



# Liberté, Égalité, Fraternité

*A mini CICS and Websphere revolution*

*William Yates – IBM CICS Test Architect*



#SHAREorg



SHARE is an independent volunteer-run information technology association that provides **education, professional networking and industry influence.**

Copyright (c) 2015 by SHARE Inc. Except where otherwise noted, this work is licensed under <http://creativecommons.org/licenses/by-nc-sa/3.0/>



# A French motto – and some Java

- Liberté, Égalité, Fraternité
- Liberty
  - The state of being free from oppressive restrictions
- Equality
  - The state of being equal in status rights or opportunities
- Fraternity
  - friendship and mutual support within a group.



*Liberté • Égalité • Fraternité*

**RÉPUBLIQUE FRANÇAISE**

# What is Liberty?



Lightweight application server created by WebSphere

Easily configurable opt-in customisation model

Runs on many platforms (Both distributed and z/OS)

Now integrated into CICS!

# What does Liberty buy me?

## What does Liberty buy me?



Support for the Java EE 6 Web Profile

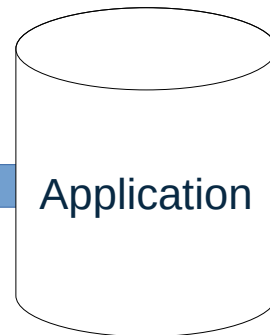
Support for a number of Java EE technologies

HTTP endpoint for applications

Ability to closely integrate front end applications with CICS

# Portability is the game!

Distributed Liberty



CICS Liberty



# Why put Liberty into CICS

1. Provide a rich java environment for web application development

More than just the base SDK

Frameworks – servlets, JPA, etc

2. Two proven stable technologies

JVMServer in CICS since version 4 (replacing pooled JVM)

Liberty profile based on WAS

3. Allow colocation of Java applications and CICS applications

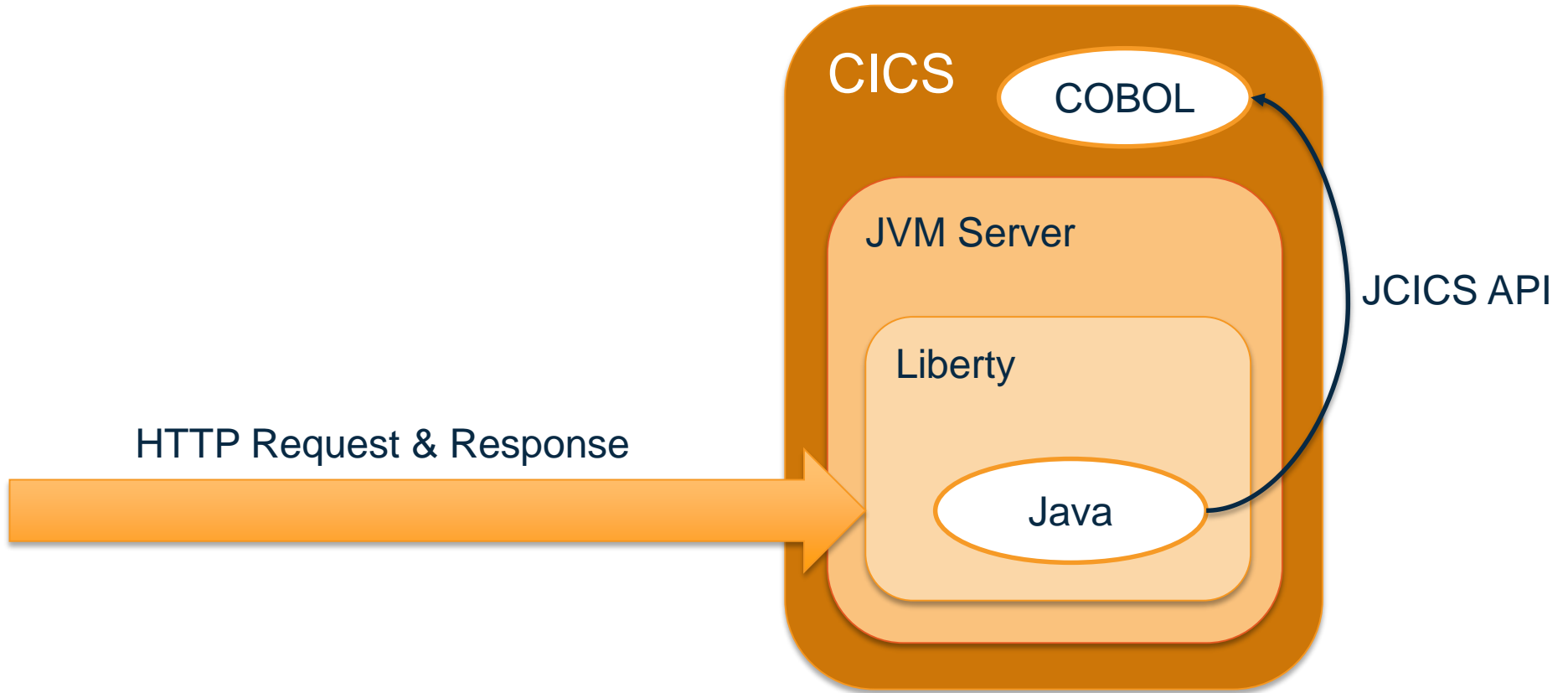
Reduction of network lag between liberty and trusted applications

# Why USE Liberty in CICS

1. Extension of SOA enablement
  1. Web Services REST JSON
2. Co-locating applications closer to the data
  1. Why run it on distributed when the data isn't?
3. API economy
  1. Change granularity of services
4. Extend applications

A richer java environment to do all of the above!

