WebSphere Application Server: Liberty Profile – Rumors Dispelled

David Follis
IBM

August 8, 2012
Session Number 11374

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

CICS*
DB2*
GDPS*
Geographically Dispersed Parallel Sysplex
HiperSockets
IBM*
IBM eServer
IBM logo*
IMS
On Demand Business logo
Parallel Sysplex*
RACF*
System z9
WebSphere*
z/OS
zSeries*

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon 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 throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Disclaimer

- The information contained in this documentation is provided for informational purposes only. While efforts were many to verify the completeness and accuracy of the information contained in this document, it is provided “as is” without warranty of any kind, express or implied.
- This information is based on IBM’s current product plans and strategy, which are subject to change without notice. IBM will not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation.
- Nothing contained in this documentation 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 the IBM software.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon 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 throughput improvements equivalent to the performance ratios stated here.
- All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

WebSphere Application Server on z/OS

Session	Day	Time	Room	Title	Speaker
11377	Monday	11:00	Grand Ballroom Salon B	What Can I Do With the SMF 120s?	David Follis
11371	Monday	3:00	Orange County Salon 2/3	Administrator Hands On Lab	David Follis / Michael Stephen / Ken Irwin
11378	Tuesday	12:15	Grand Ballroom Salon B	Spelunking the Admin Console	John Hutchinson
11375	Tuesday	4:30	Grand Ballroom Salon B	Being the Back-Up Administrator	Mike Loos
11374	Wednesday	11:00	Grand Ballroom Salon B	Liberty Profile – Rumors Dispelled	David Follis
11373	Thursday	4:30	Grand Ballroom Salon B	What's New?	David Follis / John Hutchinson / Michael Stephen
			Salon B		
11376	Friday	8:00	Platinum Ballrom Salon 10	zWAS – In Real Life	David Follis / Rod Feak

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?

An easy to configure runtime environment

- Simple, extensible, and sparse configuration model
  - Configuration can live in a single XML document
  - Configuration is by exception
    - Defaults are provided by contributing feature
    - Only modifications to the defaults are required
- Flexible configuration structure
  - Include mechanism allows for shared configuration elements
  - Variable indirection mechanism allows for customization when distributed across multiple JVMs
  - Easily managed by version control systems if desired

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

Lightweight configuration

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


    <logging consoleLogLevel="INFO" />



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




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


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

Features control what’s available in the runtime

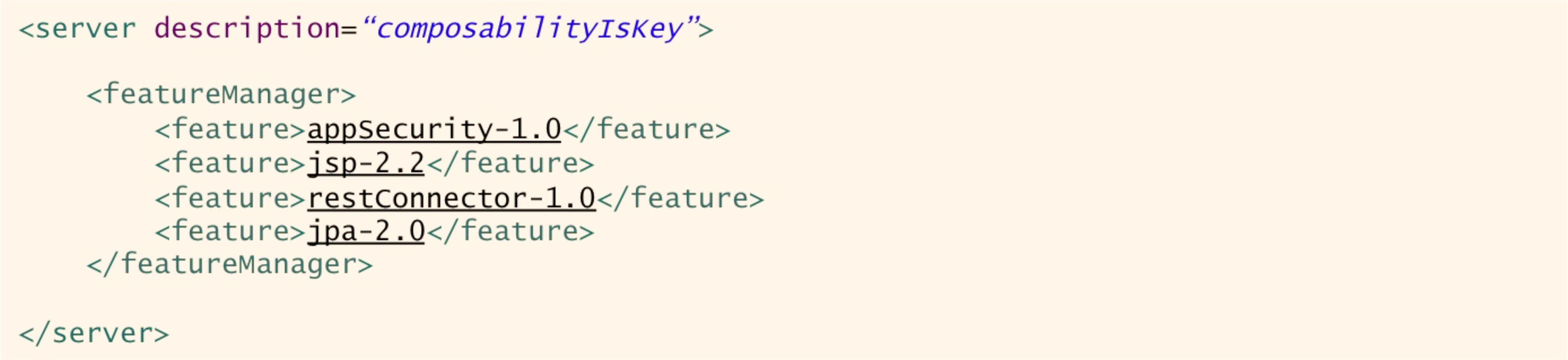
Singleton configurations specify properties for runtime services for which there is only one instance

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

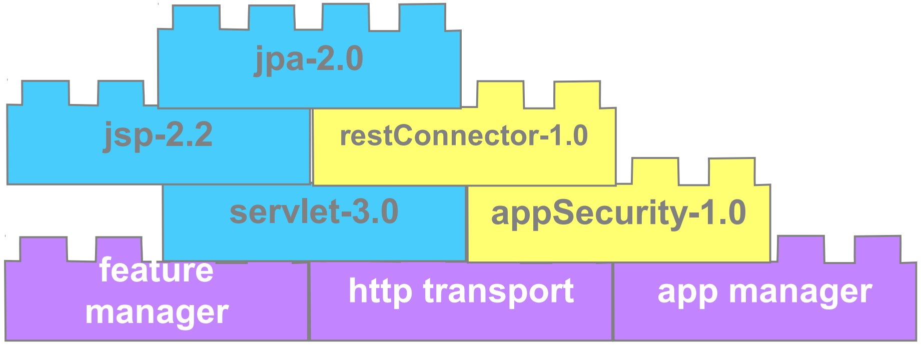
Includes can be used to implement an extensible configuration model

