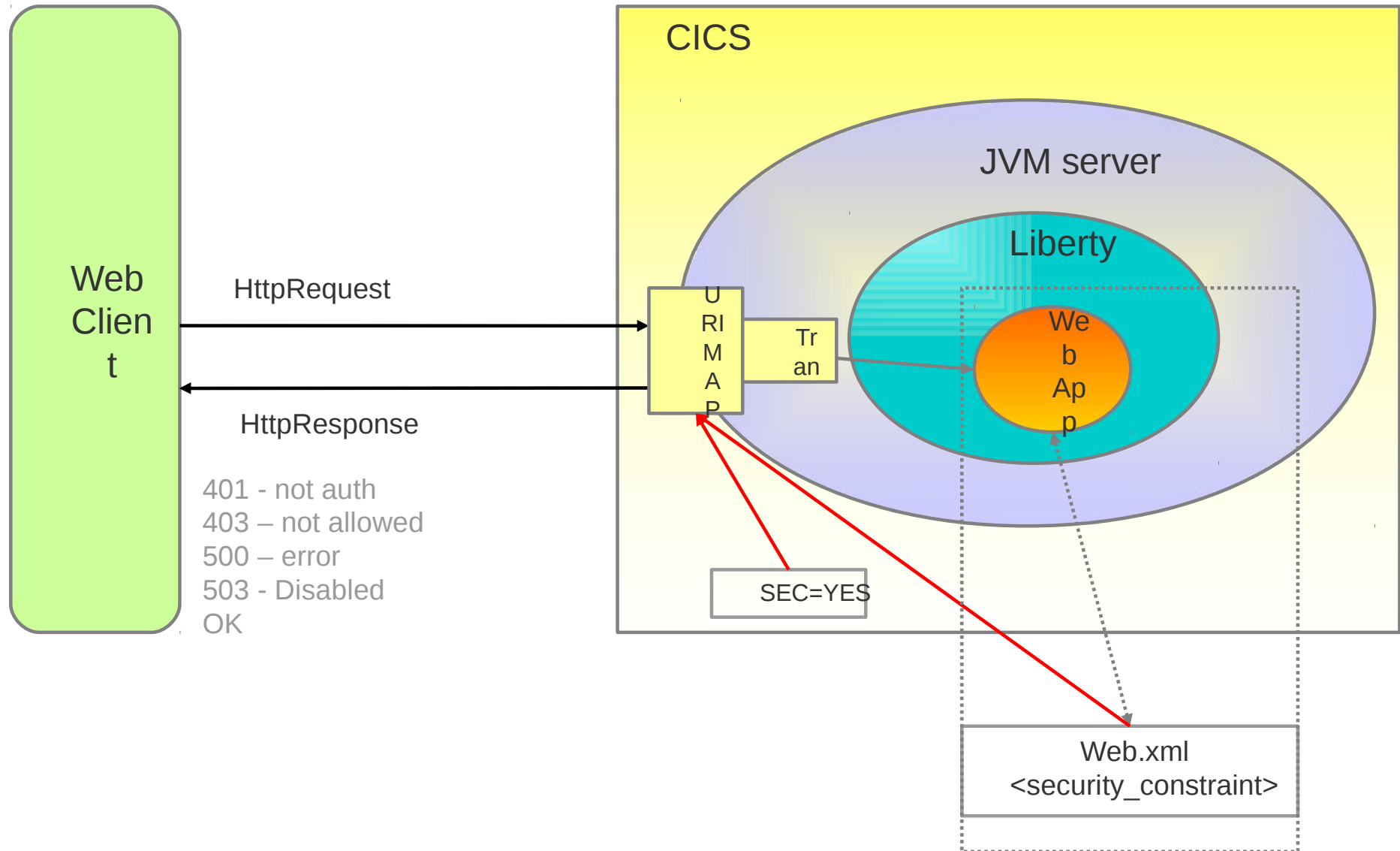
CICS Security with Liberty Profile

- Servlets run under default transaction CJSA with CICS default user.
- **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