# Java in CICS





# Areas to discuss

1. Lets get started!

2. What is this rich environment you speak of?

# Installing Liberty into CICS

- Liberty is shipped as part of the base runtime
  - But it does need to be enabled
- Liberty is installed as a feature of a CICS JVMServer
  - No new Liberty resource type
- Small changes to JVMProfile trigger Liberty to be enabled in that JVMServer
- Definition, monitoring are the same as a JVMServer

# Extending the JVMServer

```
JAVA_HOME=/usr/lpp/java/J7.0_64  
WORK_DIR=/u/cicsuser/workdir  
OSGI_FRAMEWORK_TIMEOUT=60  
-Dfile.encoding=ISO-8859-1
```

Many familiar JVM profile options are still valid.

```
WLP_INSTALL_DIR=&USSHOME;/wlp
```

WLP\_INSTALL\_DIR specifies this JVM server as a Liberty JVM server.

```
-XhealthCenter  
-Xhealthcenter:port=12345
```

Like regular JVM servers, you can make use of health center to help you monitor your JVM server

```
-Dcom.ibm.cics.jvmserver.wlp.server.http.port=23456  
-Dcom.ibm.cics.jvmserver.wlp.server.https.port=34567
```

You can use the JVM profile to specify the ports you want to use.

```
-Dcom.ibm.cics.jvmserver.wlp.autoconfigure=true
```

Auto-configure causes Liberty to create required files on start-up (We recommend that you disable set this to false after first startup)

Complete your session evaluations online at [www.SHARE.org/Orlando-Eval](http://www.SHARE.org/Orlando-Eval)

# Installing a Liberty enabled JVM server

```
DEF JVMSERVER
OVERTYPE TO MODIFY                                CICS RELEASE = 0690
CEDA DEFINE JVMserver(                            )
  JVMserver   ==> DFHWLP
  Group       ==> LIBERTY
  DESCRIPTION ==> LIBERTY ENABLED JVM SERVER
  Status      ==> Enabled           Enabled | Disabled
  Jvmprofile  ==> DFHWLP                               (Mixed Case)
  Lerunopts   ==> DFHAXRO
  Threadlimit ==> 255                1-256
DEFINITION SIGNATURE
DEFInetime   :
CHANGETime   :
CHANGEUsrid  :
CHANGEAGEnt  :           CSDApi | CSDBatch
CHANGEAGRel  :

                                           SYSID=MCS1 APPLID=MJLMAS1

PF 1 HELP 2 COM 3 END          6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
```

# Administrating & configuring Liberty JVM servers



```
S JVMS(DFHWLP)
RESULT - OVERTYPE TO MODIFY
  Jvmserver(DFHWLP)
  Enablestatus( Enabled )
  Purgetype(          )
  Prfile(DFHWLP)
  Lerunopts(DFHAXR0)
  Threadcount(000)
  Threadlimit( 015 )
  Currentheap(43342560)
  Initheap(4M)
  Maxheap(512M)
  Gcpolicy(-Xgcpolicy:gencon)
  Occupancy(35328568)
  Pid(0000067339)
  Profiledir(/u/michaej/JVMProfiles/java7)
  Installtime(07/05/15 09:56:25)
  Installusrid(CICSUSER)
  Installagent(Csdapi)
+  Definesource(LIBERTY)
```

SYSID=MCS1 APPLID=MJLMAS1

TIME: 11.00.48 DATE: 15/05/15

PF 1 HELP 2 HEX 3 END

5 VAR

7 SBH 8 SFH

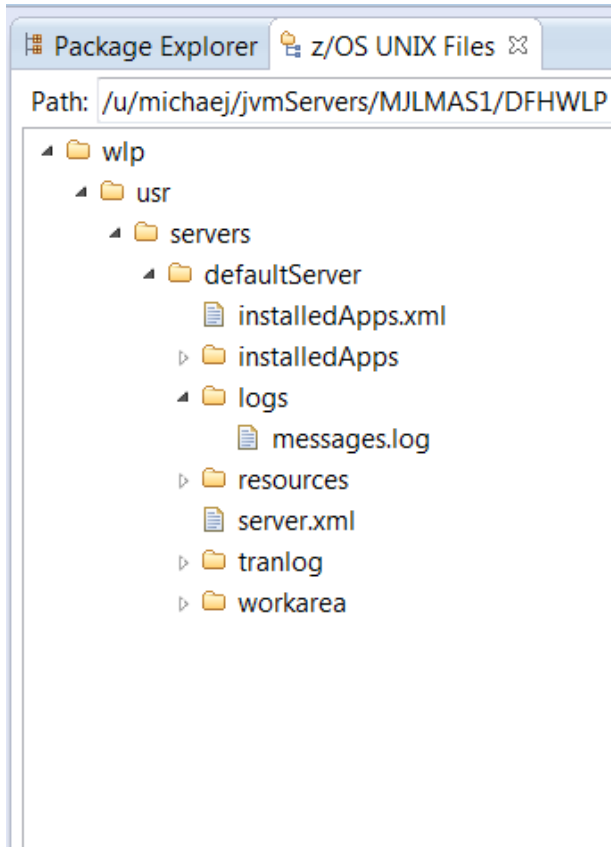
10 SB 11 SF

Complete your session evaluations online at [www.SHARE.org/Orlando-Eval](http://www.SHARE.org/Orlando-Eval)

in Orlando **2015**



## Liberty unique configuration – USS files



**WLP\_USR\_DIR** - Specifies the location where your usr directory will be created. This will contain all of your Liberty configuration files.