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

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>
```

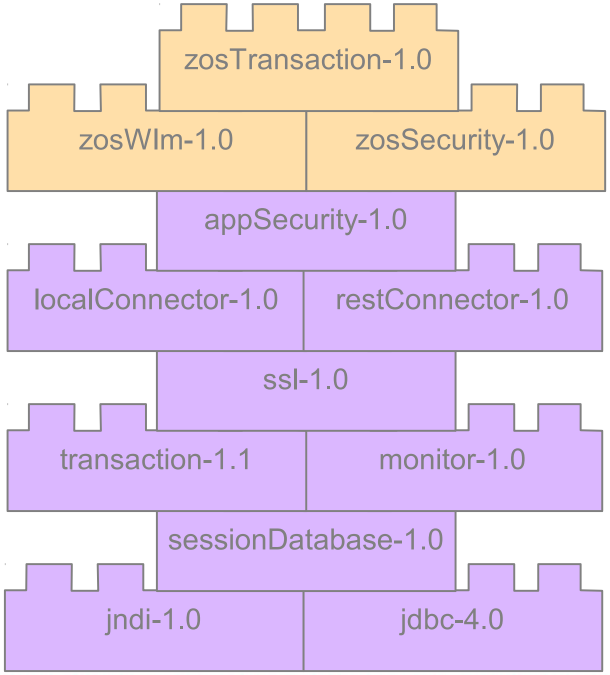


What is the WAS for z/OS Liberty profile?

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



zosTransaction-1.0

zosWlm-1.0

zosSecurity-1.0

appSecurity-1.0

localConnector-1.0

restConnector-1.0

ssl-1.0

transaction-1.1

monitor-1.0

sessionDatabase-1.0

jndi-1.0

jdbc-4.0

z/OS Feature Sets

Common Feature Sets

Liberty and traditional profile capabilities

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 …

Angel – Enabling *authorized* services

- Many z/OS services require callers to be *authorized*
  - Typically documented as “in a system key or supervisor state”
  - These services, when abused, have side effects that could impact the stability or integrity of the system so the system requires callers to have extra privileges
- Exploiting most z/OS features requires authorized code
  - Workload management
  - Transaction management
  - SAF (security) interface exploitation
  - Cross-memory communications
- The *Angel* enables unauthorized Liberty profile servers to access these authorized services

Angel – Details

- The Angel is **not** the same as the WAS for z/OS daemon process
  - No communication end point is hosted by the process
  - No ties to the WebSphere topology (cells, nodes)

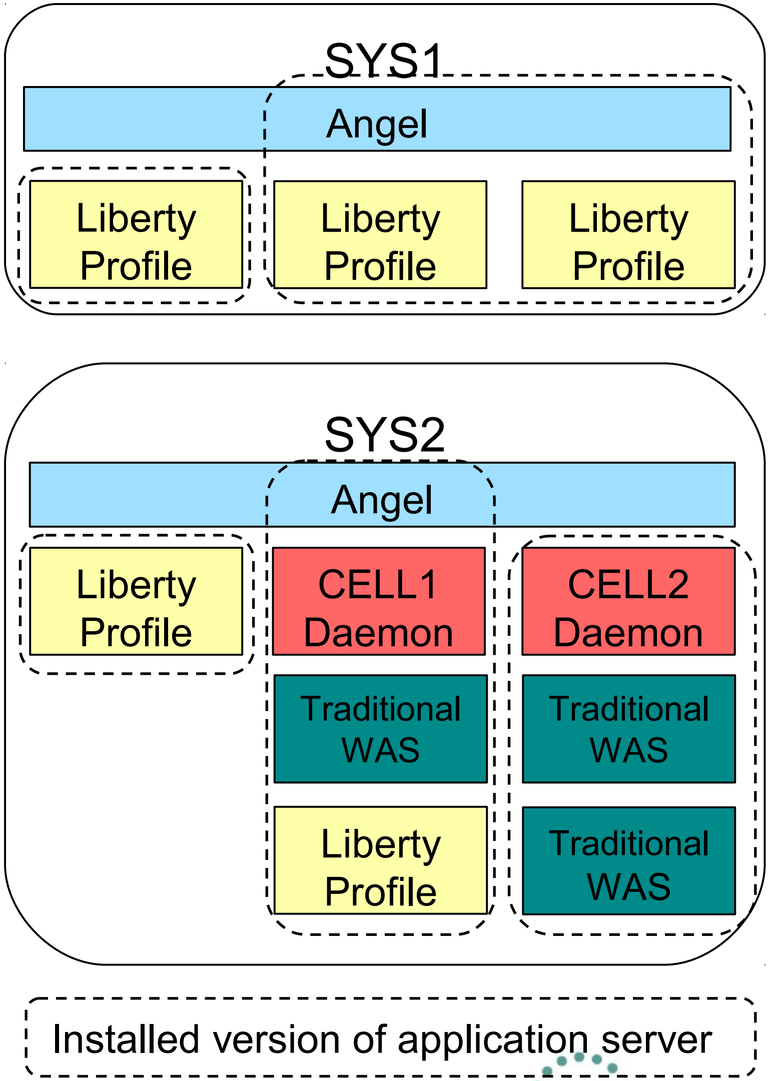
- The Angel is an **optional** process
  - Provides a system LX that enables Liberty JVMs to bootstrap and wire up PC routines
  - Only needed if Liberty JVMs need to run system authorized code
  - Provides fine grained access controls around authorized services

- The Angel does not execute code except in response to operator commands

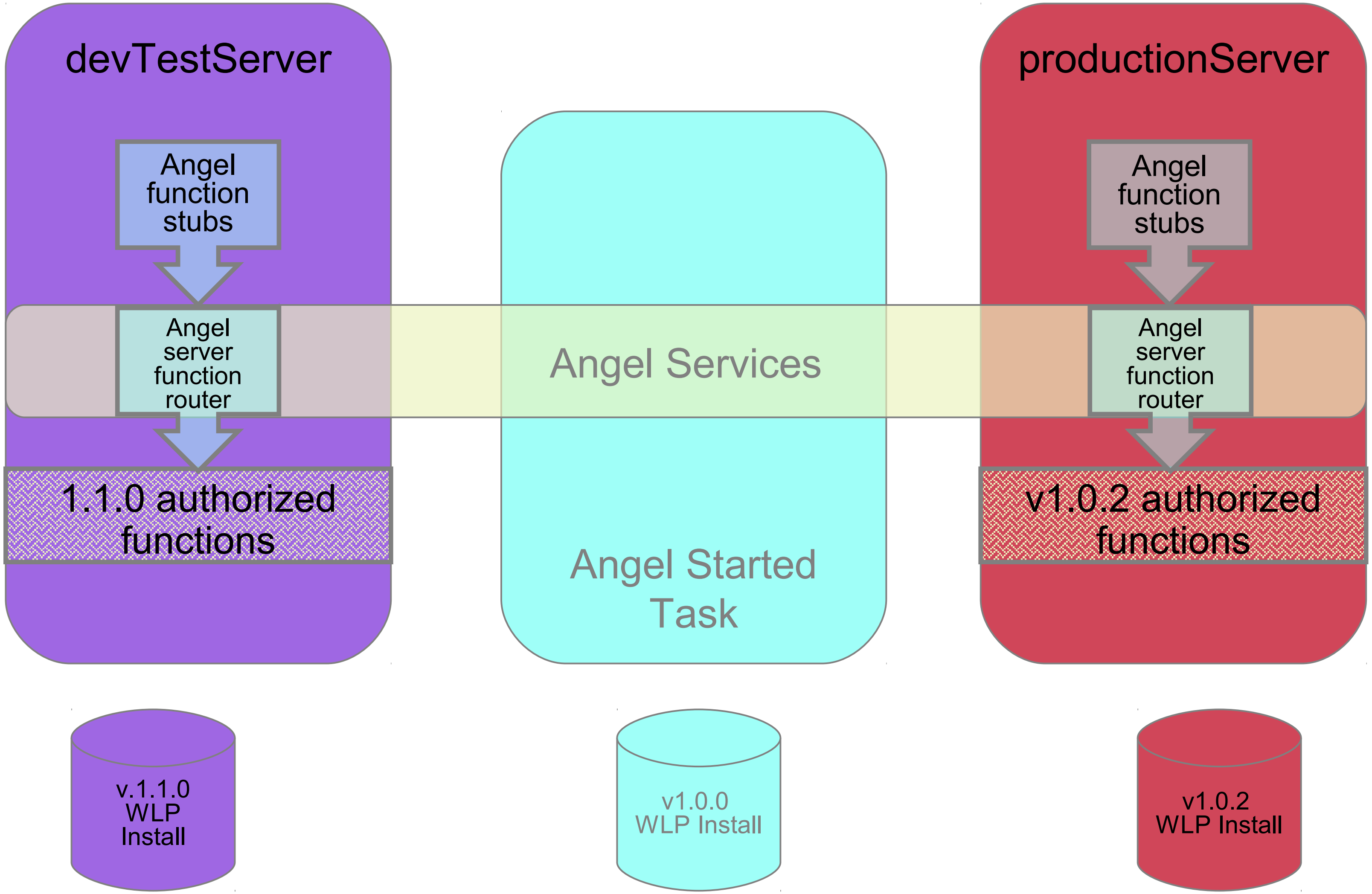
- The Angel is structured to allow service without restart
  - MODIFY RELOAD will load a new version of code

WebSphere for z/OS – Processes

- Angel
  - Extremely light-weight started task
  - Single instance per system image *regardless of WAS topology*
  - No configuration to manage
  - No code level dependency between angel and server
- Liberty profile server
  - Single process implementation
  - Uses agent infrastructure to exploit authorized z/OS services
- Traditional WAS
  - Split process
  - Requires daemon infrastructure



Angel – Version agnostic



Angel access control – Examples

Allow a user to access the angel

