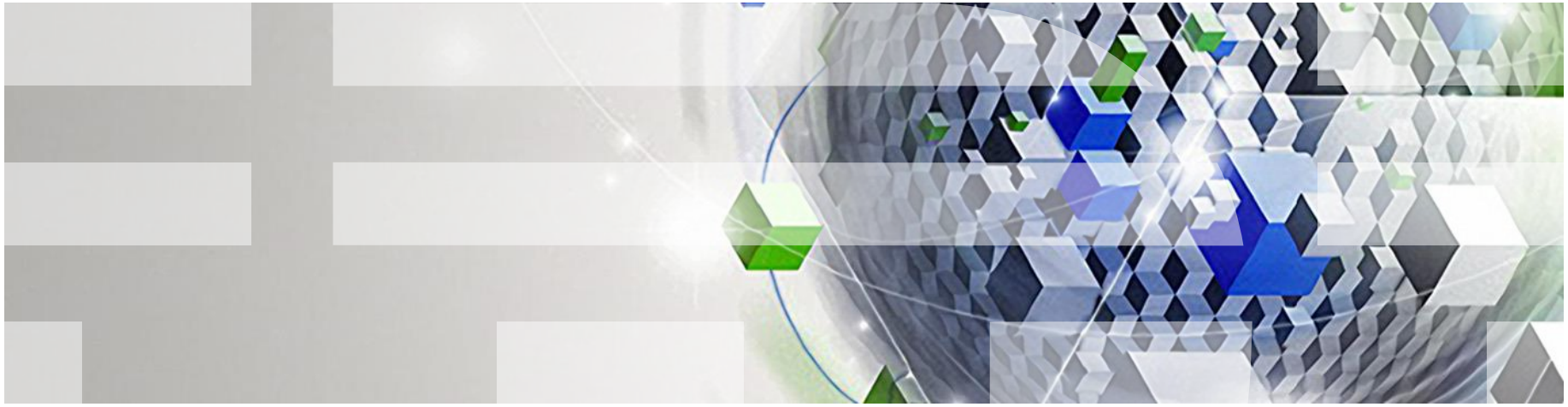
Give Me Liberty or Give Me……

um… let me get back to you….



David Follis
WebSphere Java Batch Architect
IBM Poughkeepsie
follis@us.ibm.com

Important Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.

IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

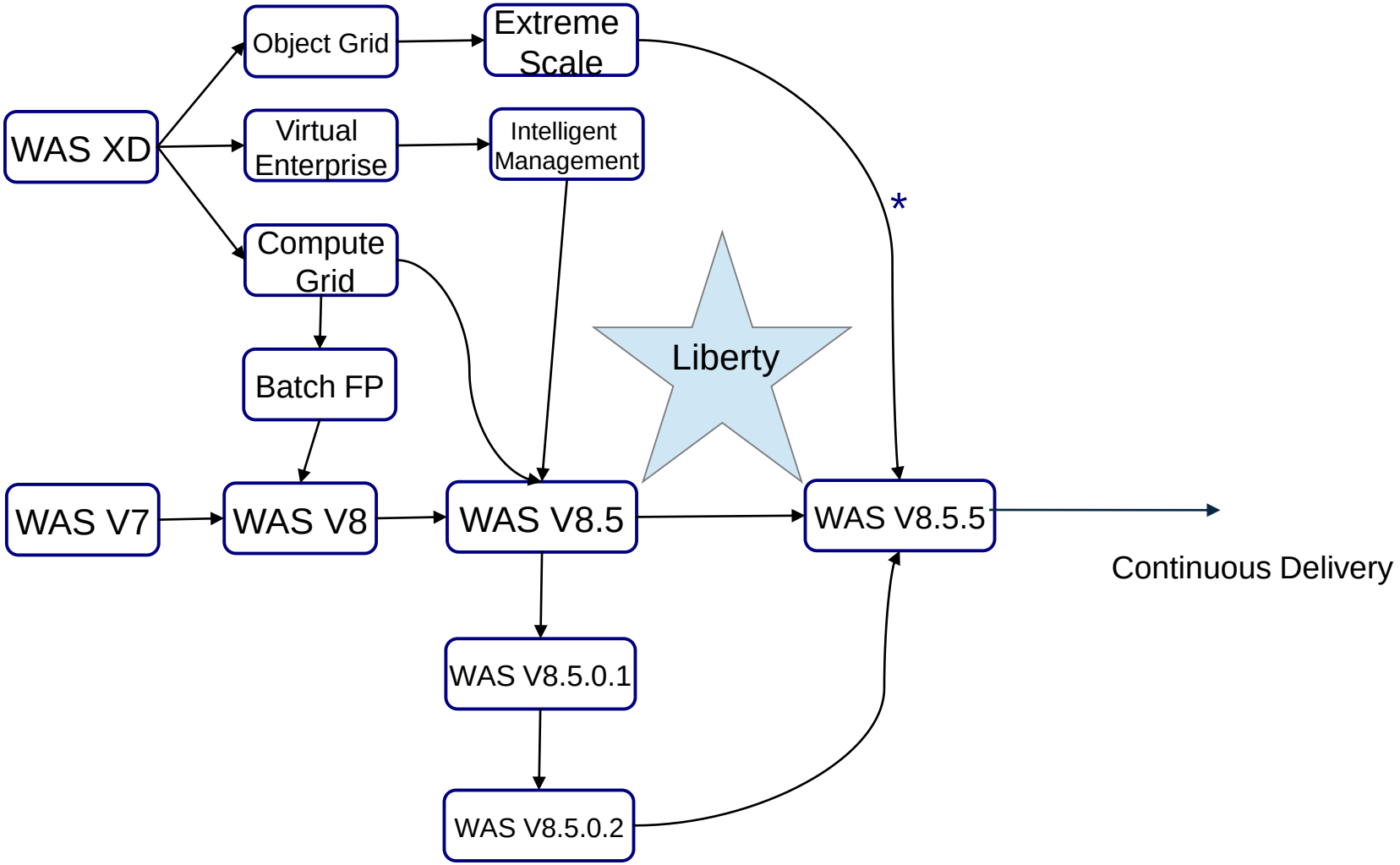
NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

- CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
- ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

Find the Match!

TWAS	Liberty Profile Technology in CICS
WLP	Install Manager
AMT	Optimized Local Adapters
WXS	z/OS Management Facility
WCG	Application Migration Toolkit
XD	WebSphere Compute Grid
WVE	WebSphere eXtreme Scale
IM	Intelligent Management
WOLA	WebSphere Full Profile
z/OS MF	z/OS Connect
z/OS Connect	WebSphere Liberty Profile
Liberty in CICS	WebSphere Extended Deployment

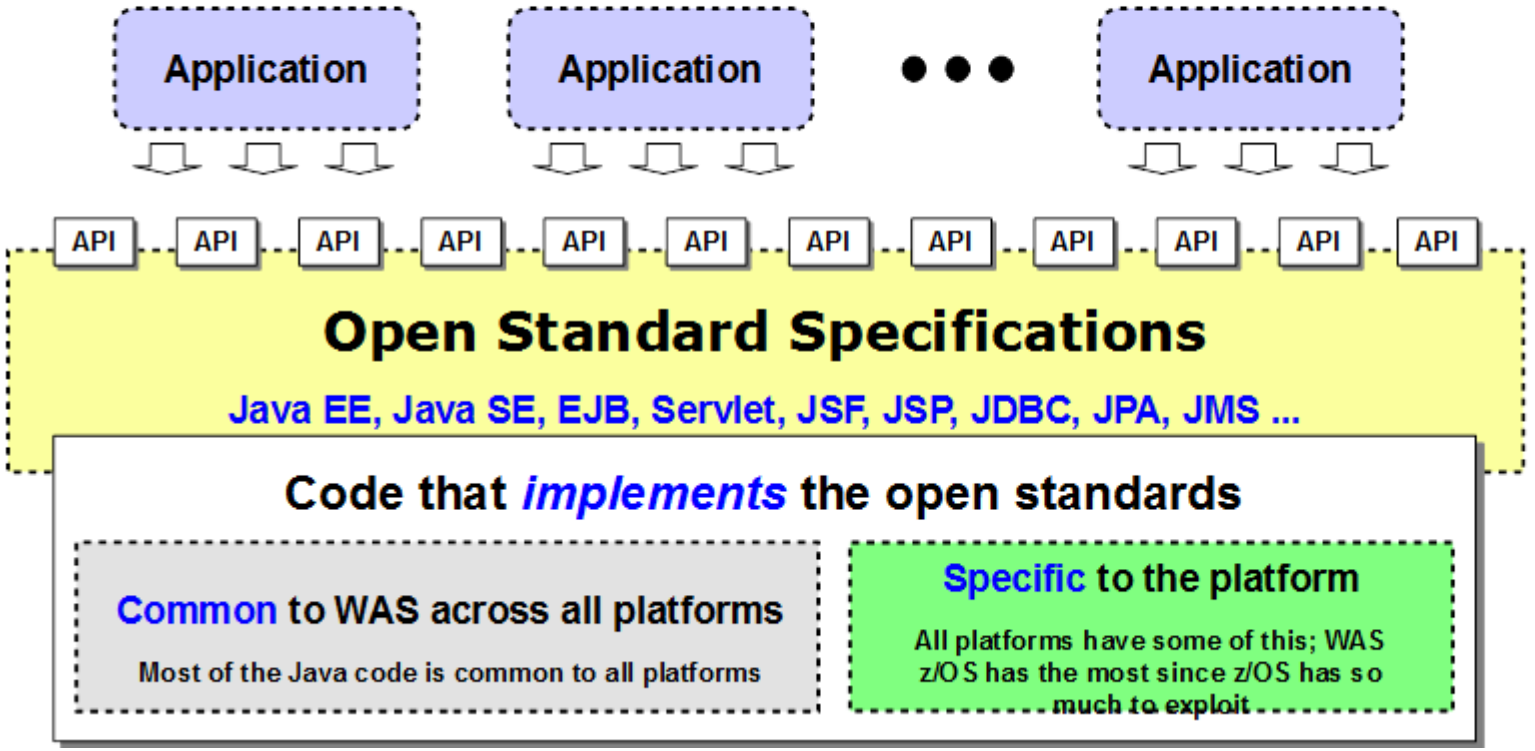
The Big Picture



* Client Only on z/OS

WAS is WAS

This is an important starting concept -- it's what makes application development a platform-neutral consideration:



WAS is common across platforms at this layer and above

Under the open specification line is where WAS takes advantage of platform attributes where they exist

Points of Commonality / Points of Differences

The following chart offers a summary of those things the same across platforms, and those things that are different on z/OS:

Commonality

Topology Design
WAS z/OS has the same cell, node, server and cluster design as WAS on other platforms.