**WLP\_OUTPUT\_DIR** – Specifies the directory where your Liberty server will place any output files.

**InstalledApps.xml** – Installed CICS bundles are listed here

**Messages.log** – Liberty's log file

**Server.xml** – Server configuration file

**Tranlog** – Files required by Liberty to track transactions

**Workarea** – Runtime files required for Liberty

# Liberty unique configuration

```
<?XML version="1.0" encoding="UTF-8"?><server description="CICS Liberty profile sample configuration">
```

```
<!-- Enable features -->
<featureManager>
  <feature>cicsts:core-1.0</feature>
  <feature>jsp-2.2</feature>
  <feature>wab-1.0</feature>
  <feature>blueprint-1.0</feature>
  <feature>ssl-1.0</feature>
</featureManager>
```

```
<!-- Default HTTP End Point -->
```

```
<httpEndpoint host="*" httpPort="20049" httpsPort="20050" id="defaultHttpEndpoint"/>
```

```
<!-- CICS Bundle Installed Applications -->
```

```
<include location="{server.output.dir}/installedApps.xml"/>
```

```
<config monitorInterval="5s" updateTrigger="polled"/>
```

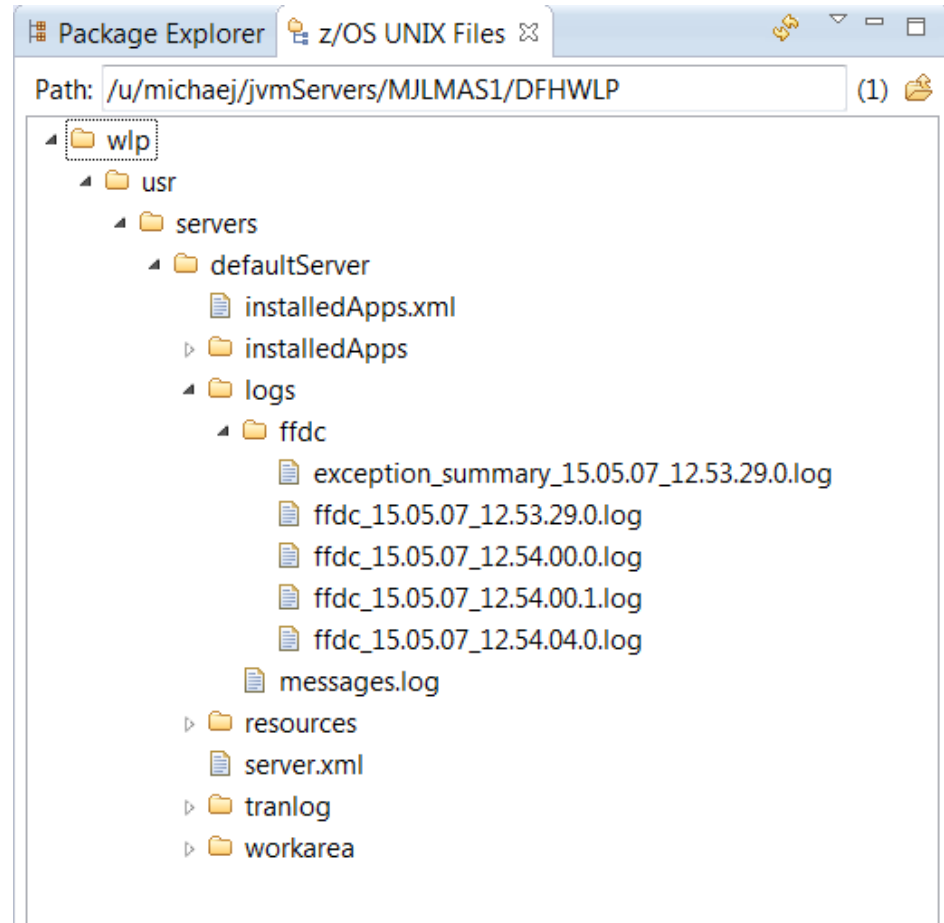
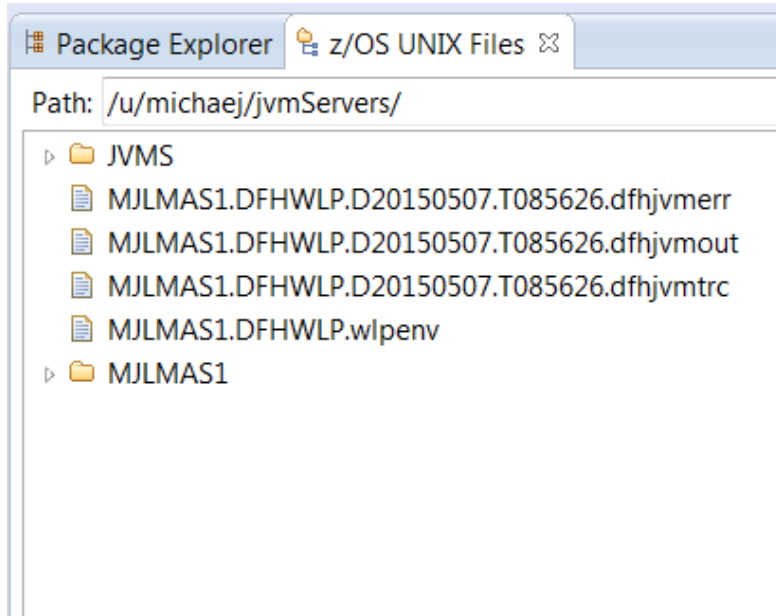
```
<applicationMonitor dropins="dropins" dropinsEnabled="false" pollingRate="5s" updateTrigger="disabled"/>
```

```
<ssl id="defaultSSLConfig" keyStoreRef="defaultKeyStore" sslProtocol="TLS"/>
```

```
<keyStore id="defaultKeyStore" password="defaultPassword"/>
```

```
</server>
```

# Liberty configuration – Logging and monitoring





# Startup messages

```
*****
product = CICS Transaction Server for z/OS 5.2.0, CICS LIBERTY NOTUSAGE, WebSphere Application Server 8.5.5.3,
WAS FOR Z/OS 8.5.5.3 (wlp-1.0.6.c150320140905-1604)
wlp.install.dir = /cics/cics690/wlp/
server.config.dir = /u/michaej/jatp/BEANV/JATP1351/JATP1351/WLPJVMs/wlp/usr/servers/defaultServer/
java.home = /java/J7.1_64
java.version = 1.7.0
java.runtime = Java(TM) SE Runtime Environment (pmz6470_27sr2fp10-20141218_02 (SR2 FP10))
os = z/OS (01.13.00; s390x) (en_US)
*****
I TRAS0018I: The trace state has been changed. The new trace state is *=info.
A CWWKE0001I: The server defaultServer has been launched.

...

I CWWK00219I: TCP Channel defaultHttpEndpoint has been started and is now listening for requests on host * (IPv6) port 29540.
A CWWKF0015I: The server has the following interim fixes installed: PI29785,PI15943,PI26809.
I CWWKF0008I: Feature update completed in 6.884 seconds.
A CWWKF0011I: The server defaultServer is ready to run a smarter planet.
```