```
RDEF SERVER BBG.ANGEL UACC(NONE)
PE BBG.ANGEL CLASS(SERVER) ACCESS(READ) ID(USERID)
```

Allow a user to load the server authorized function module

```
RDEF SERVER BBG.AUTHMOD.BBGZSAFM UACC(NONE)
PE BBG.AUTHMOD.BBGZSAFM CLASS(SERVER) ACCESS(READ) ID(USERID)
```

Allow a user to access RRS in support of local data access

```
RDEF SERVER BBG.AUTHMOD.BBGZSAFM.TXRRS UACC(NONE)
PE BBG.AUTHMOD.BBGZSAFM.TXRRS CLASS(SERVER) ACCESS(READ) ID(USERID)
```

Allow a user to access workload management services:

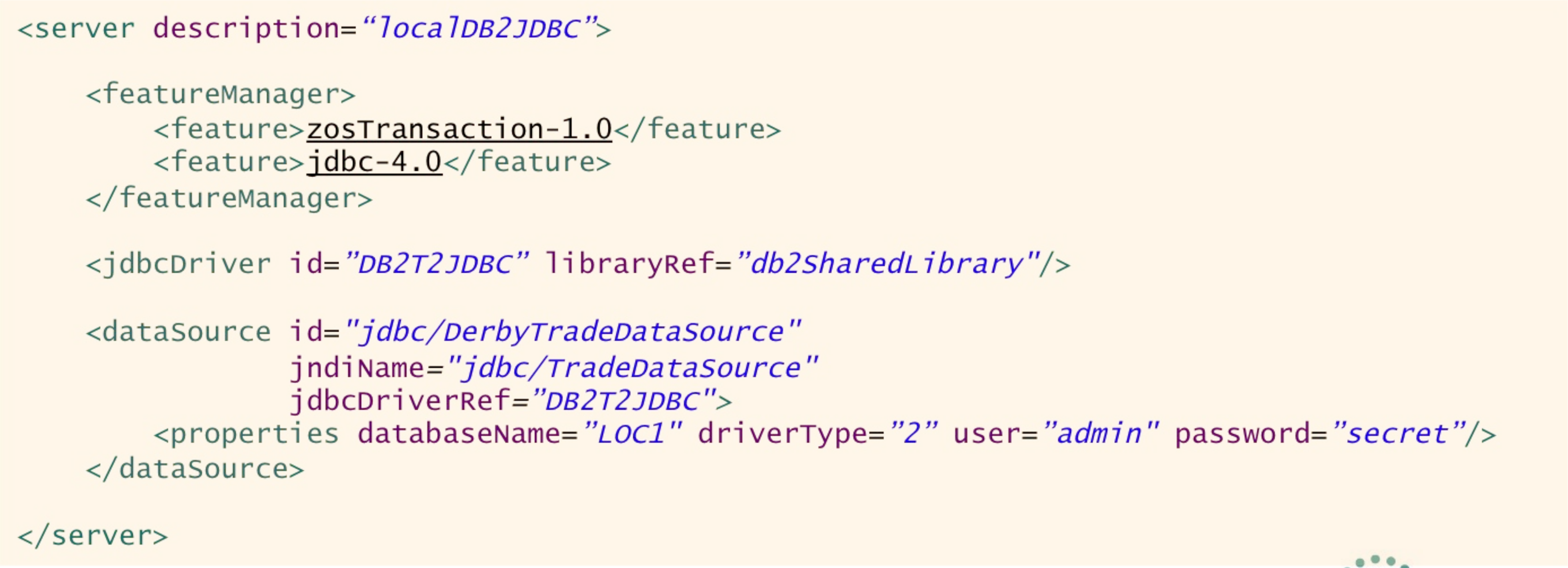
```
RDEF SERVER BBG.AUTHMOD.BBGZSAFM.ZOSWLM UACC(NONE)
PE BBG.AUTHMOD.BBGZSAFM.ZOSWLM CLASS(SERVER) ACCESS(READ) ID(USERID)
```

Allow a user to use native credential management services:

```
RDEF SERVER BBG.AUTHMOD.BBGZSAFM.SAFCRED UACC(NONE)
PE BBG.AUTHMOD.BBGZSAFM.SAFCRED CLASS(SERVER) ACCESS(READ) ID(USERID)
```

Feature – z/OS Transactions

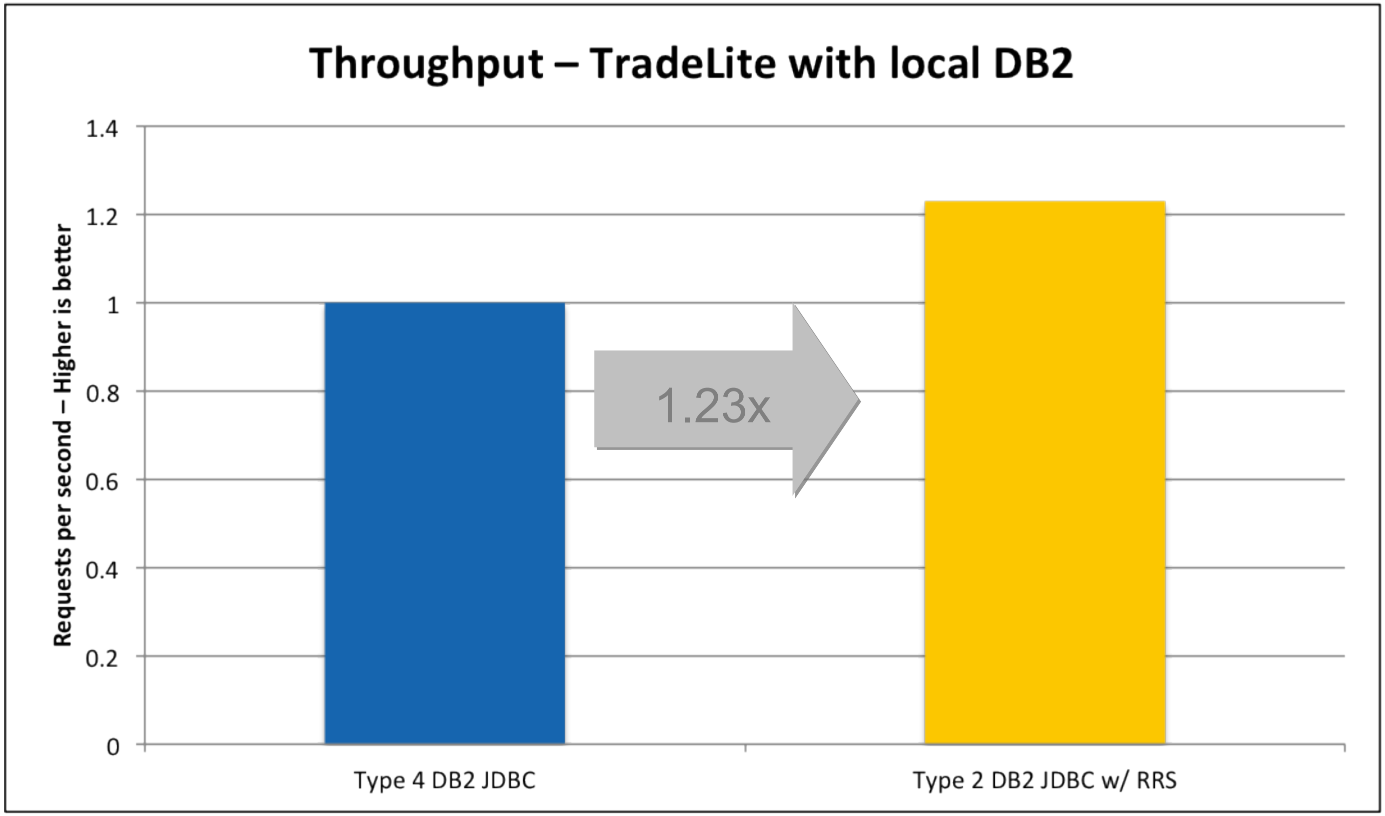
- Extends the WebSphere transaction manager
  - Provides native transaction context management via MVS context services and resource recovery services (RRS)
  - Implements 2PC across JTA/XA resource managers and RRS enabled resource managers
- Required to support Local DB2 connectivity via JDBC