Configuration File System Layout
The file system layout and the XML files contained in the file system is very similar to that on other platforms (not exact, but close)

Admin Console
Nearly identical except where you get down to things close to the operating system.

Operational Shell Scripts
Start server, stop server, etc., exist for WAS z/OS as well

Application Deployment
The same on all platforms

WSADMIN
The same on all platforms

Differences

Runtime construction
Involves running JCL jobs

z/OS Started Tasks
Servers run as z/OS started tasks, so START and STOP commands are unique to z/OS

Logging
Logging on z/OS is to JES. Logs can be sent to UNIX file system or HPEL if you wish.

Controller / Servant model
Servers have a multi-region model that we'll talk about at some length later.

z/OS Exploitation/Integration

WLM

z/OS Workload Manager

- Controller / Servant structure as discussed on previous chart
- Request classification for separate service classes and reporting classes

SAF

z/OS System Authorization Facility

- Sysplex-aware security definition repository and resource access control
- Userids and passwords, SSL certificates, EJBROLE definitions
- Security workshop covers WAS z/OS security in detail (ask for details if interested)

SMF

z/OS System Management Facilities

- SMF 120.9 record to record detailed information about request activity
- Useful for analysis and chargeback
- See WP102205 at ibm.com/support/techdocs for guide to SMF Techdocs

MODIFY

z/OS MODIFY interface

- Allows dynamic operations against WAS z/OS servers
- Long list of actions to display and act up on server operational behavior

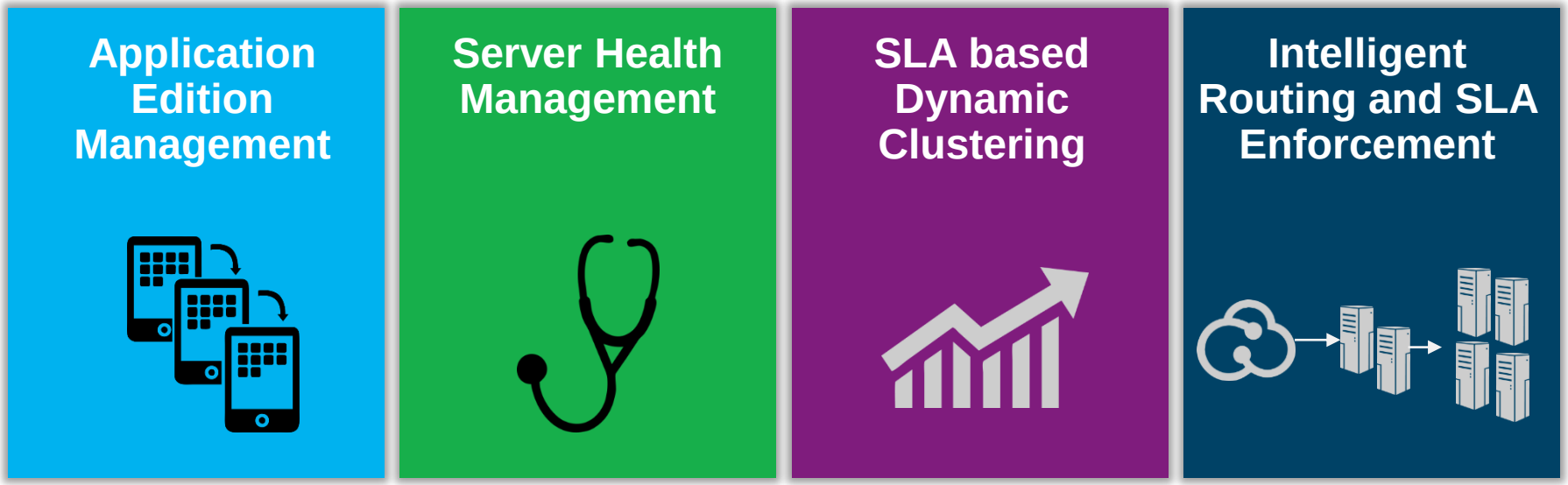
Cross Memory

z/OS Cross-Memory Exchange

- DB2 Type 2 connector, CICS EXCI, MQ BINDINGS, WOLA
- Low latency, better security

WAS ND – Intelligent Management

Part of WAS ND V8.5.0+ and WAS for z
Added Liberty Profile (assisted lifecycle subset) in V8.5.5.1



Better TCO through management efficiency and performance, Intelligent Management delivers the ability to sense and respond quickly to changes

Up to **45%** **less hardware**

Up to **60%** **less administration**

Up to **45%** **less software**

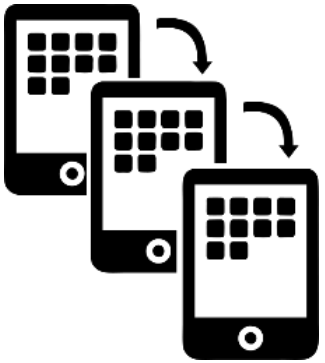
Up to **90%** **fewer outages**

Source: Based on 60+ Operations Optimization Value Assessments done to date by IBM for real customers
Cost reductions are compared to traditional WAS ND deployment

Application Edition Management

Applications can be upgraded or downgraded without incurring outages or requiring additional hardware and license costs

- Upgrade Applications without interruption to end users
- Concurrently run multiple editions of an application
  - Automatically route users to a specific application
- Multiple editions can be activated for extended periods of time
- Rollout policies to switch from one edition to another without service loss
- Easily update OS or WebSphere without incurring down time
- Easy-to-use edition control center in admin console
- Full scripting support



Validation Mode

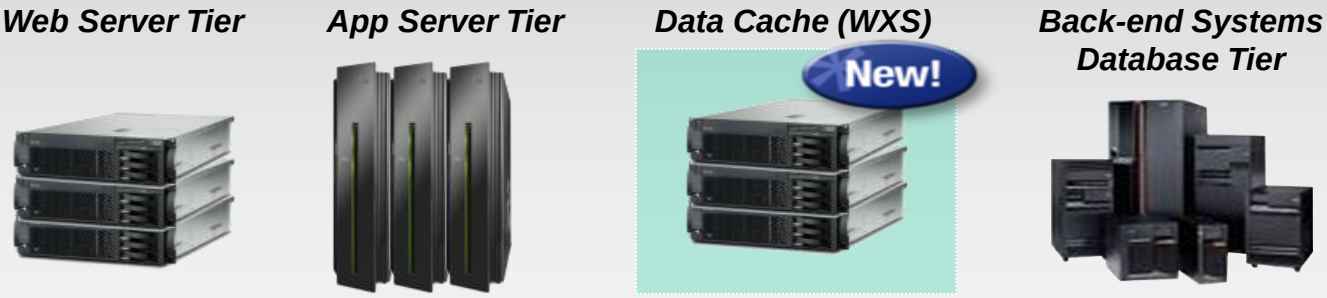
Rollout Policies

Concurrent Activation

Elastic Caching for WAS V8.5.5

WebSphere eXtreme Scale entitlement with WAS

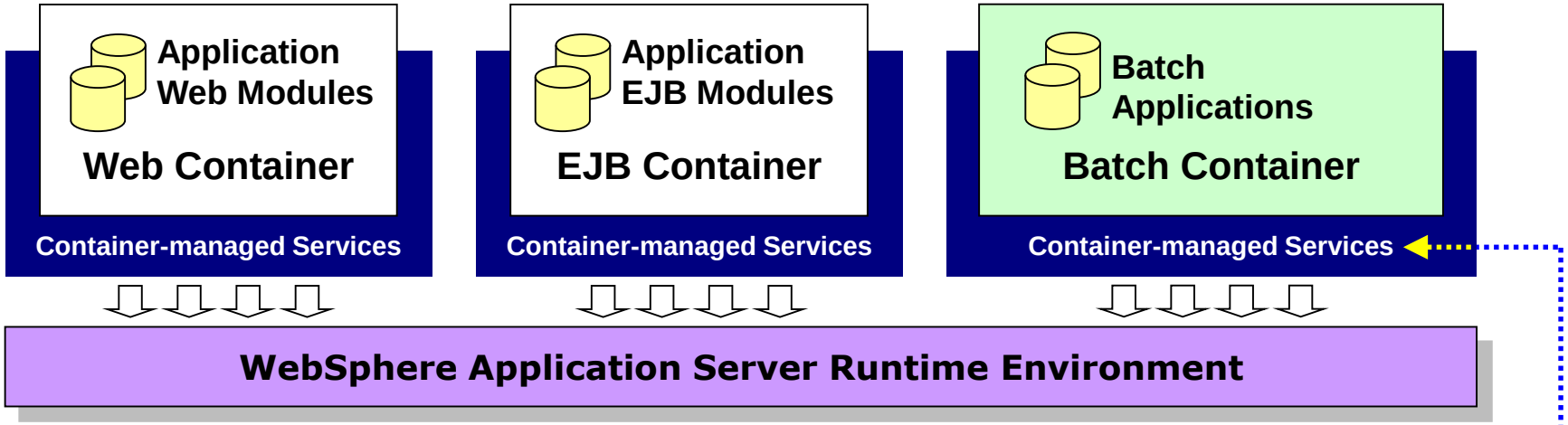
- Customers can now develop with caching in mind
- ***Benefits for developers – free access*** to caching for unit testing on developer's desktop
- Benefits for business – improvements to performance at lower cost
- Ability to enhance customer experience for web sites which can also improve revenue



- ✓ WAS Network Deployment includes full entitlement to eXtreme Scale
- ✓ WAS includes entitlement to use eXtreme Scale for Session Management and distributing the WAS DynaCache service

What is WebSphere Java Batch?