# More messages

```
A CWWKG0016I: Starting server configuration update.
A CWWKG0028A: Processing included configuration resource: /u/michaelj/wlp/usr/servers/defaultServer/installedApps.xml
A CWWKG0017I: The server configuration was successfully updated in 0.090 seconds.
I CWWKZ0018I: Starting application cics.fv.wlp.BeanValidation.
I SRVE0169I: Loading Web Module: cics.fv.wlp.BeanValidation.
I SRVE0250I: Web Module cics.fv.wlp.BeanValidation has been bound to default_host.
A CWWKT0016I: Web application available (default_host): http://test.ibm.com:29540/cics.fv.wlp.BeanValidation/
A CWWKZ0001I: Application cics.fv.wlp.BeanValidation started in 0.809 seconds.

...

O TEST
O Couldn't read items on queue:
CICS QIDERR Condition(RESP=QIDERR, RESP2=0)
A CWWKG0016I: Starting server configuration update.
A CWWKG0028A: Processing included configuration resource: /u/michaelj/wlp/usr/servers/defaultServer/installedApps.xml
A CWWKG0017I: The server configuration was successfully updated in 0.050 seconds.
A CWWKT0017I: Web application removed (default_host): http://test.hursley.ibm.com:29540/cics.fv.wlp.BeanValidation/
I SRVE0253I: [cics.fv.wlp.BeanValidation] [/cics.fv.wlp.BeanValidation] Destroy successful.
A CWWKZ0009I: The application cics.fv.wlp.BeanValidation has stopped successfully.

...

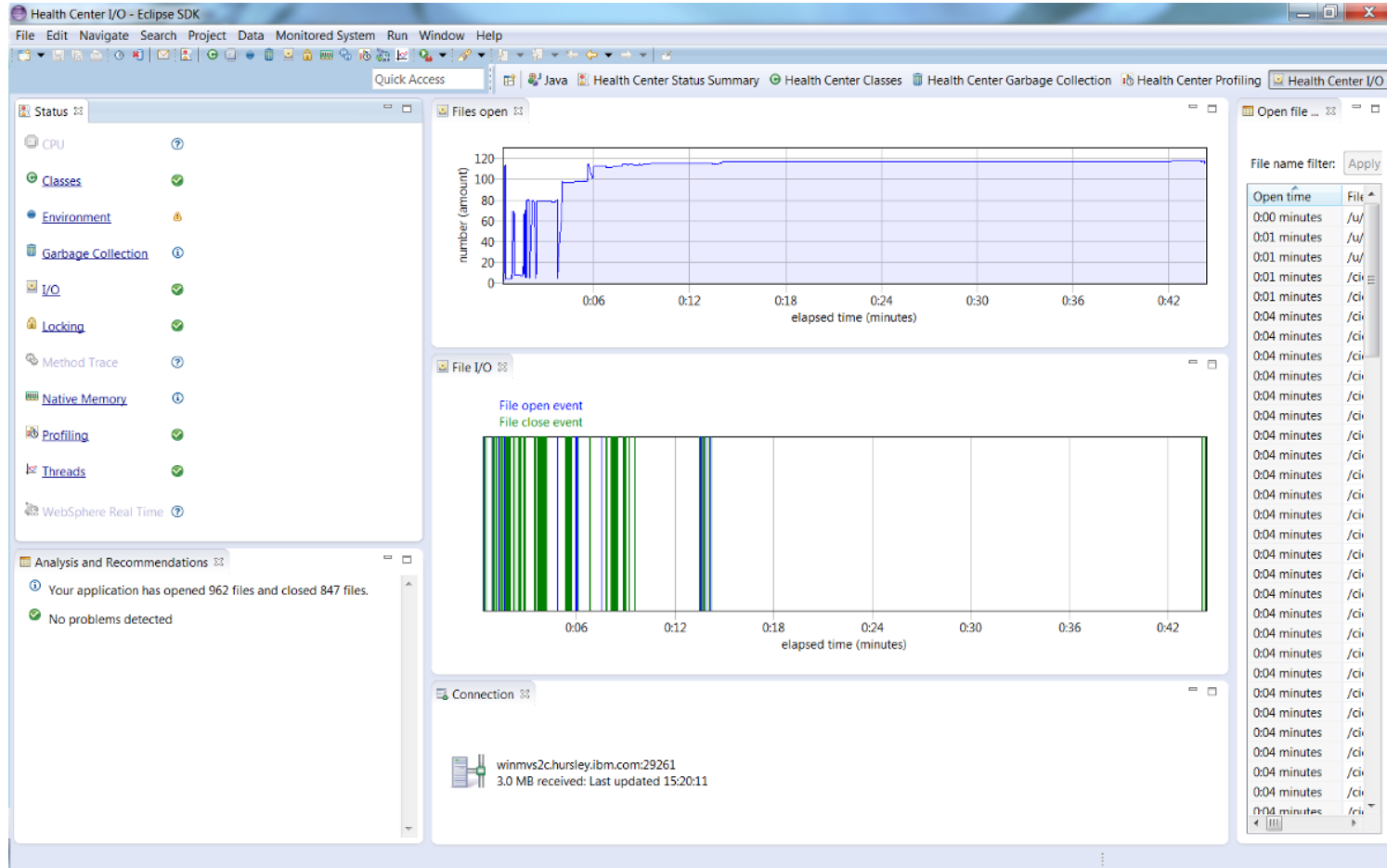
I CWWK00220I: TCP Channel defaultHttpEndpoint has stopped listening for requests on host * (IPv6) port 29540.

...

A CWWKE0036I: The server defaultServer stopped after 1 minutes, 42.677 seconds.
```

Complete your session evaluations online at [www.SHARE.org/Orlando-Eval](http://www.SHARE.org/Orlando-Eval)