```
<server description=“localDB2JDBC”>

    <featureManager>
        <feature>zosTransaction-1.0</feature>
        <feature>jdbc-4.0</feature>
    </featureManager>

    <jdbcDriver id=”DB2T2JDBC” libraryRef=”db2SharedLibrary"/>

    <dataSource id="jdbc/DerbyTradeDataSource"
                jndiName="jdbc/TradeDataSource"
                jdbcDriverRef=”DB2T2JDBC">
        <properties databaseName=”LOC1" driverType=”2” user=”admin" password=”secret”/>
    </dataSource>

</server>
```

Feature – z/OS Transactions: Performance

Optimized local connectivity for higher throughput

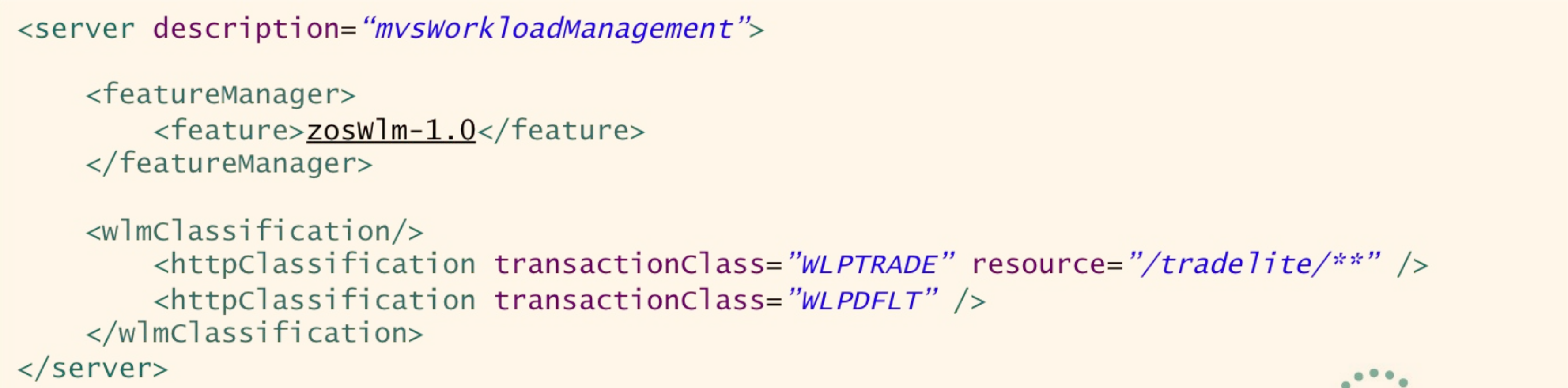


- z196, 4-way LPAR running z/OS 1.13
- 64bit IBM Java 6.0.1 with compressed references, 1M large pages, 2GB heap
- IBM DB2 for z/OS v10, JDBC with keepDynamic

Feature – z/OS Workload Manager

Adds support to classify HTTP requests with z/OS WLM

- Classification associates response time goals and importance to work run in WebSphere
- z/OS workload manager will manage the resources available on the system in a way that ensures the most important work runs while attempting to meet response time goals
- RMF reports provide information about completed transactions, response times, etc by service class



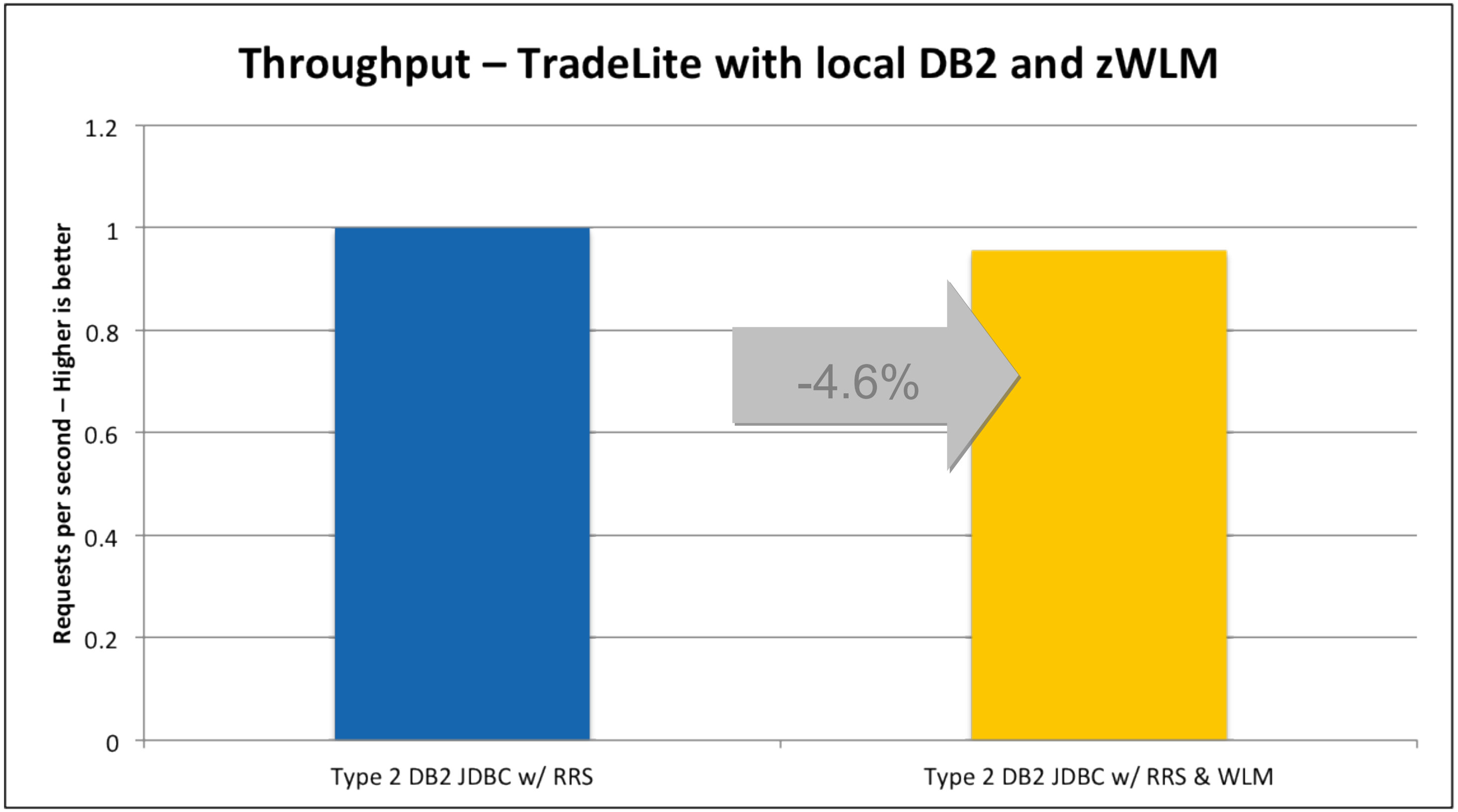
```
<server description="mvsWorkloadManagement">