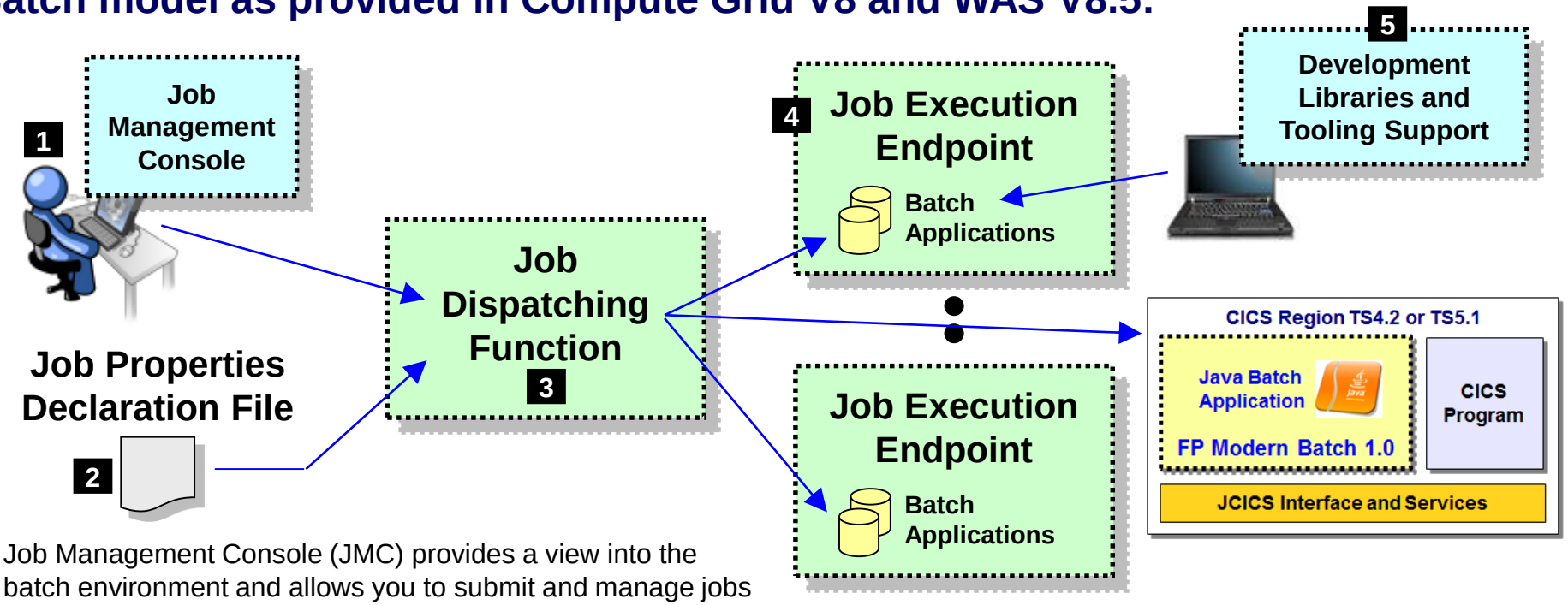
A comprehensive batch execution platform, built on WebSphere Application Server runtime and providing development and job management tools.



Batch job dispatching and management system	**Job resiliency services (skip record, step retry)**
Data record read and write support services	**Parallel job management and execution services**
Checkpoint and job restart services	**COBOL module call services**

Overview of the Management and Execution Model

This picture illustrates some of the key components of the WebSphere Java Batch model as provided in Compute Grid V8 and WAS V8.5:

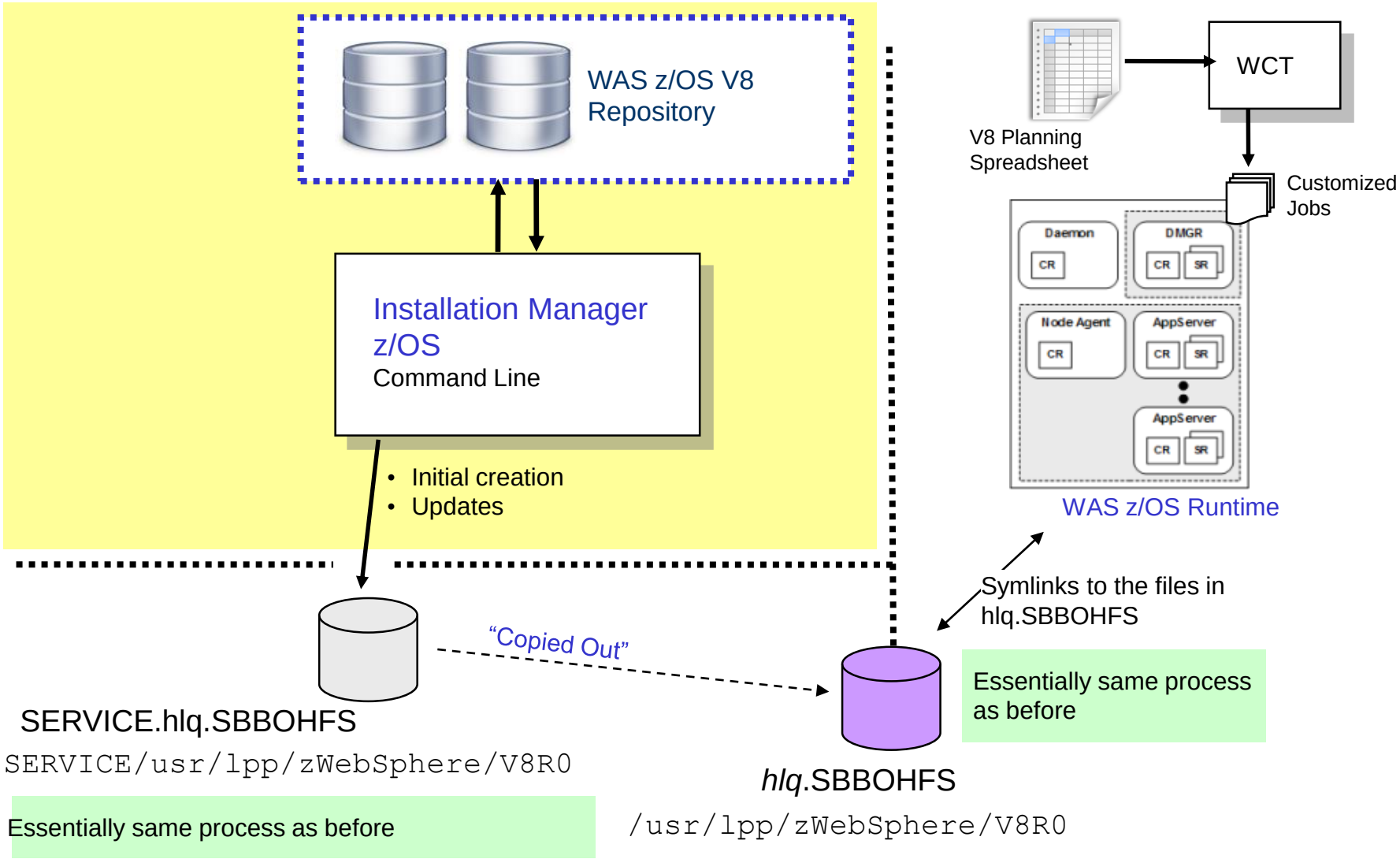


1. Job Management Console (JMC) provides a view into the batch environment and allows you to submit and manage jobs
2. Job declaration file (xJCL) provides information about the job to be run, such as the steps, the data input and output streams and the batch class files to invoke
3. The Job Dispatching function interprets the xJCL, dispatches the job to the endpoint where the batch application resides, and provides ability to stop and restart jobs
4. The Execution Endpoint is a WAS server in which the deployed batch applications run
5. The development libraries (eg. jZOS) and tooling assist in the creation of the batch applications

A comprehensive Java batch execution platform
Built on the proven Java runtime environment of WebSphere Application Server

Install Manager and z/OS

This is entirely new for WAS z/OS V8 … the use of command line IM on z/OS to create and maintain the hlq.SBBOHFS file product file system:



Application Migration Toolkit

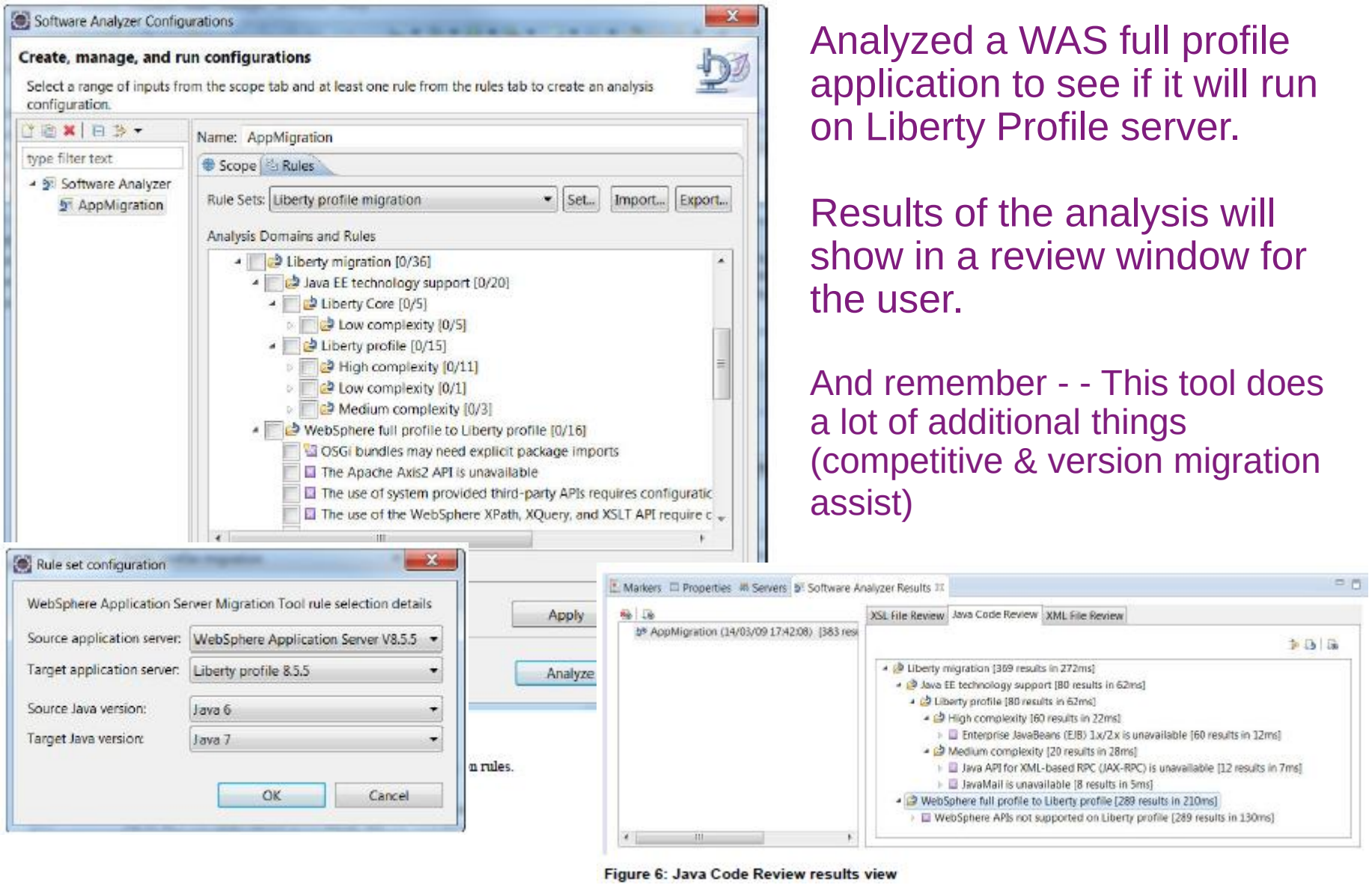


Figure 6: Java Code Review results view

Analyzed a WAS full profile application to see if it will run on Liberty Profile server.

Results of the analysis will show in a review window for the user.

And remember - - This tool does a lot of additional things (competitive & version migration assist)

WebSphere Application Server **Liberty Profile**

For rapid development and light-weight production deployment with high scalability

- Web Profile certified (Liberty Core Edition)
- Small footprint (< **54MB**), quick startup (< **3 sec**)
- Developer-first design of simple, shareable XML configuration
- Dynamic runtime and configuration
- Unzip install and deploy
- Fidelity to WebSphere Application Server Full Profile
- Add custom features and integrate 3rd party components via Liberty extensions interface
- Install new features from repository with no server restart
- Lightweight collective management scales to 10,000 servers

Simplified Server Configuration

```
<server>
    <featureManager>
        <feature>jsp-2.2</feature>
        <feature>jdbc-4.0</feature>
    </featureManager>

    <logging traceSpecification="webcontainer=all=enabled:*=info=enabled"/>

    <application name="tradelite" location="tradelite.war"/>

    <dataSource jndiName="jdbc/TradeDataSource">
        <properties.derby.embedded databaseName="${server.config.dir}/tradedb"/>
    </dataSource>
</server>
```

Features control which capabilities (bundles) are installed in the server

'singleton' configurations specify properties for a runtime service like logging

'instance' configurations specify multiple resources like applications and datasource definitions

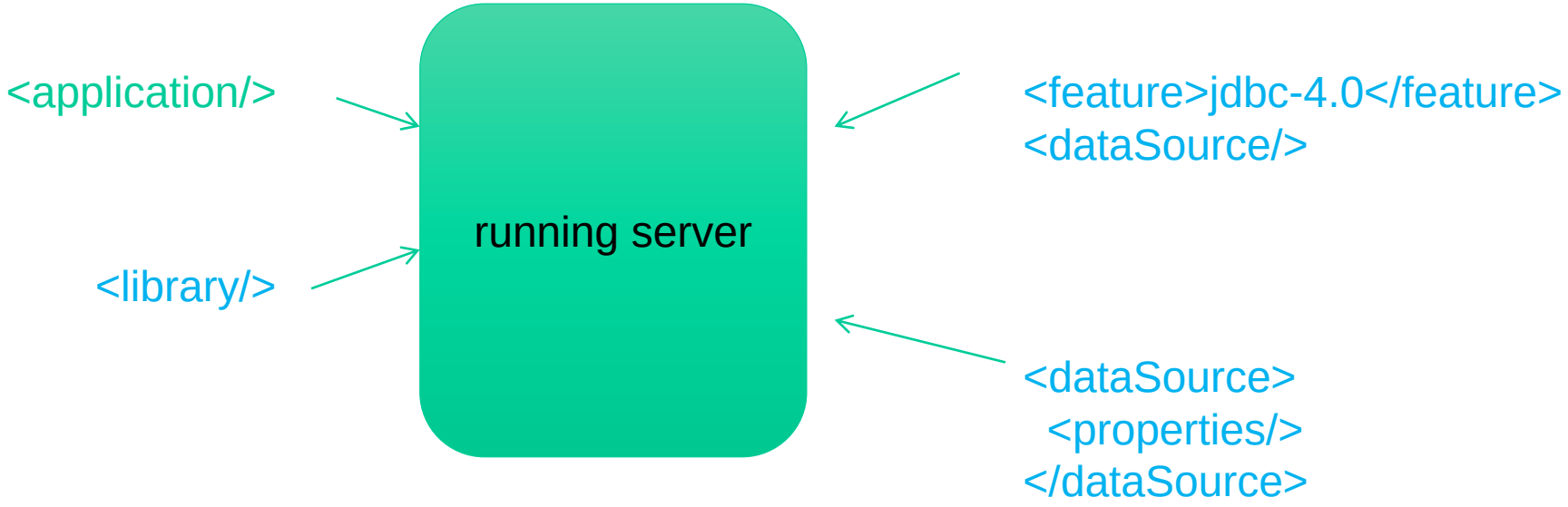
Any of this configuration could be put into a separate xml file and 'included' in this 'master' configuration file

Dynamic Configuration

Dropin application install and update

Configuration files also monitored for updates

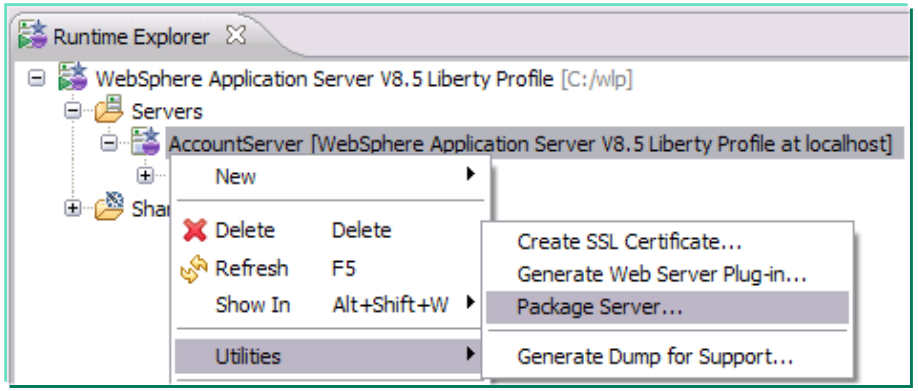
- All configuration changes are dynamic



Liberty – “Packaged Server” Deployment Option

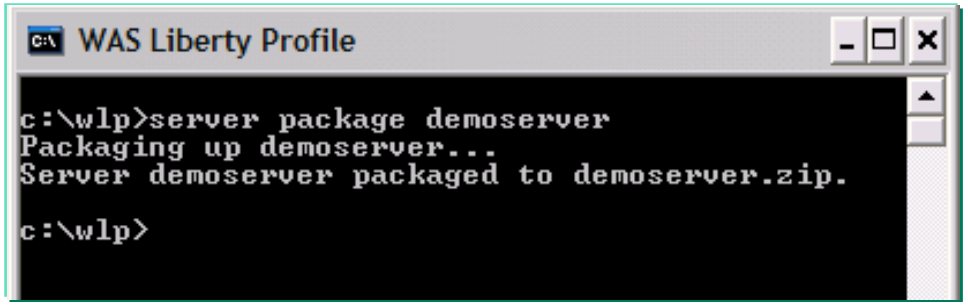
Package up a compressed archive of a configured Liberty server type along with its applications

- Directly from Eclipse environment
- Resulting zip can be copied to integration or production environment and unzipped.



For test automation outside the IDE, a command-line program to manage the lifecycle of server instances:

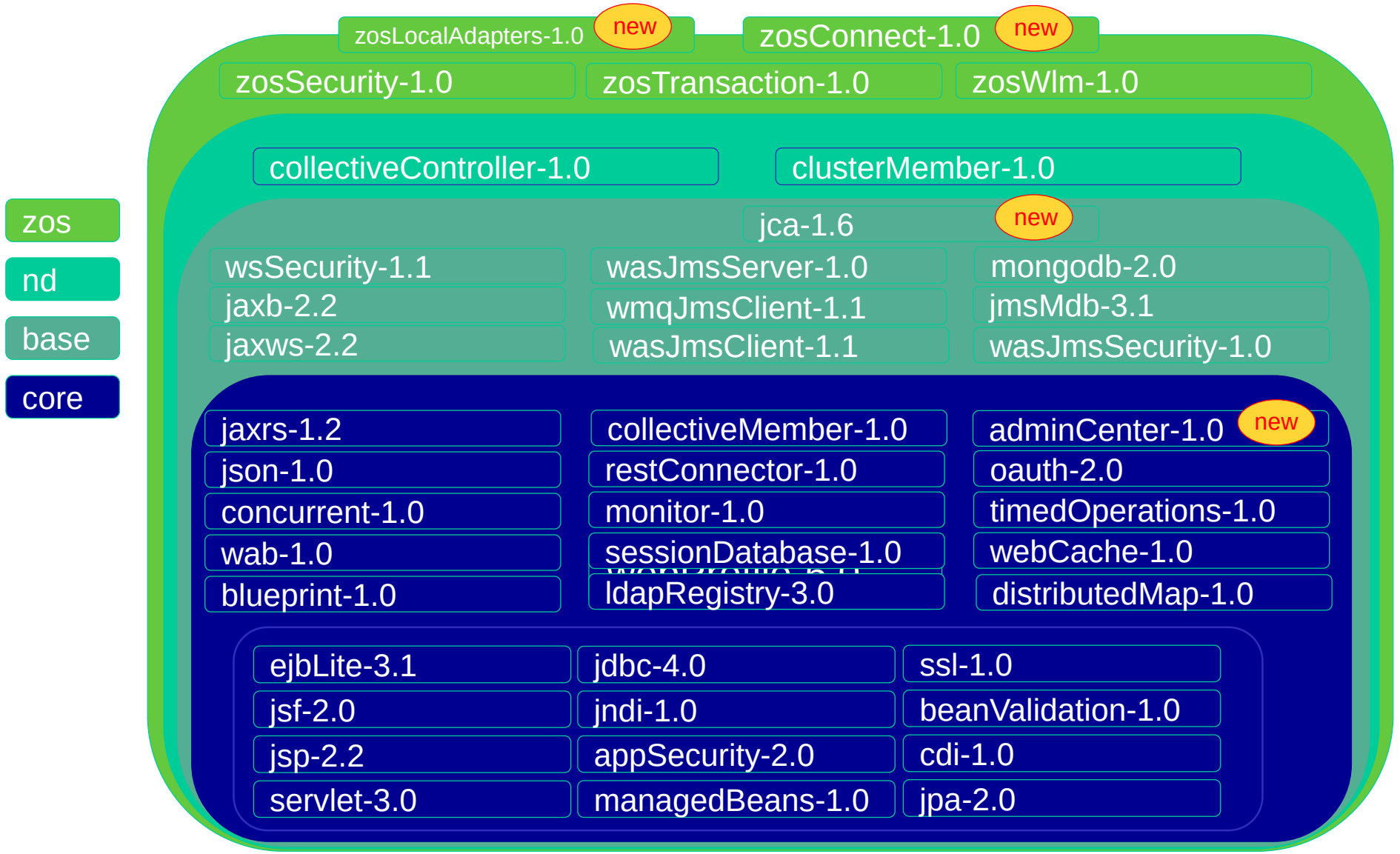
- Create [serverName]
- Start and stop [serverName]
- Package [serverName]
- Status [serverName]