# Monitoring Liberty

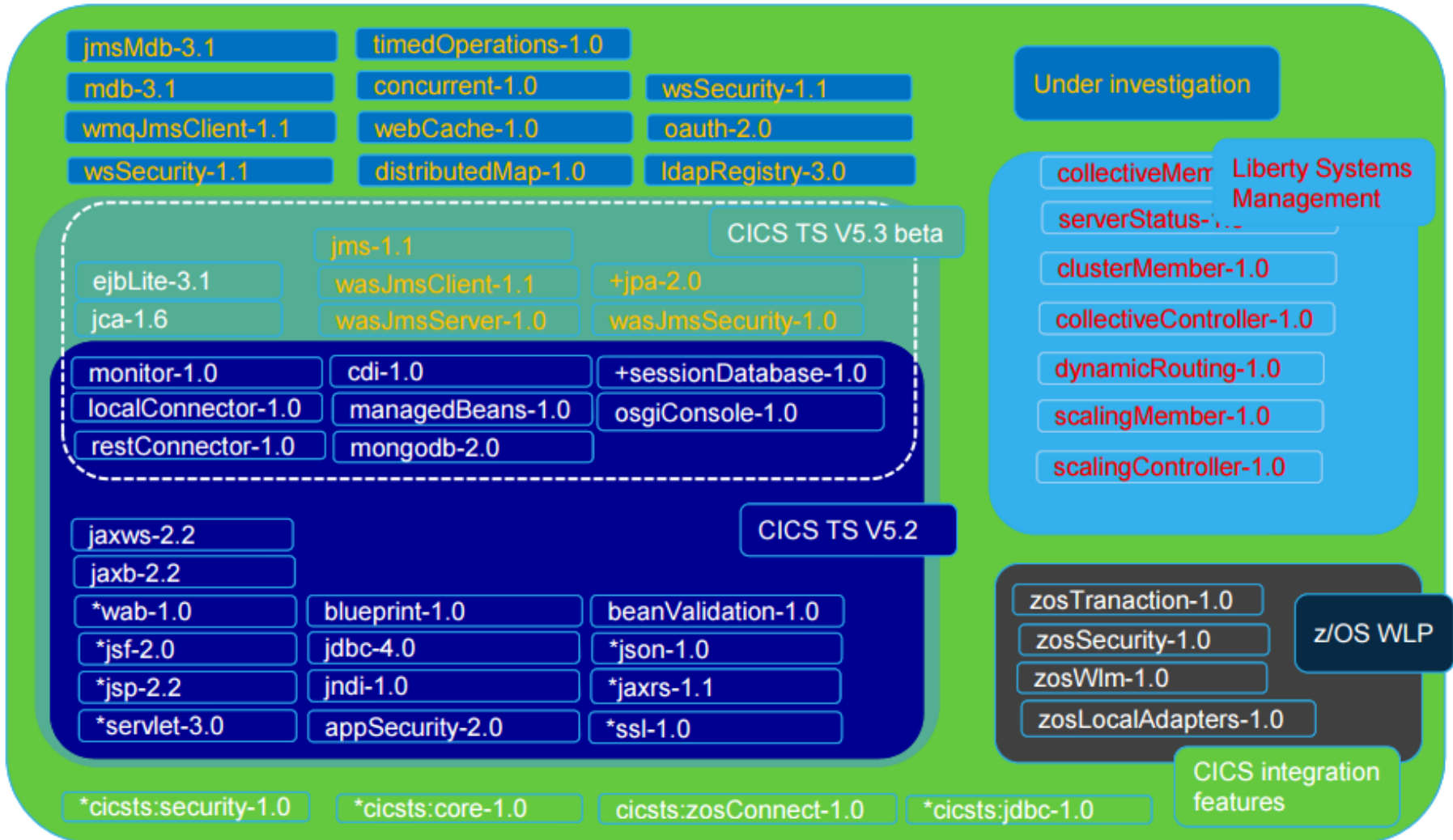


Complete your session evaluations online at [www.SHARE.org/Orlando-Eval](http://www.SHARE.org/Orlando-Eval)

# Areas to discuss

1. Lets get started!
2. What is this rich environment you speak of?

# CICS & Liberty – Available Features

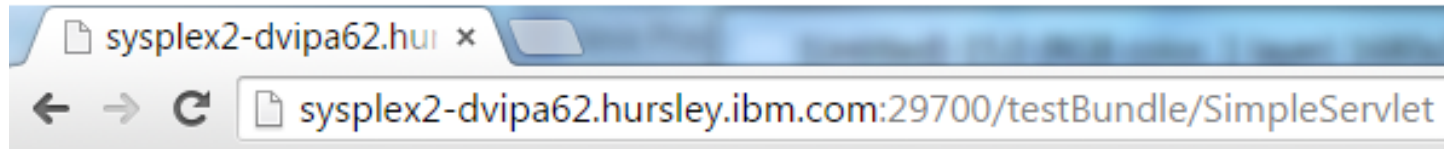


\* Originally available on CICS TS V5.1

+ Limited to T4 database drivers

# JCICS & Liberty

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
  
    // Set up the output stream  
    ServletOutputStream out = response.getOutputStream();  
  
    // Get the current CICS task  
    Task task = Task.getTask();  
  
    try{  
        // Print out information from CICS  
        out.println("Program name: " + task.getProgramName());  
        out.println("Transaction name: " + task.getTransactionName());  
        out.println("Task id: " + task.getTaskNumber());  
        out.println("User id: " + task.getUserID());  
    } catch(InvalidRequestException e){  
        throw new IOException(e);  
    }  
}
```



```
Program name:    DFHSJTHP  
Transaction name: C3SA  
Task id:        80  
User id:        CICSUSER
```

# JDBC Type 2 datasource

- Native database driver
- Uses the same interface to DB2 as a CICS COBOL application
- Same transaction support as CICS COBOL
- Can only connect to the same local DB2 instance as the CICS system

# Type 2 datasources