    <featureManager>
        <feature>zosWlm-1.0</feature>
    </featureManager>

    <wlmClassification/>
        <httpClassification transactionClass="WLPTRADE" resource="/tradelite/**" />
        <httpClassification transactionClass="WLPDFLT" />
    </wlmClassification>
</server>
```

RMF Report with WLM – Example

```
REPORT BY: POLICY=STANDARD   WORKLOAD=NEWWORK    SERVICE CLASS=WASCLASS   RESOURCE GROUP=*NONE      PERIOD=1 IMPORTANCE=2
                                                 CRITICAL     =NONE

-TRANSACTIONS-  TRANS-TIME HHH.MM.SS.TTT  --DASD I/O--  ---SERVICE---   SERVICE TIME  ---APPL %---  --PROMOTED--  ----STORAGE----
AVG      17.70  ACTUAL                 6  SSCHRT   0.0  IOC         0   CPU 1022.793  CP    341.49  BLK    0.000  AVG         0.00
MPL      17.70  EXECUTION              6  RESP     0.0  CPU    57020K   SRB    0.000  AAPCP   0.00  ENQ    0.000  TOTAL       0.00
ENDED   832303  QUEUED                 0  CONN     0.0  MSO         0   RCT    0.000  IIPCP   0.00  CRM    0.000  SHARED      0.00
END/S  2778.86   R/S AFFIN             0   DISC     0.0  SRB         0   IIT    0.000                LCK    0.000
#SWAPS       0  INELIGIBLE             0  Q+PEND   0.0  TOT    57020K   HST    0.000  AAP      N/A                -PAGE-IN RATES-
EXCTD        0  CONVERSION             0  IOSQ     0.0  /SEC   190376   AAP      N/A  IIP      N/A                SINGLE       0.0
AVG ENC  17.70  STD DEV               21                                IIP      N/A                              BLOCK        0.0
REM ENC   0.00                                          ABSRPTN   11K                                             SHARED       0.0
MS ENC    0.00                                          TRX SERV  11K                                             HSP          0.0

GOAL: RESPONSE TIME 000.00.00.250 FOR  80%

         RESPONSE TIME EX   PERF  AVG    --EXEC USING%--  -------------- EXEC DELAYS % -----------  -USING%-  --- DELAY % ---    %
SYSTEM      ACTUAL%     VEL% INDX ADRSP  CPU AAP IIP I/O  TOT CPU                                   CRY CNT  UNK IDL CRY CNT  QUI

SP5           100      55.1  0.5  17.8    14 N/A N/A 0.0   11  11                                   0.0 0.0   75 0.0 0.0 0.0  0.0