- 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



Part 4 – Summary and Future

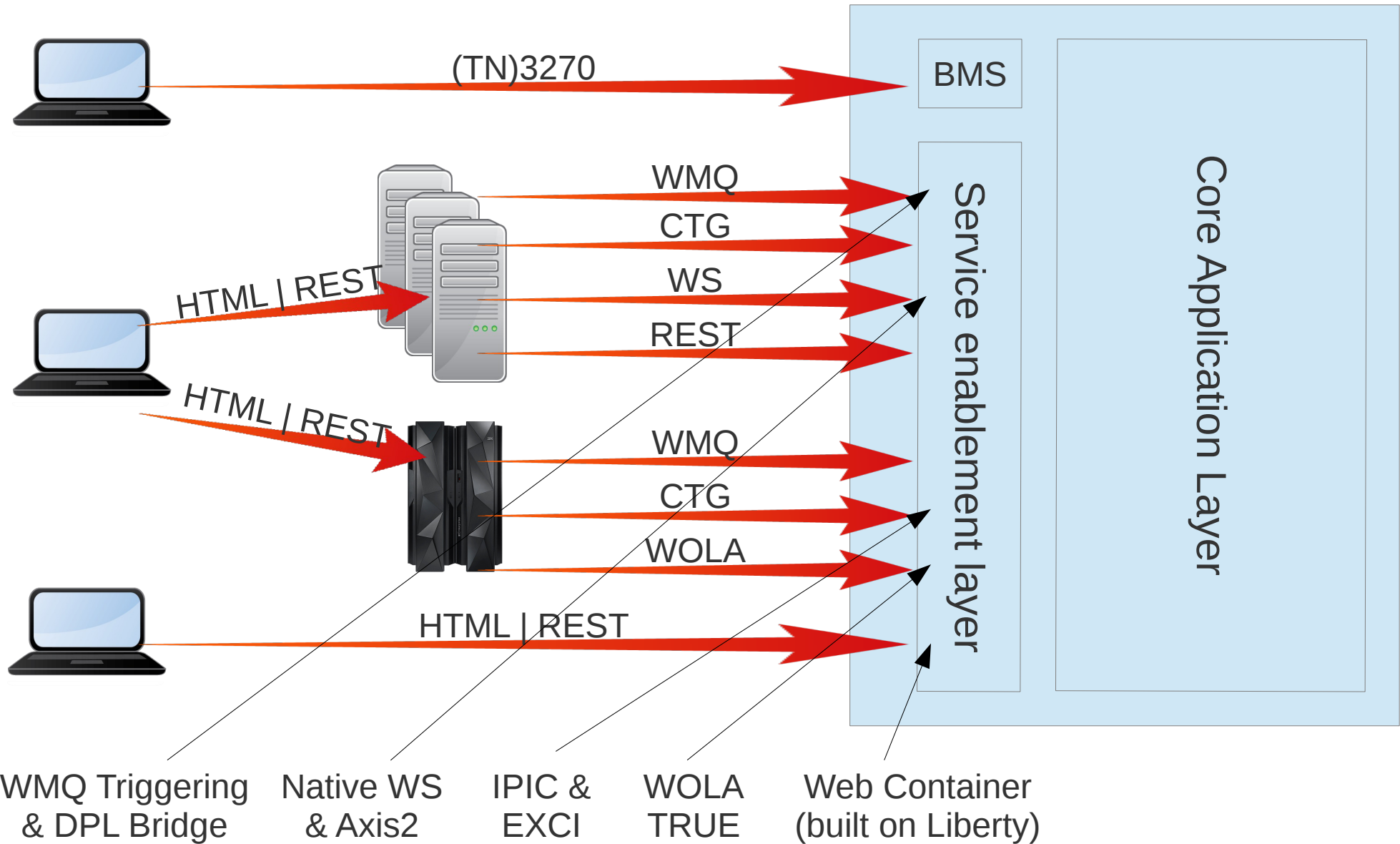
Liberty Features (as of WAS 8.5.0)

- Bean validation
- Blueprint
- Java Database Connectivity (JDBC)
- Java Management Extensions (JMX)
- Java Persistence API (JPA)
- JavaServer Faces (JSF)
- JavaServer Pages (JSP)
- 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
- **Java Database Connectivity (JDBC)**
- Java Management Extensions (JMX)
- Java Persistence API (JPA)
- **JavaServer Faces (JSF)**
- **JavaServer Pages (JSP)**
- **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 – **CICS Security**
- zOS Transactions – **CICS Transactions**

So what does that mean?... more choices!



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

Kontakt

Tobias Leicher

Client Technical Professional
– CICS & CICS Tools

IBM Allee 1
D-71139 Ehningen
Mobil: 0151 – 15 16 24 89
Mail: tobias.leicher@de.ibm.com



VIELEN DANK FÜR IHRE AUFMERKSAMKEIT.