```
<cicsts_dataSource id="t2" jndiName="jdbc/cicsDataSource"/>
<cicsts_jdbcDriver id="db2t2Driver" libraryRef="db2Lib"/>
<library id="db2Lib">
  <fileset dir="/usr/lpp/db2v11/jdbc/classes" includes="db2jcc4.jar db2jcc_license_cisuz.jar"/>
  <fileset dir="/usr/lpp/db2v11/jdbc/lib"/>
</library>
```

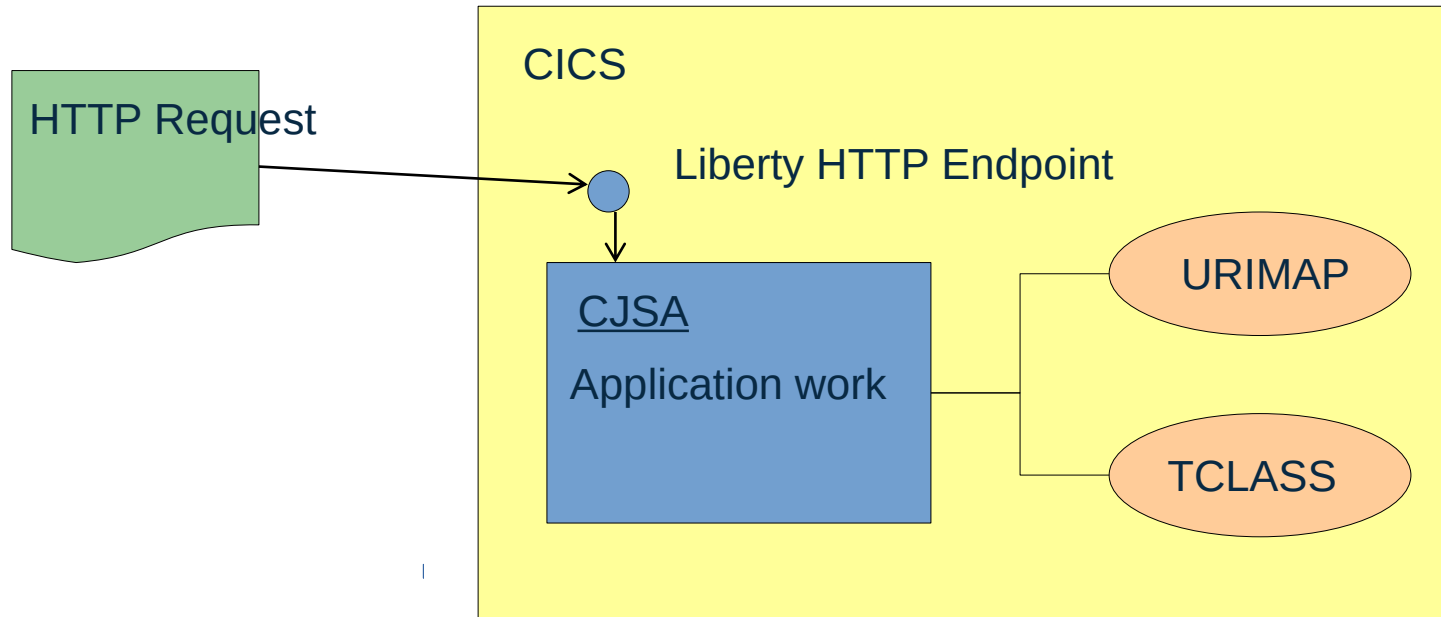
```
I DB2CON
STATUS: RESULTS - OVERTYPE TO MODIFY
Db2conn
Accountrec( None )
Authid(      )
Authtype( Userid )
Comauthid(      )
Comauthtype( Cuserid )
Comthreadlim( 0001 )
Comthreads(0000)
Connecterror( Sqlcode )
Connectst( Connected )
Db2groupid(      )
Db2id( DJ2C )
Db2release(1110)
Drollback(Rollback)
Msgqueue1( CDB2 )
Msgqueue2(      )
Msgqueue3(      )
+ Nontermrel( Release )
```



# Java Transaction API

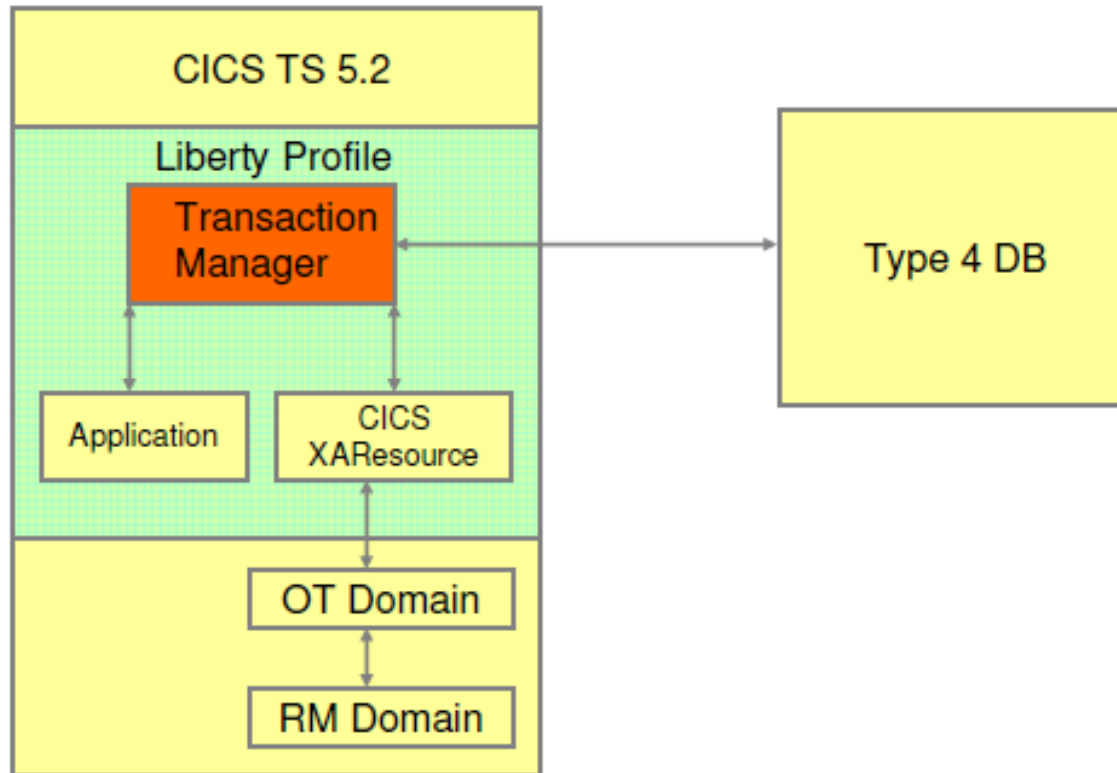
- Does NOT replace existing CICS transactional model
- Allows java to extend and co-ordinate transactions across multiple resources
  - Of which CICS is just one
- Transactional scope is explicitly defined
- JTA is not required for Type 2 datasources
- CICS transaction recovery and syncpoint is NOT affected

# Transaction Model



# User Transactions

The Java Transaction API allows programmers to define their own transactions, which can span multiple XA resources.



# User Transactions cont