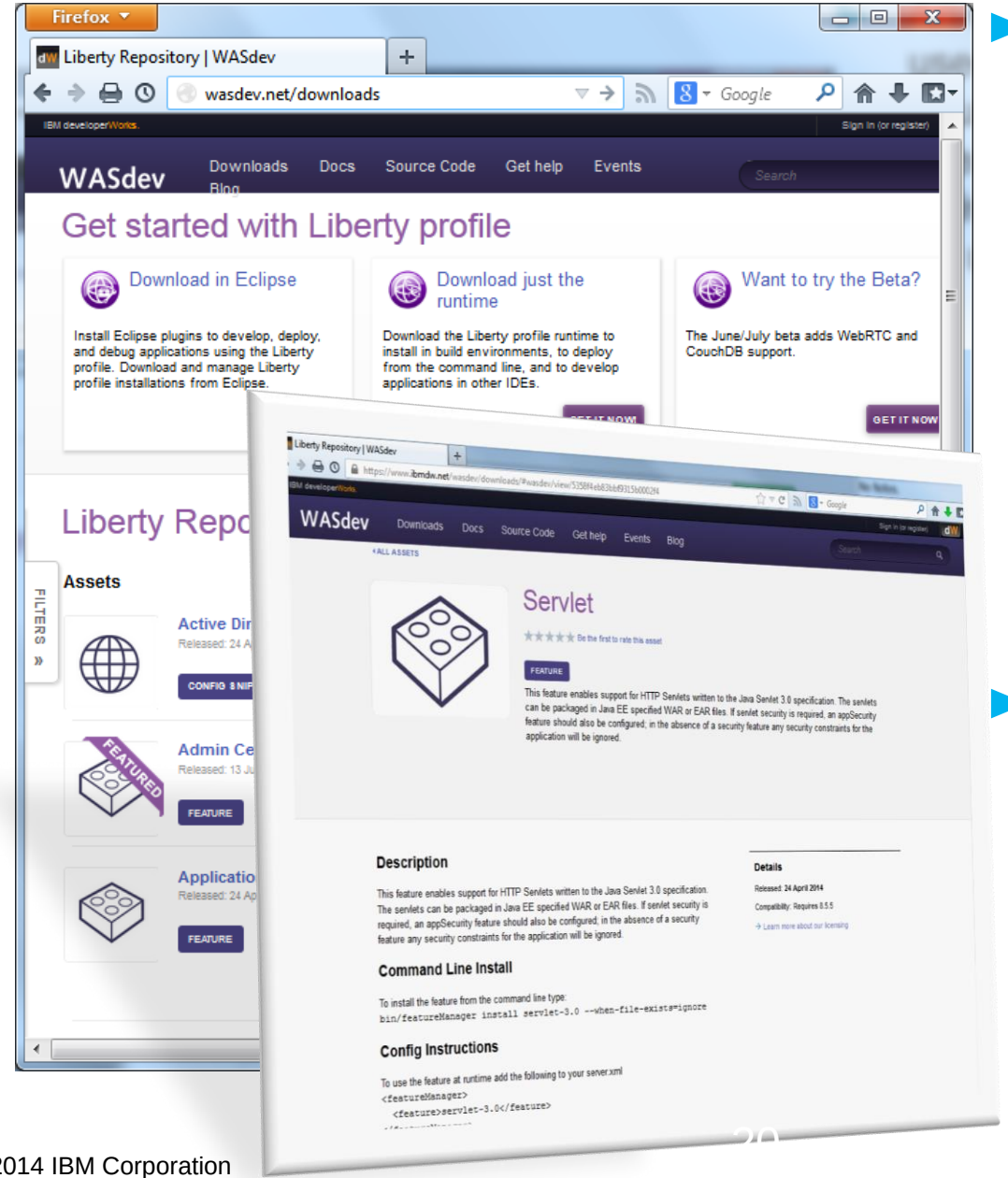
Updates to configuration of running server are effective immediately.

Add/remove apps dynamically by drag/drop to monitored directory.

…Based on Composable Features



New Features Being Delivered In A New Way



- An **online repository** to deliver new Liberty function to users. Starting with the current release, new Liberty features will be delivered via the Liberty repository for use in production.
  - Accessed through Installation Manager, Liberty featureManager command and WDT
  - Dependencies calculated and installed
  - Service applied the same way regardless of how feature installed.
- Other Content available in the repository:
  - Code that allows non-Liberty products to easily integrate with Liberty
  - Samples using Liberty features and/or 3rd party tool integrations that work as part of Liberty's Ecosystem

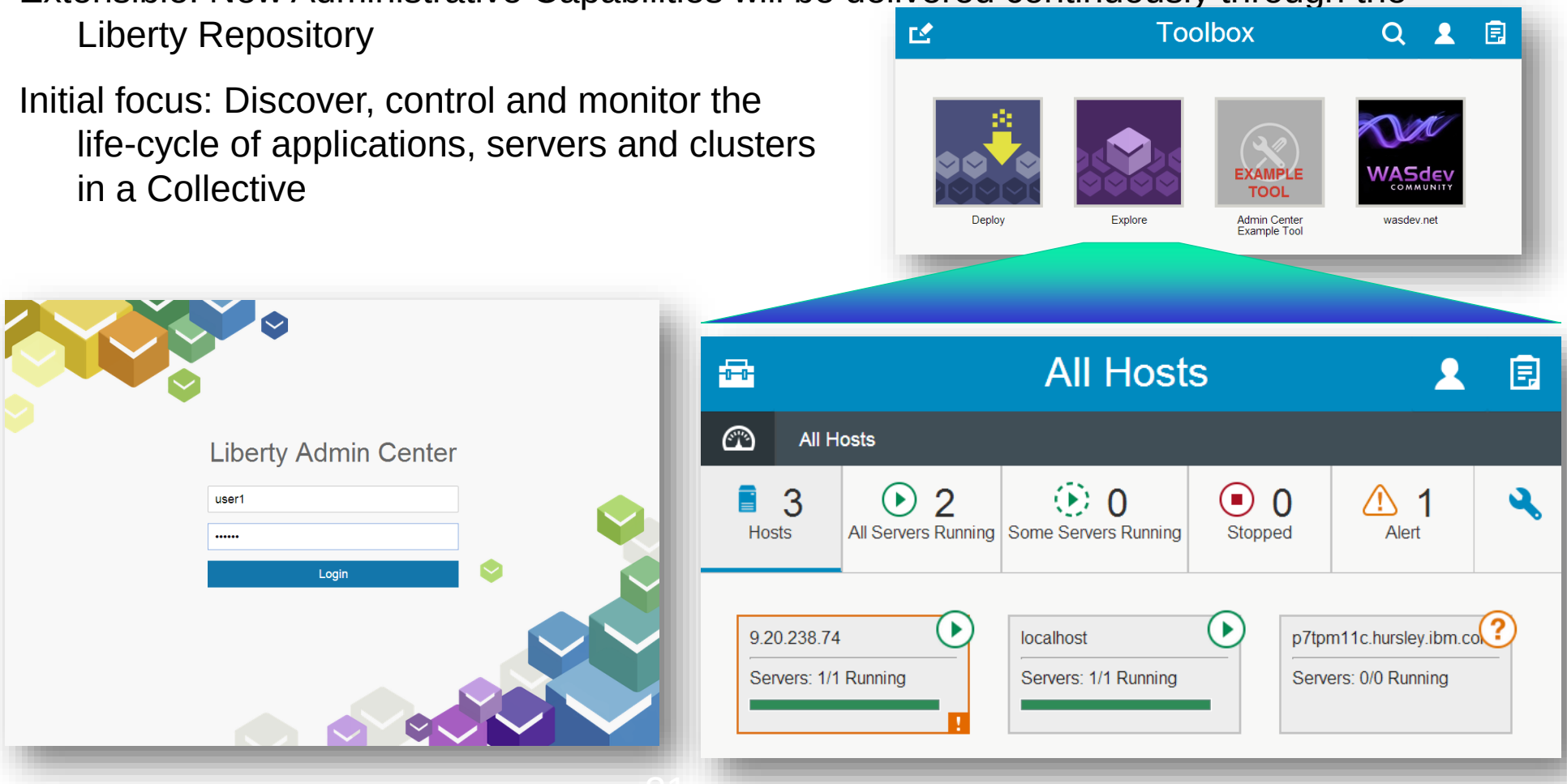
…Including the Liberty Administrative Center