                                      ----------RESPONSE TIME DISTRIBUTION----------
   ----TIME----     --NUMBER OF TRANSACTIONS--     -------PERCENT-------  0    10   20   30   40   50   60   70   80   90   100
   HH.MM.SS.TTT     CUM TOTAL          IN BUCKET     CUM TOTAL    IN BUCKET  |....|....|....|....|....|....|....|....|....|....|
<  00.00.00.125         830K            830K          100          100   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
<= 00.00.00.150         830K             560          100          0.1   >
<= 00.00.00.175         831K             377          100          0.0   >
<= 00.00.00.200         831K             311          100          0.0   >
<= 00.00.00.225         831K             223          100          0.0   >
<= 00.00.00.250         831K             198          100          0.0   >
<= 00.00.00.275         831K             162          100          0.0   >
<= 00.00.00.300         832K             113          100          0.0   >
<= 00.00.00.325         832K             108          100          0.0   >
<= 00.00.00.350         832K              85          100          0.0   >
<= 00.00.00.375         832K              76          100          0.0   >
<= 00.00.00.500         832K             210          100          0.0   >
<= 00.00.01.000         832K             215          100          0.0   >
>  00.00.01.000         832K              46          100          0.0   >
```

Feature – z/OS Workload Manager

The impact of enabling zWLM is under 5%



- z196, 4-way LPAR running z/OS 1.13
- 64bit IBM Java 6.0.1 with compressed references, 1M large pages, 2GB heap
- IBM DB2 for z/OS v10, T2 JDBC with keepDynamic

Feature – z/OS Security: Authentication

- The z/OS Security feature provides two implementations of a user registry that perform authentication with z/OS SAF interfaces
- The implementation that is used is based the presence of an Angel and the server's authorization to use the SAFCRED functions

Authorized

- Requires an active angel and appropriate access to Liberty `SAFCRED` authorized functions
- Uses the SAF `IRRSIA00` callable service
- Enables creation of native credentials required for SAF authorization

Unauthorized

- Requires the server to run in an environment that satisfies the `BPX.DAEMON` requirements
- Uses the LE / USS `__passwd_applid` implementation
- Unable to create native credentials required for SAF authorization

Feature – z/OS Security: Authorization

- SAF Authorization in Liberty allows a server to use the z/OS security product for access control checks
- Whenever a Subject tries to access a protected resource or requires access to an application role, the authorization check is rendered as a SAF check against a profile in the `EJBROLE` class
  - Class descriptor table entry allows for mixed case profile names
  - Maximum length of a profile is 246 characters
  - Rules to map role names to profile names can be configured
- Authorized credential services are required for SAF authorization
  - RACO / ACEE are passed to the SAF FASTAUTH service to perform access check

Feature – z/OS Security: Sandbox

- Extra access controls are provided with Liberty to prevent misuse of the SAF security interfaces. The user ID associated with a server process must be allowed to use the profile prefix

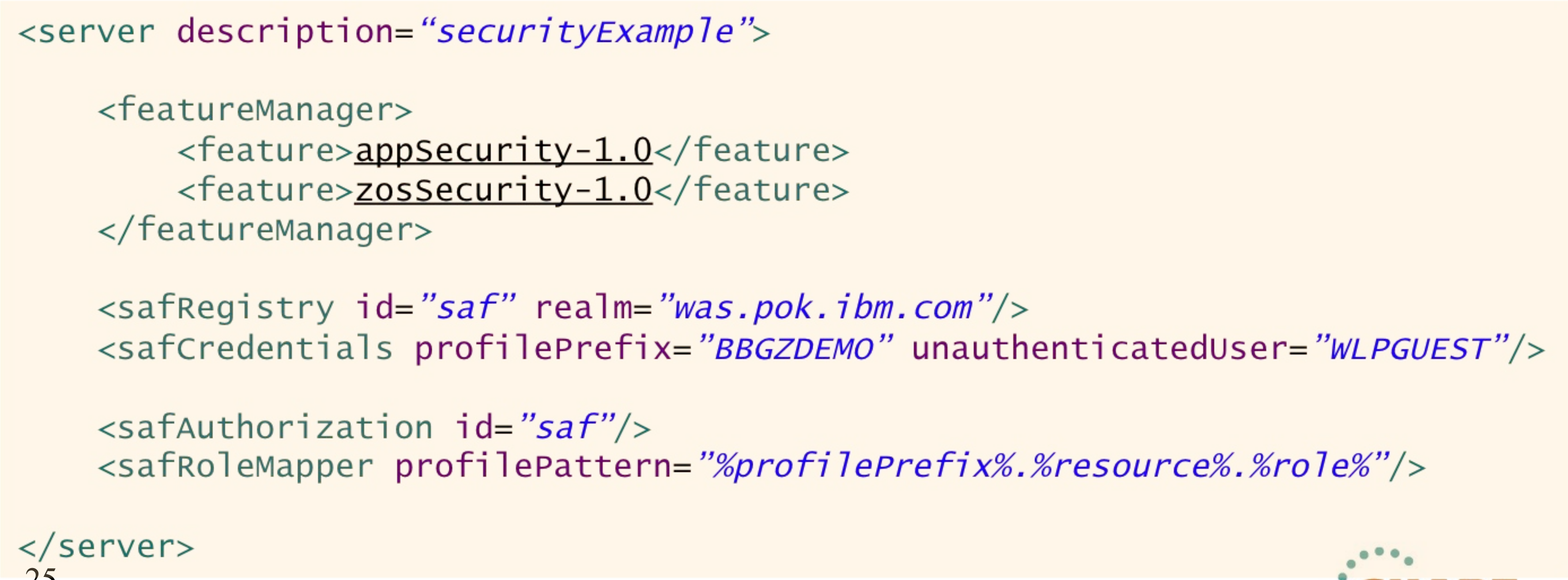
```
RDEF SERVER BBG.SECPFX.BBGZDFLT UACC(NONE)
PE BBG.SECPFX.BBGZDFLT CLASS(SERVER) ACCESS(READ)
   ID(USERID)
```

- The calls to services that generate native credentials are provided with an “application ID” based on the “profile prefix”
  - The APPL class can be used to prevent credential creation
- All user IDs associated with a server process must have SECPFX access to the first qualifier of EJBROLE
  - Prevents users from scanning authorization rules for access
- Provides the infrastructure necessary to enable security integration in a mixed workload environment

z/OS Security – Example

Full SAF exploitation

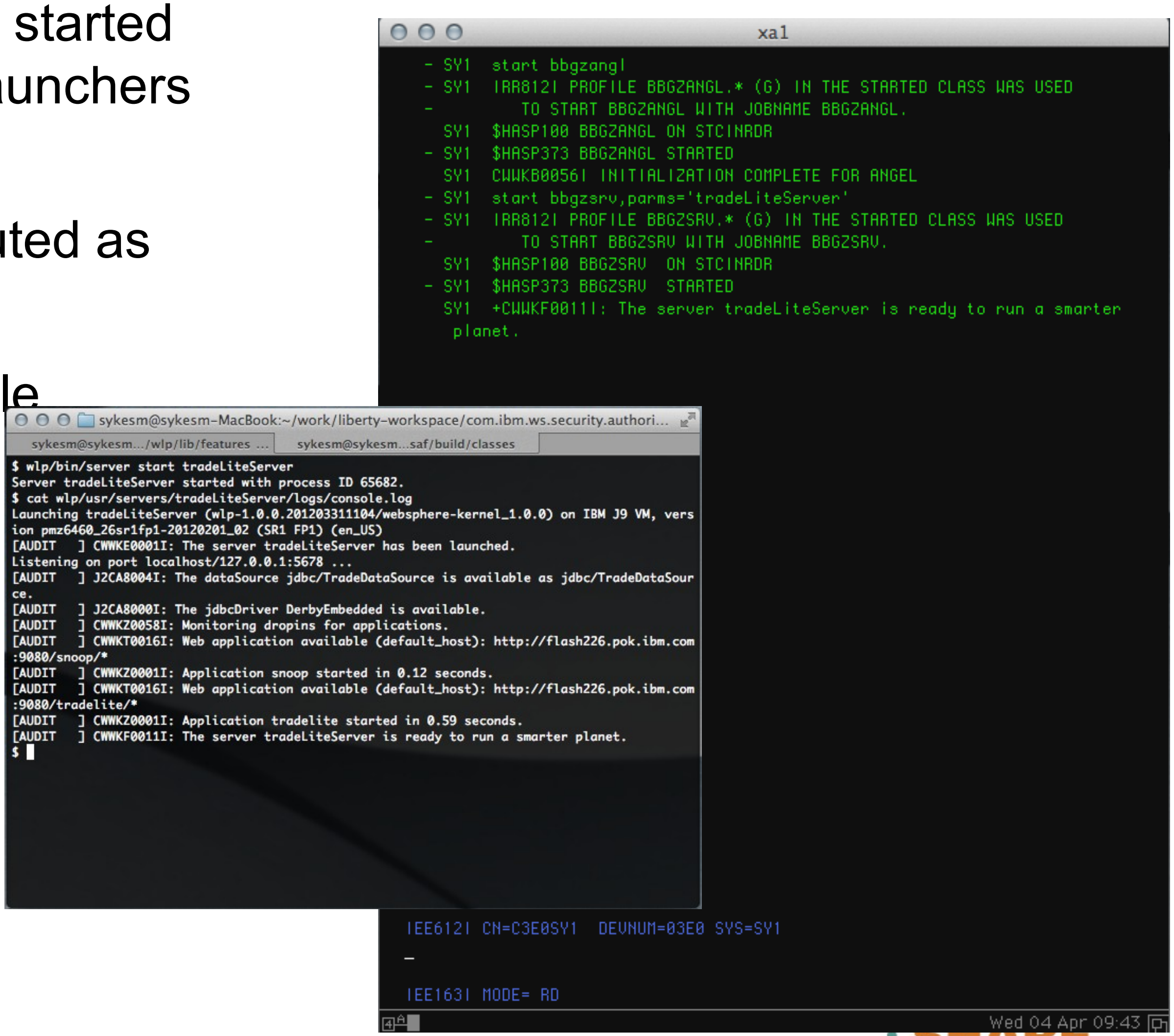
- Authentication performed with the local z/OS security product
- Credentials only created for users with access to the “BBGZDEMO” application ID
- The local z/OS security product is used for authorization



```
<server description=“securityExample”>

    <featureManager>
        <feature>appSecurity-1.0</feature>
        <feature>zosSecurity-1.0</feature>
    </featureManager>

    <safRegistry id=”saf” realm=”was.pok.ibm.com”/>
    <safCredentials profilePrefix=”BBGZDEMO” unauthenticatedUser=”WLPGUEST”/>

    <safAuthorization id=”saf”/>
    <safRoleMapper profilePattern=”%profilePrefix%.%resource%.%role%”/>

</server>
```

z/OS Operations – Choose your interface

- Run from a shell or as a started task with the provided launchers and PROCs
- Important messages routed as WTOs for automation
- Modify commands enable changes to trace specification or to request a diagnostic dump



```
 - SY1  start bbgzangl
 - SY1  IRR812I PROFILE BBGZANGL.* (G) IN THE STARTED CLASS WAS USED
 -         TO START BBGZANGL WITH JOBNAME BBGZANGL.
   SY1  $HASP100 BBGZANGL ON STCINRDR
 - SY1  $HASP373 BBGZANGL STARTED
   SY1  CWWKB0056I INITIALIZATION COMPLETE FOR ANGEL
 - SY1  start bbgzsrv,parms='tradeLiteServer'
 - SY1  IRR812I PROFILE BBGZSRV.* (G) IN THE STARTED CLASS WAS USED
 -         TO START BBGZSRV WITH JOBNAME BBGZSRV.
   SY1  $HASP100 BBGZSRV  ON STCINRDR
 - SY1  $HASP373 BBGZSRV  STARTED
   SY1  +CWWKF0011I: The server tradeLiteServer is ready to run a smarter
    planet.
```

```
 IEE612I CN=C3E0SY1  DEVNUM=03E0 SYS=SY1
 _
 IEE163I MODE= RD
```

sykesm@sykesm-MacBook:~/work/liberty-workspace/com.ibm.ws.security.authori...
sykesm@sykesm.../wlp/lib/features ...
sykesm@sykesm...saf/build/classes