```
// Get the user transaction from the context
InitialContext ctx = new InitialContext();
UserTransaction transaction = (UserTransaction) ctx.lookup("java:comp/UserTransaction");

// Begin the transaction
transaction.begin();

// Do your CICS work
TSQ targetTsqr = new TSQ();
targetTsqr.setName("RECORDS");
targetTsqr.writeString("01:FRANK:500A");

// Rollback or commit the changes
transaction.commit();
```

# JPA – Java Persistence API

Allows a programmer to map relational database entities to simple classes without needing to use SQL. It also allows programmers to extend CRUD operations.

```
@Entity
@Table(name = "STTESTER.JPA")
public class Employee implements Serializable {

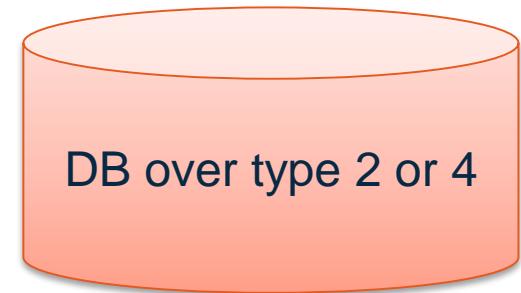
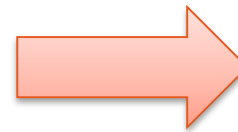
    @Id
    @Column(name = "EMPNO")
    private Long EMPNO;

    @Column(name = "NAME", length = 8)
    private String NAME;
    private static final long serialVersionUID = 1L;

    public Employee() {
        super();
    }
    public Long getEMPNO() {
        return this.EMPNO;
    }

    public void setEMPNO(Long EMPNO) {
        this.EMPNO = EMPNO;
    }
    public String getName() {
        return this.NAME;
    }

    public void setName(String NAME) {
        this.NAME = NAME;
    }
}
```



# JPA – Java Persistence API

```
<persistence-unit name="cics.test.wlp.jpa.annotation.cics.datasource">  
  <jta-data-source>defaultCICSDataSource</jta-data-source>  
</persistence-unit>
```

```
// Set up the entity manager for our operation  
EntityManager em = emf.createEntityManager();
```

```
UserTransaction transaction = (UserTransaction) ctx.lookup("java:comp/UserTransaction");
```

```
// Begin the transaction and join it to the entity manager  
transaction.begin();  
em.joinTransaction();
```

```
// Locate the employee in the database  
Employee changeEmp = em.find(Employee.class, id);
```

```
// Update the record with the desired name then persist the change  
changeEmp.setName("Alan");  
em.persist(changeEmp);  
em.flush();
```

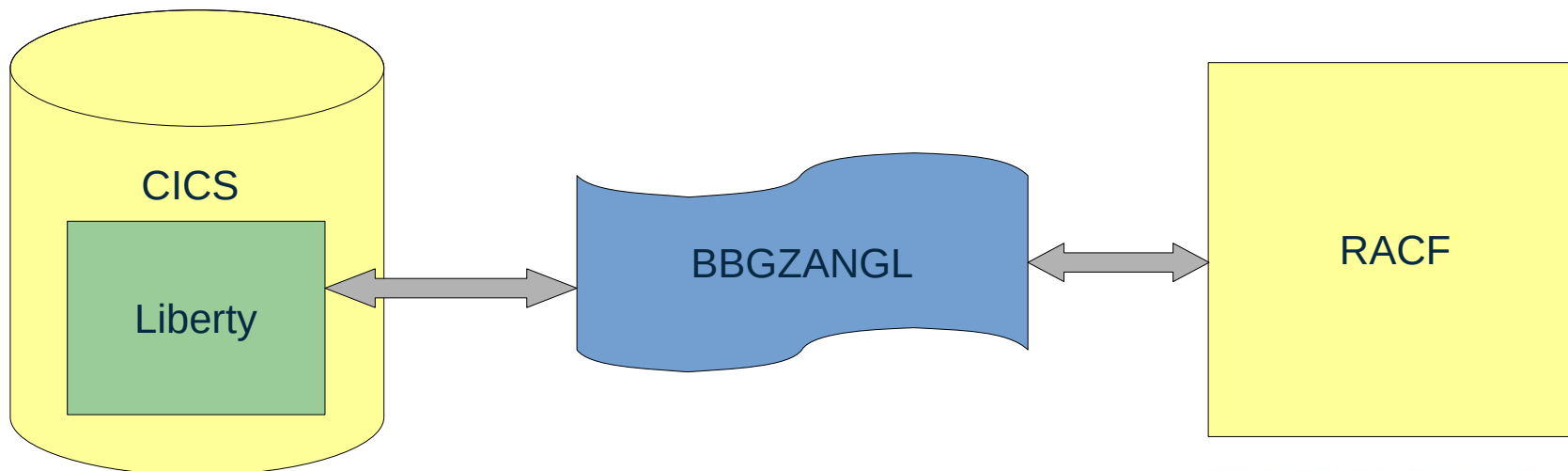
```
// Commit the change  
transaction.commit();
```

# Security (Angel Process)