Evolve from one-size-fits-all console

Lightweight, task-oriented applications / app store approach with customizable toolbox for "right-sized" UI per user / device

Extensible: New Administrative Capabilities will be delivered continuously through the Liberty Repository

Initial focus: Discover, control and monitor the life-cycle of applications, servers and clusters in a Collective



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

Feature – z/OS Transactions

Extends the WebSphere transaction manager

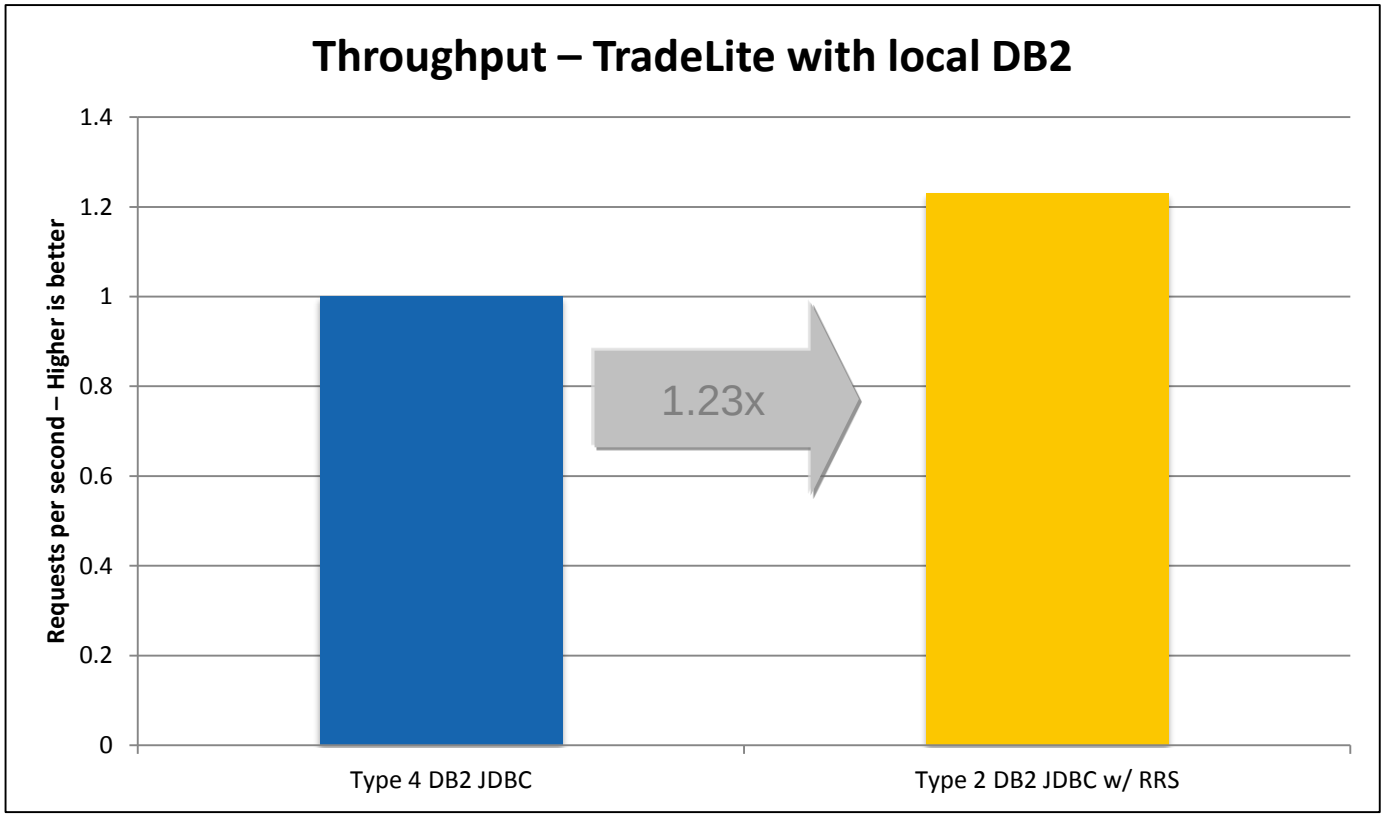
- Provides native transaction context management via MVS context services and resource recovery services (RRS)
- Implements 2PC across JTA/XA resource managers and RRS enabled resource managers

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

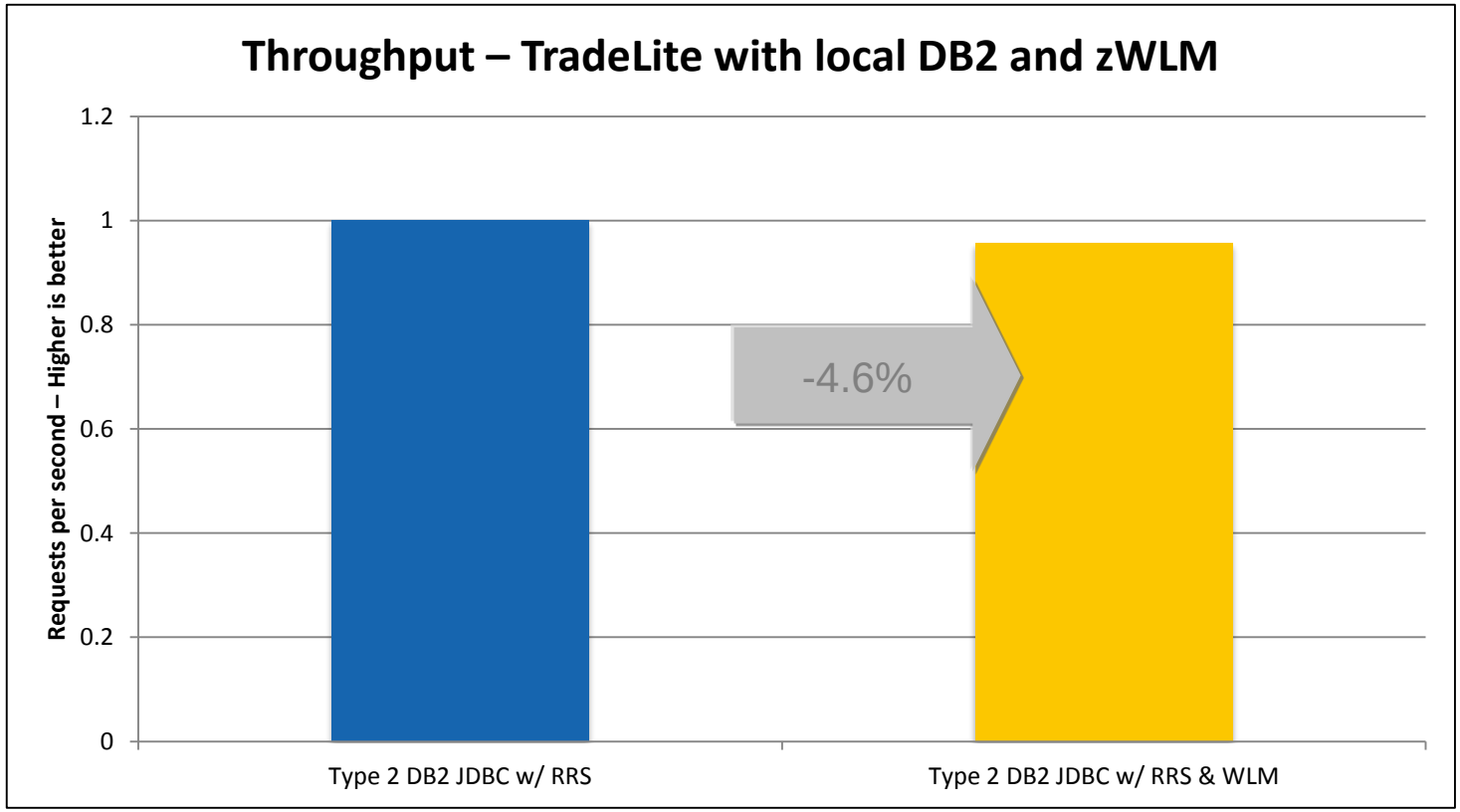
```
<server description="mvsWorkloadManagement">

    <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

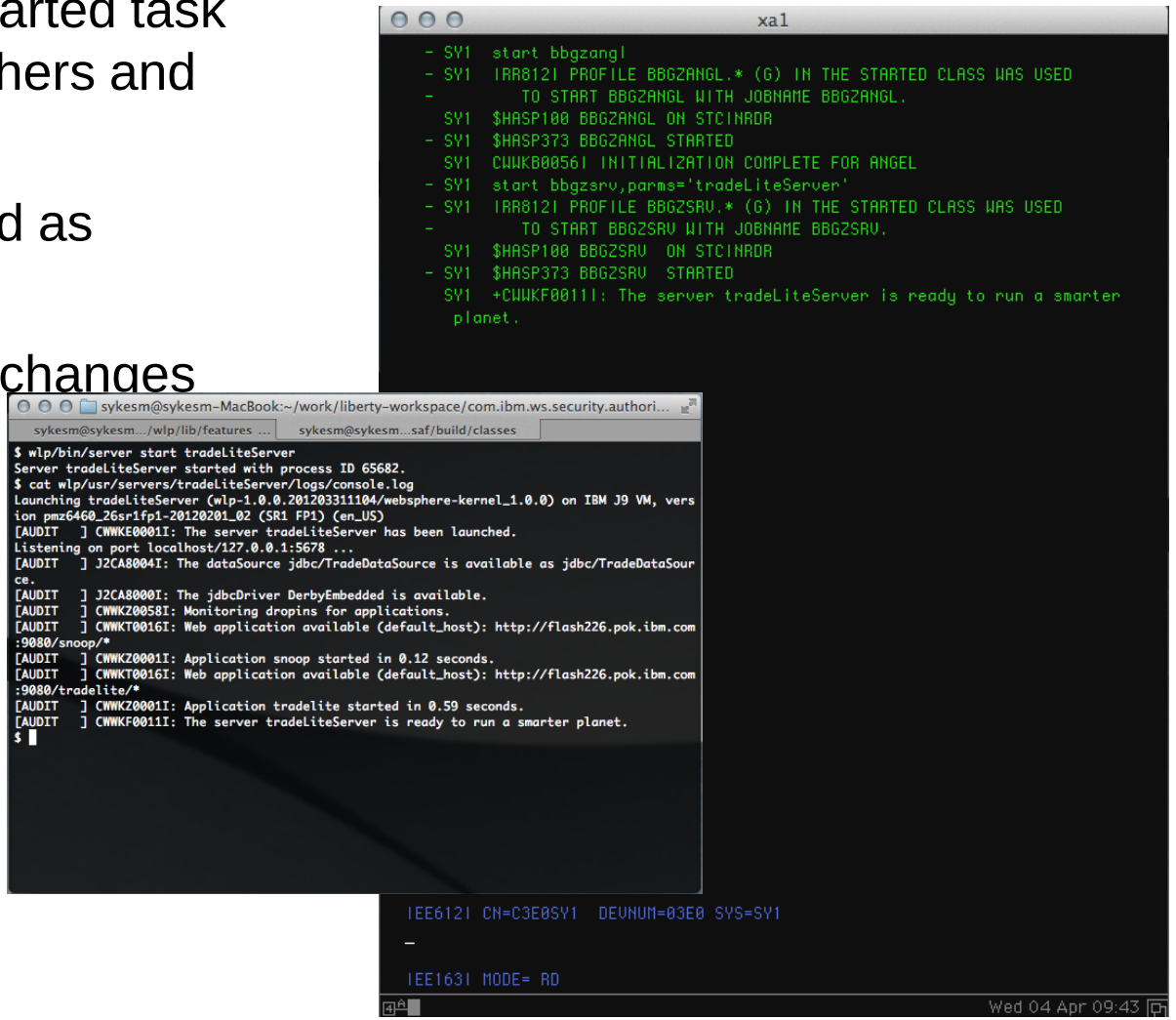
```
<server description="securityExample">