```
$ wlp/bin/server start tradeLiteServer
Server tradeLiteServer started with process ID 65682.
$ cat wlp/usr/servers/tradeLiteServer/logs/console.log
Launching tradeLiteServer (wlp-1.0.0.201203311104/websphere-kernel_1.0.0) on IBM J9 VM, vers
ion pmz6460_26sr1fp1-20120201_02 (SR1 FP1) (en_US)
[AUDIT   ] CWWKE0001I: The server tradeLiteServer has been launched.
Listening on port localhost/127.0.0.1:5678 ...
[AUDIT   ] J2CA8004I: The dataSource jdbc/TradeDataSource is available as jdbc/TradeDataSour
ce.
[AUDIT   ] J2CA8000I: The jdbcDriver DerbyEmbedded is available.
[AUDIT   ] CWWKZ0058I: Monitoring dropins for applications.
[AUDIT   ] CWWKT0016I: Web application available (default_host): http://flash226.pok.ibm.com
:9080/snoop/*
[AUDIT   ] CWWKZ0001I: Application snoop started in 0.12 seconds.
[AUDIT   ] CWWKT0016I: Web application available (default_host): http://flash226.pok.ibm.com
:9080/tradelite/*
[AUDIT   ] CWWKZ0001I: Application tradelite started in 0.59 seconds.
[AUDIT   ] CWWKF0011I: The server tradeLiteServer is ready to run a smarter planet.
$ █
```

Why Liberty on z/OS?

Simplification

- Liberty environments don’t need significant z/OS configuration and customization
  - RRS, WLM, and SAF exploitation and configuration is optional
  - No authorized code is ***required*** to host applications
- Liberty runs in a single process instead of 3+ started tasks
  - Significantly reduced resource consumption
  - No started task definitions are ***required***
  - No need to create users and groups for controllers, servants
- Server instances can be quickly created or cloned