```
Display Filter View Print Options Search Help
-----
SDSF STATUS DISPLAY ALL CLASSES
COMMAND INPUT ==>
PREFIX=BBGZANGL DEST=(ALL) OWNER=* SORT=JobID/A FIL
NP  JOBNAME  JobID  Owner  Prty  Queue  C  Pos
   BBGZANGL STC53245 WLPUSER  15  EXECUTION
   BBGZANGL STC63775 WLPUSER  15  EXECUTION
   BBGZANGL STC95617 WLPUSER  15  EXECUTION
```

JCL for the angel process is provided by CICS in your USS home directory:

*USSHOME/wlp/templates/zos/procs/bbgzangl.jcl*



# Security (Angel Process) cont.

```
<safRegistry id="saf" realm="Test Realm"/>
<ssl id="defaultSSLConfig" keyStoreRef="defaultKeyStore" sslProtocol="TLS"/>
<keyStore id="defaultKeyStore" password="defaultPassword"/>
<safCredentials authenticatedUser="CICSUSER" profilePrefix="APPLID"/>
```

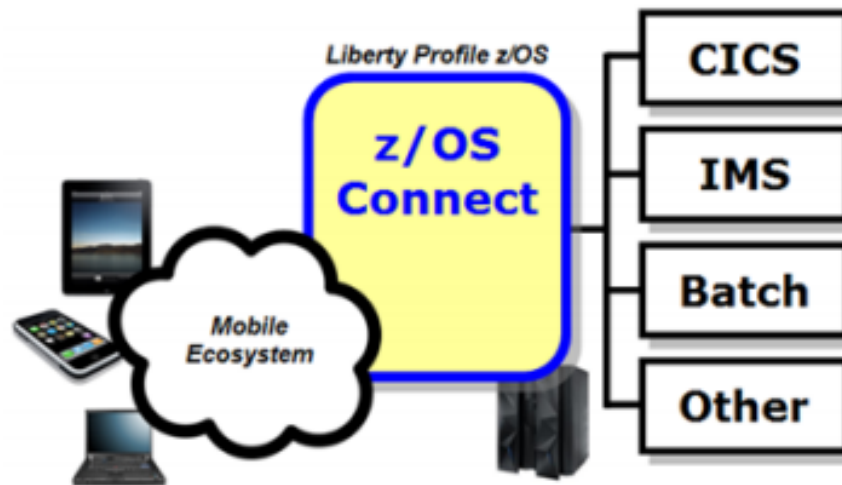
```
<security-constraint id="SecurityConstraint">
  <web-resource-collection id="WebResourceCollection">
    <web-resource-name>Protected with cicsAllAuthenticated role</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <auth-constraint id="AuthConstraint">
    <role-name>cicsAllAuthenticated</role-name>
  </auth-constraint>
</security-constraint>
```

```
<application-bnd>
  <security-role name="allAuthenticated">
    <special-subject type="ALL_AUTHENTICATED_USERS"/>
  </security-role>
</application-bnd>
```



# z/OS Connect in CICS

A unified mobile-friendly interface for z/OS resources and services



# z/OS Connect in CICS

```
<feature>cicsts:zosConnect-1.0</feature>
```

```
<com.ibm.cics.wlp.zosconnect.CICSEndpoint id="com.ibm.cics.wlp.zosconnect.CICSEndpointService"/>
```

```
{"zosConnectServices":[  
  {"ServiceName":"Test Service",  
   "ServiceDescription":"Simple service for testing.",  
   "ServiceProvider":"com.ibm.cics.wlp.zosconnect.CICSEndpointServiceImpl",  
   "ServiceURL":"https://winmvs2c.hursley.ibm.com:29461/zosConnect/services/Test Service"}}]
```

# Servlets

- Probably the first feature you would EVER use when trying Liberty
- Basic way of interacting with HTTP requests
  - Base for REST and SOA interactions
- Can be tested by simply pointing a browser at your deployment

# Web Servlets

```
@WebServlet("/Read")
public class Read extends HttpServlet {
    @PersistenceUnit(unitName = "cics.test.wlp.jpa.annotation.cics.datasource")
    EntityManagerFactory emf;

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        // Get the customer number from the servlet parameters
        long id = Long.valueOf(request.getParameter("id"));

        // Retrieve the employee object
        EntityManager em = emf.createEntityManager();
        Employee myEmployee = em.find(Employee.class, id);

        // Print out the retrieved information
        ServletOutputStream out = response.getOutputStream();
        out.println("== SUCCESS ==");
        out.println("Found employee with id: " + myEmployee.getEMPNO());
        out.println("Associated name is:      " + myEmployee.getNAME());
    }
}
```

# JAX-WS & JAX-RS

- JAX - ...
  - WS – Web Services
  - RS – Rest Services
- Standard java framework for writing and consuming these services

## JAX-RS

```
{  
  "JSON": "REST"  
}
```

## JAX-WS

```
<SOAP><xml /></SOAP>
```

# JAX-WS

```
@Path("/account")
public class AccountsResource {

    @GET
    @Path("/{accountNumber}")
    @Produces("application/json")
    public Response getAccount(@PathParam("accountNumber") Long accountNumber) {

        /*snip*/

        /*Create the JSON response*/
        JSONObject response = new JSONObject();
        response.put("sortCode", responseAccountData.getScode());
        response.put("accountNumber", responseAccountData.getAccno());

        return Response.status(200)
            .entity(response.toString())
            .build();
    }
}
```

# Summary

- Liberté, Égalité, Fraternité
- Liberty
  - An opt – in lightweight java application server
- Equality
  - Equal features in distributed or on the mainframe
- Fraternity
  - Extend, support and modernise trusted applications



*Liberté • Égalité • Fraternité*

**RÉPUBLIQUE FRANÇAISE**