    <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

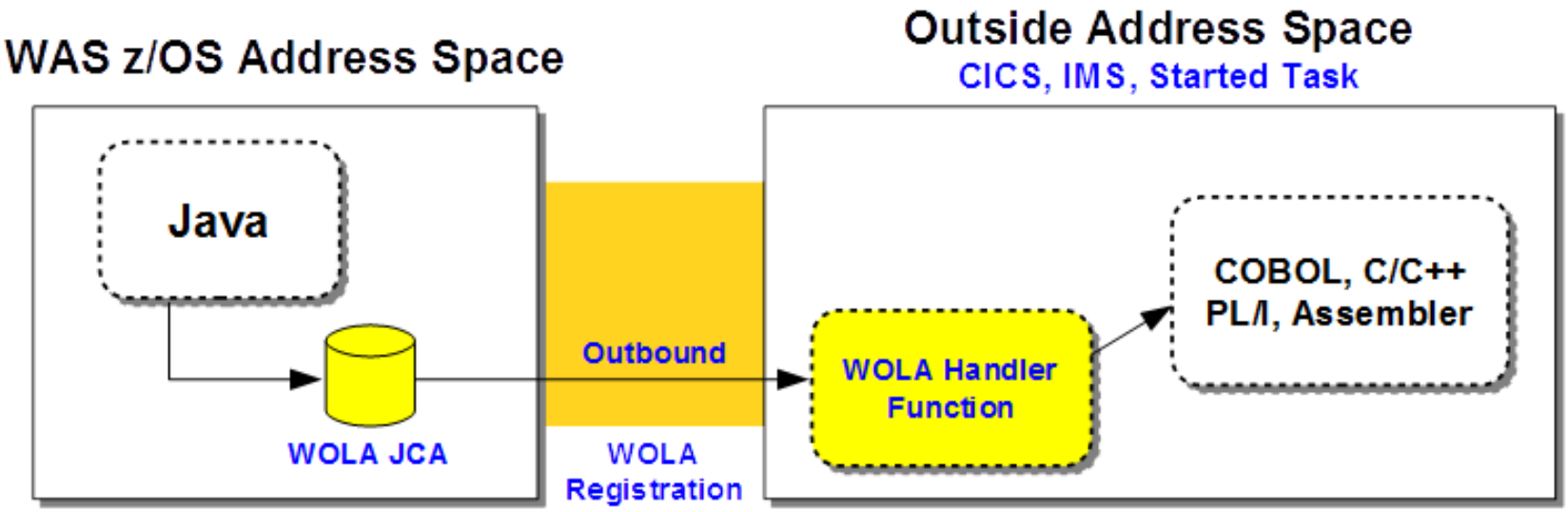
Important messages routed as
WTOs for automation

Modify commands enable changes
to trace specification
or to request a
diagnostic dump



WOLA - Outbound

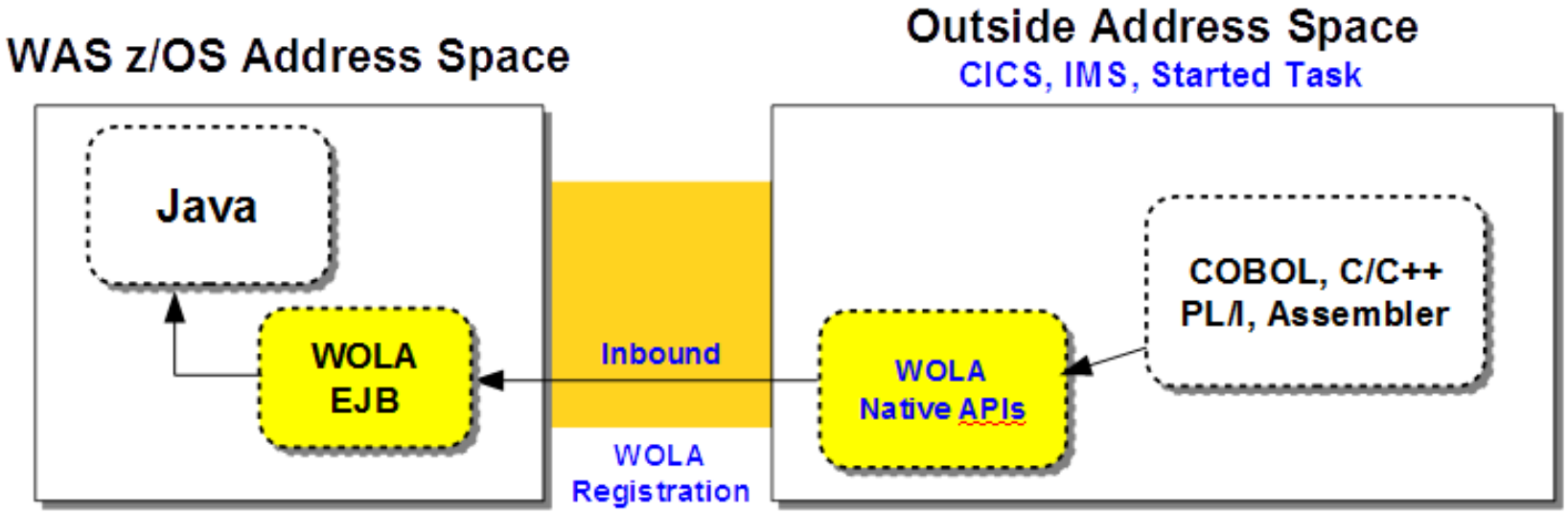
Java program in WAS initiates conversation to program in external A/S:



- Outbound = Java taking to outside (and getting response)
- Java uses supplied WOLA JCA resource adapter
- External requires a "WOLA Handler" function
  - In CICS this takes form of a supplied long running task (BBO$, more details coming)
  - For IMS this can be handled with JCA resource adapter mapping to OTMA
  - Otherwise, this involves using the WOLA native APIs (much more detail on this)

WOLA - Inbound

External program calls into WAS

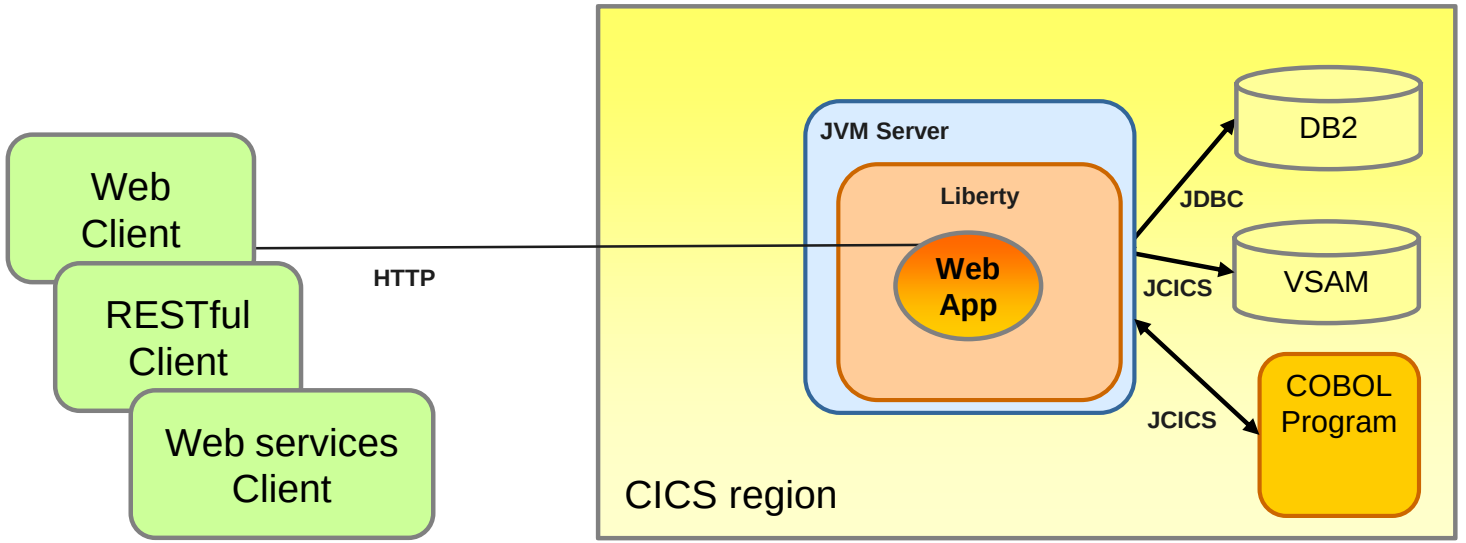


- Inbound = Outside talks to Java target in WAS
- WOLA Native APIs required for this (more details coming)
- In WAS target is an EJB
  - Implements execute() method
  - Implmenents interfaces using supplied WOLA class libraries
  - Business logic in same EJB, or other EJB

z/OSMF is designed to use the **Liberty profile** in IBM WebSphere Application Server for z/OS, this improved and simplified