```
server create serverName [options]
server package serverName [options]
```

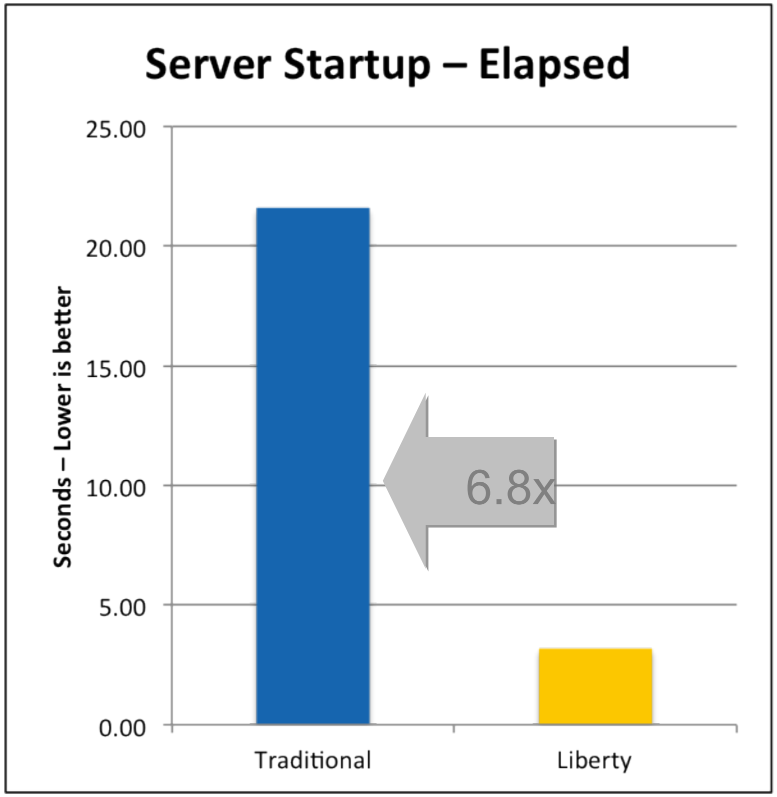
Why Liberty on z/OS?

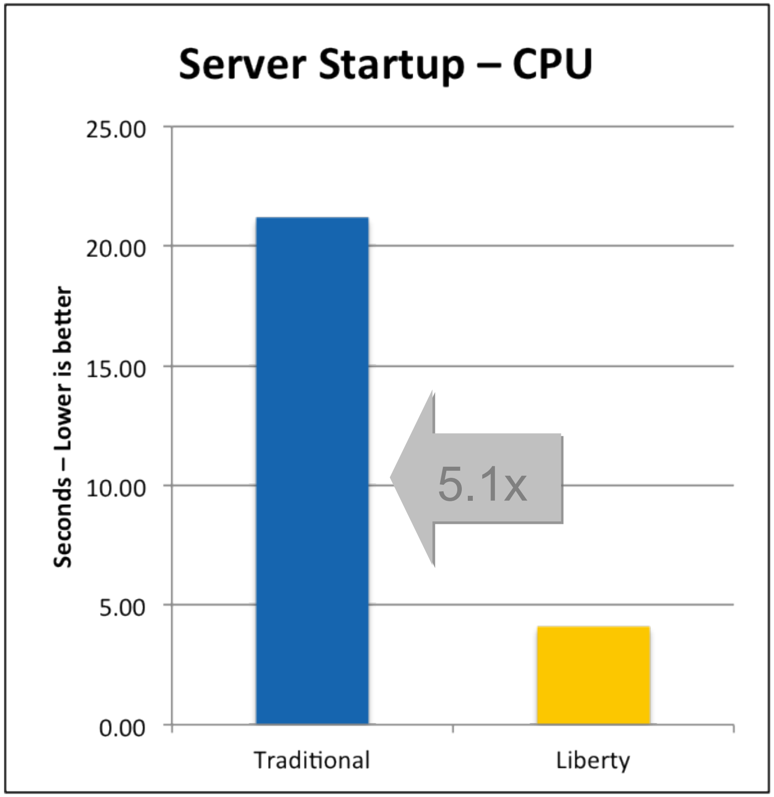
Application portability and stack consistency

- Liberty behaves *exactly the same* on all platforms out of the box
  - z/OS specific behaviour must configured if desired
- Administration is the same for all platforms out of the box
  - Server operations are controlled by the same `server` script
  - Logs, trace, and configuration live in the hierarchical file system and are tagged with the appropriate code page for easy viewing and editing
  - Existing server configurations can be brought to z/OS from distributed without modification
- An extremely light-weight, single process runtime
  - Removes deployment and runtime complications introduced by the split process, multi-JVM runtime of traditional WAS for z/OS

Why Liberty on z/OS?

Performance: Startup time – 3.2 seconds!

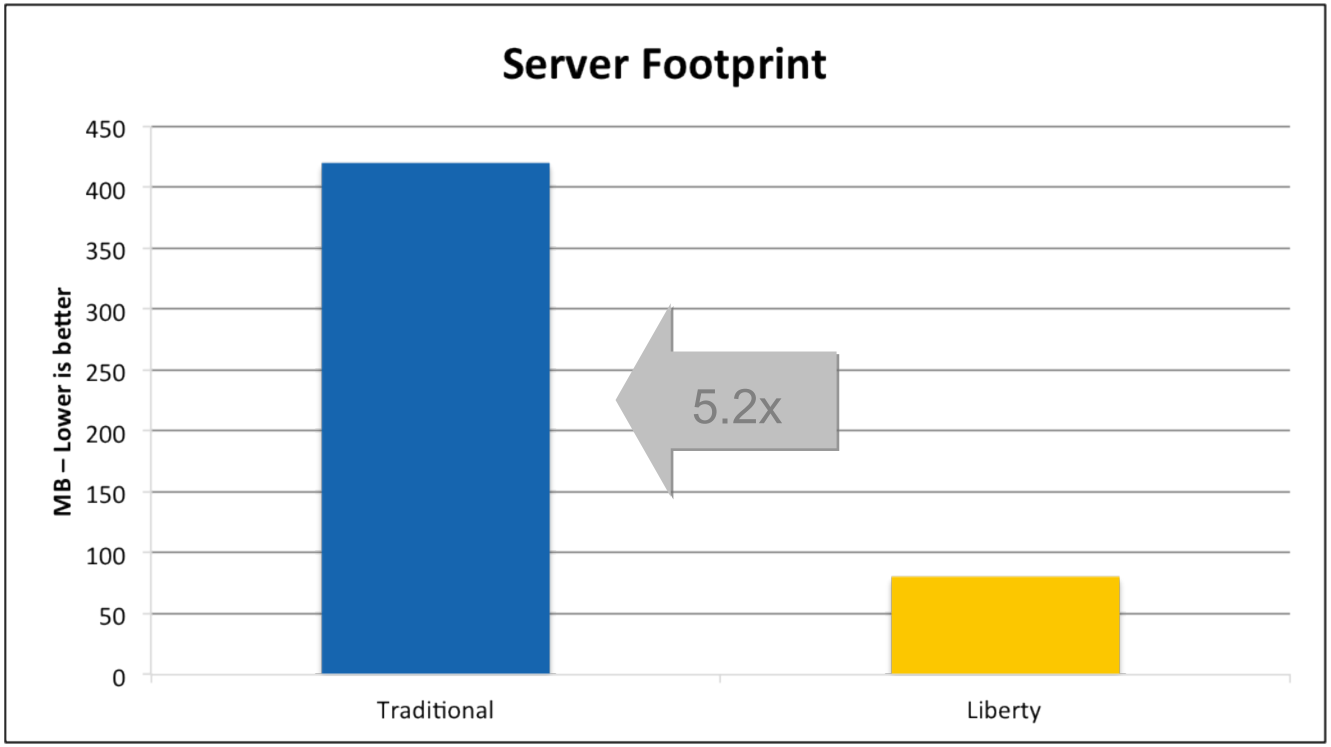




- Liberty – 64bit IBM Java 6.0.1, 64/64MB min/max heap, 60MB shared class cache, TradeLite installed
- Traditional – 64bit IBM Java 6.0.1, 1SR,128/256MB min/max CR heap, 256/512MB min/max SR heap, 75MB CR shared class cache, 75MB SR shared class cache, no applications installed

Why Liberty on z/OS?

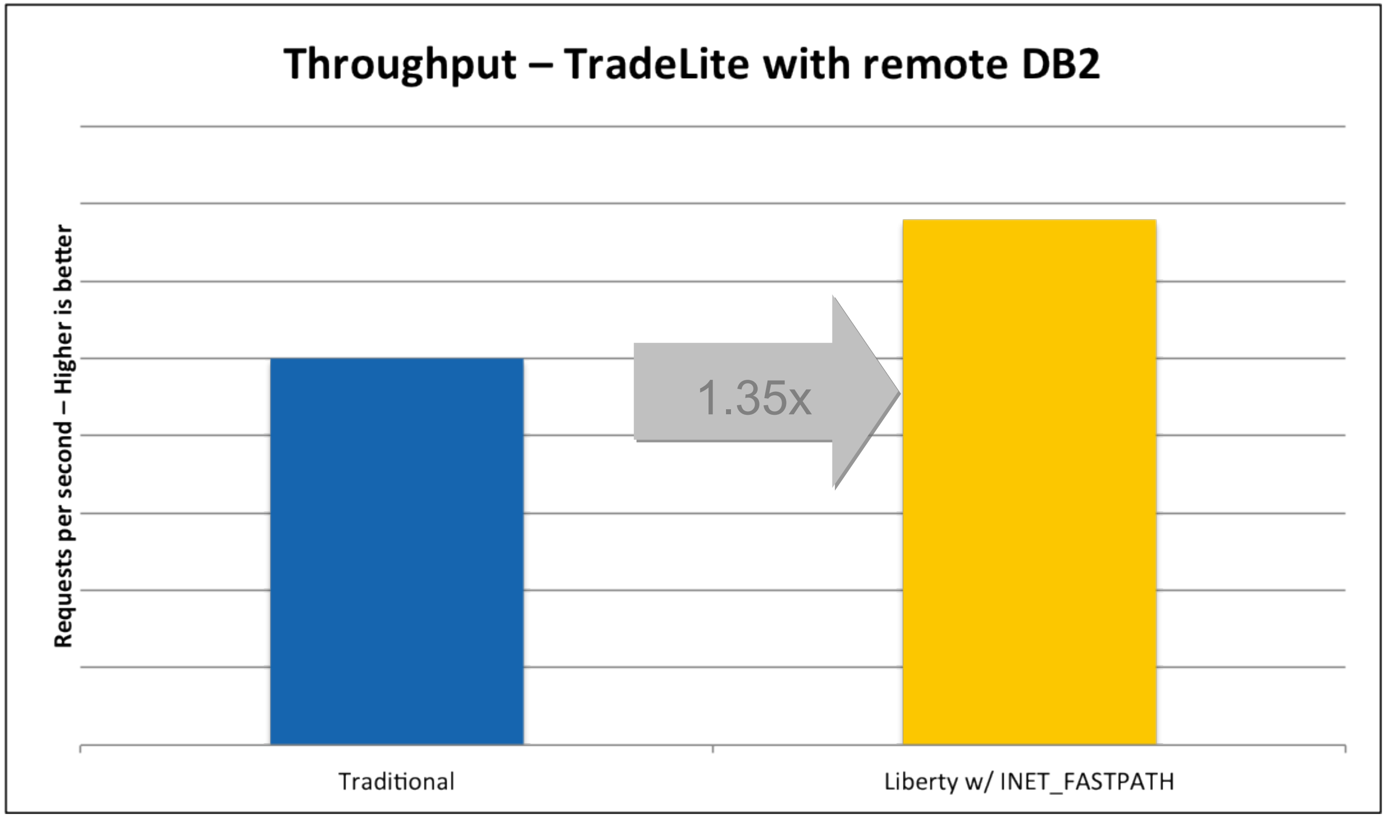
Performance: Memory footprint – 80% reduction



- 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

Why Liberty on z/OS?

Performance: Throughput – Up to 35% improvement



- z196, 2-way LPAR running z/OS 1.13
- 64bit IBM Java 6.0.1 with compressed references, 1M large pages, 2GB heap
- IBM DB2 for z/OS v10, JDBC T4 with keepDynamic
- _BXK_INET_FASTPATH=* set to enable CommServer “fast path” for Liberty

Liberty for z/OS – Key Use Cases

Accelerate application development and deployment while leveraging z/OS qualities of service

- **Test Web Applications using z/OS Resources** – Easily perform z/OS platform testing of web applications regardless of development platform
- **Lightweight Production** – Where a lightweight application server is appropriate for production web applications, leverage the rapid startup and small footprint of Liberty profile based applications
- **Incremental adoption of unique z/OS extensions** – Enable incremental exploitation of optional z/OS extensions to leverage z/OS qualities of service
- **Efficient packaging and deployment of applications** – Create and deploy Liberty profile applications as packages that include both the application and configuration

System z Social Media

- **System z official Twitter handle:**
  - @ibm_system_z
- **Top Facebook pages related to System z:**
  - Systemz Mainframe
  - IBM System z on Campus
  - IBM Mainframe Professionals
  - Millennial Mainframer
- **Top LinkedIn Groups related to System z:**
  - Mainframe Experts Network
  - Mainframe
  - IBM Mainframe
  - System z Advocates
  - Cloud Mainframe Computing
- **YouTube**
  - IBM System z
- **Leading Blogs related to System z:**
  - Evangelizing Mainframe (Destination z blog)
  - Mainframe Performance Topics
  - Common Sense
  - Enterprise Class Innovation: System z perspectives
  - Mainframe
  - MainframeZone
  - Smarter Computing Blog
  - Millennial Mainframer