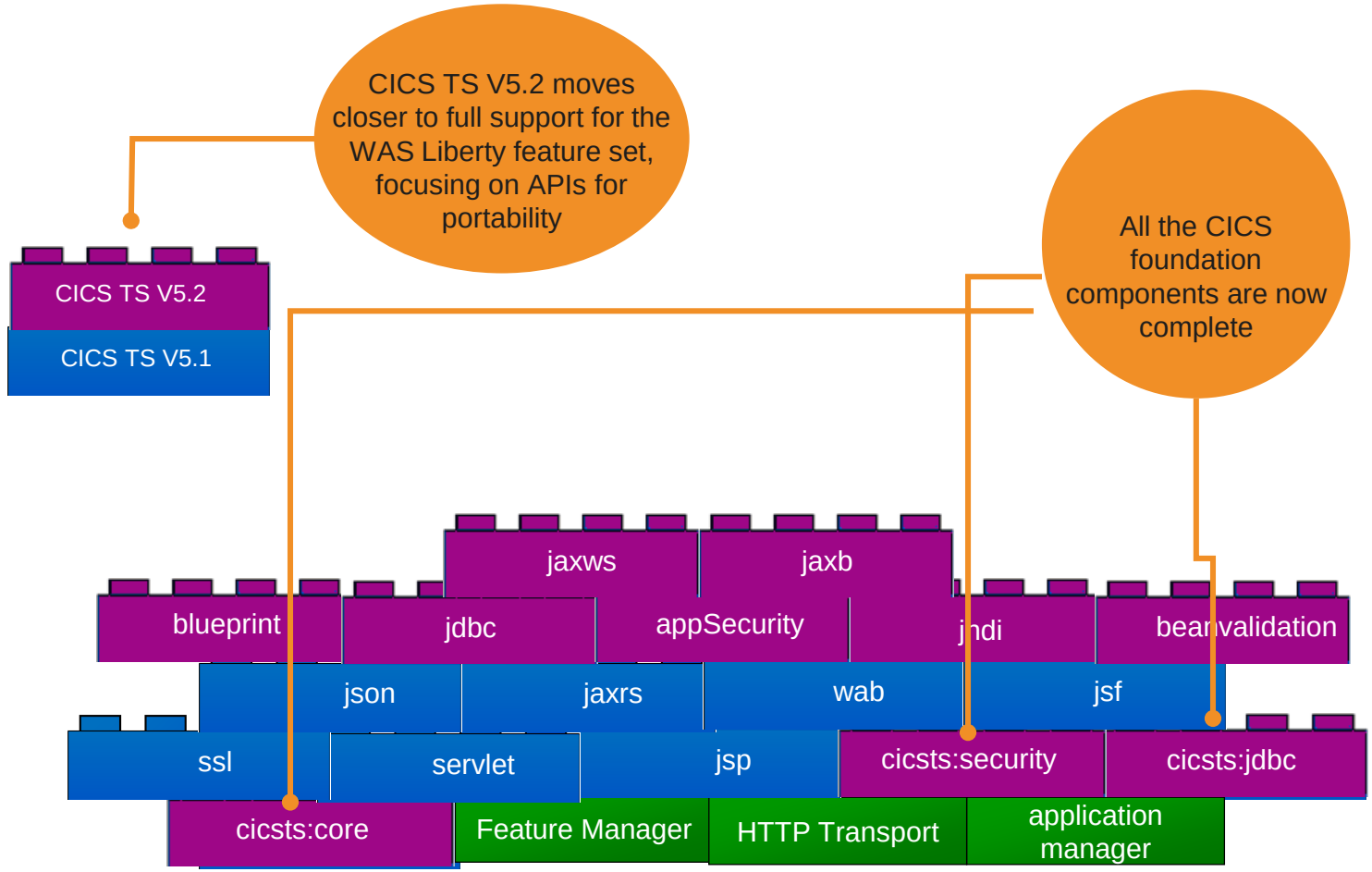
- **Packaging:** Provide a smaller and faster product package that requires fewer resources. WAS OEM is now removed from z/OSMF, WAS Liberty Profile is part of z/OSMF package, the new package footprint is 300+ MB
- **Installation and Configuration:** Setup no longer requires two separate configurations for runtime(WASOEM) and application(z/OSMF), reduced to one single stream configuration (one setup only), this change also results fewer overall prompts and variables.
- **Service:** Follow the normal z/OS model through normal SMPE receive/apply and restart z/OSMF to pick up new service, no longer required separate step for activation
- **Performance:** Faster startup (<15 seconds)

Liberty in CICS scenarios

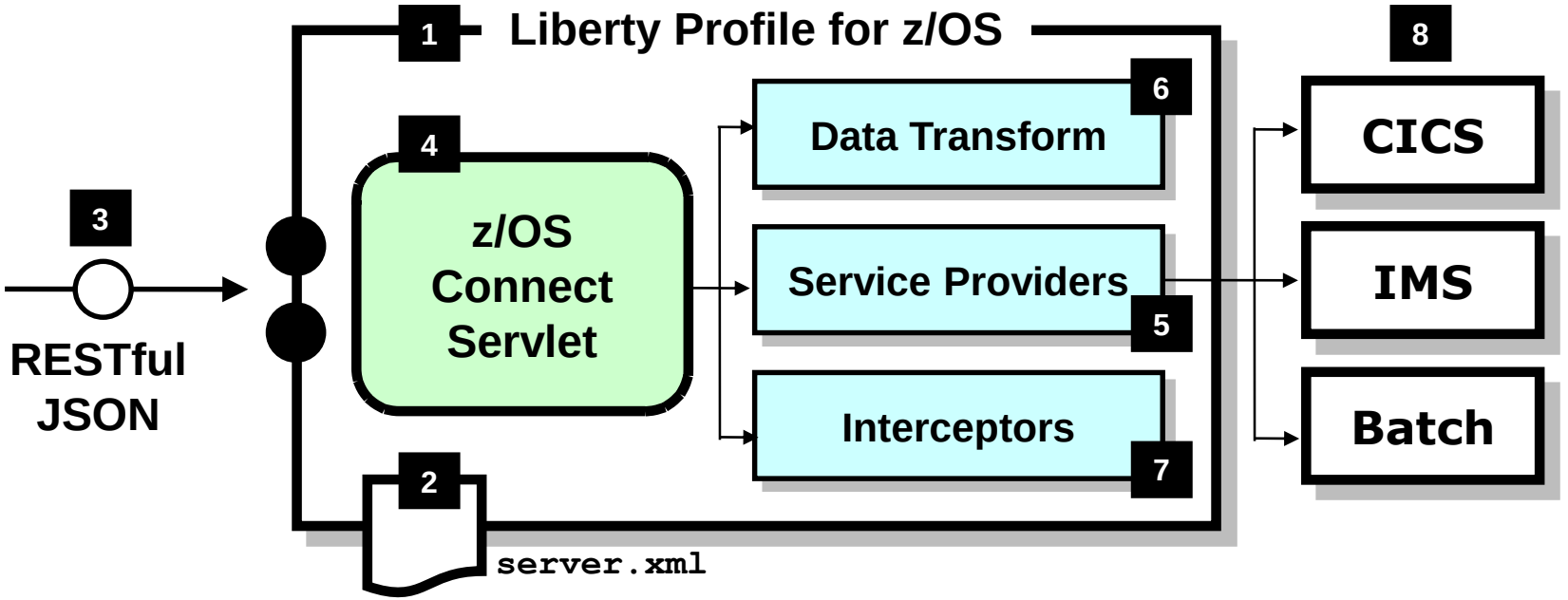


- Typical scenarios for Web app. deployment to CICS
  1. Access to existing CICS programs or data VSAM data
  2. Sharing access to DB2 tables controlled by CICS
  3. Reducing network I/O by removing remote connector

WebSphere Application Server Liberty Profile within CICS TS V5.2



z/OS Connect



1 z/OS Connect is software function that runs in Liberty Profile for z/OS.

2 z/OS Connect is described and configured in the Liberty `server.xml` file

3 z/OS Connect is designed to accept RESTful URIs with JSON data payloads

4 One part of z/OS Connect is a servlet that runs in Liberty Profile z/OS.

5 A 'Service Provider' is software that provides the connectivity to the backend system

6 z/OS Connect provides the ability to transform JSON to the layout required by backend

7 'Interceptors' are callout points where software can be invoked to do things such as SAF authorization and SMF activity recording

8 Initially the backend systems supported will be CICS, IMS and Batch

Java EE7, JSR-352 and Java Batch

JSR-352 Specification started in 2011 by IBM as spec lead

Based on programming models and experience from Compute Grid, z/OS Batch, and Spring Batch.

Specification developed through the JCP process by industry leaders in the batch space.

Approved by expert group and accepted as part of Java EE 7

Finally, a standard for the batch programming model!

The Java Batch Programming Model

Batchlets and Chunks

- Unmanaged and managed batch step processing
- Ability to stop, restart, checkpoint a job

Listeners

- Provide hook points into various lifecycle events of a job

Splits and Partitions

- Allow work to be parallelized

Restart, skip, and retry capabilities

- Allow for increased resiliency of a long running batch job

Job Specification Language

- XML based job definition with built in dependency injection

Why Java Batch and Liberty?

We have wrapped all the qualities of service of the Liberty profile around the batch programming model

- Dynamic configuration
- Operational management
- Transactions
- Logging
- High availability
- Scalability
- Tooling

More about Operational Management

REST API

- Java Batch in the Liberty profile provides an easy to use rest interface to remotely manage your batch jobs
- Ability to start, stop, restart, view job instance and execution data, and access job logs

Job Logging

- Server logs are interleaved with application records for easy debugging

External scheduler integration

- Provides the ability to combine enterprise quality scheduling with batch

Find the Match!