



**THINK GLOBAL – ACT LOCAL**



## Title: S511 zOS Java Batch Update

- Native Java on z/OS
- z/OS Batch Runtime
- WebSphere z/OS
- Java in CICS
- Java in IMS

Susann Thomas

SWG Client Technical Professional for zSoftware

Java, Java Batch, XML & WebSphere on System z

IBM Sales & Distribution, Software Sales

Susann.Thomas@de.ibm.com



## Copyright and Trademarks

© IBM Corporation 2013. All Rights Reserved.

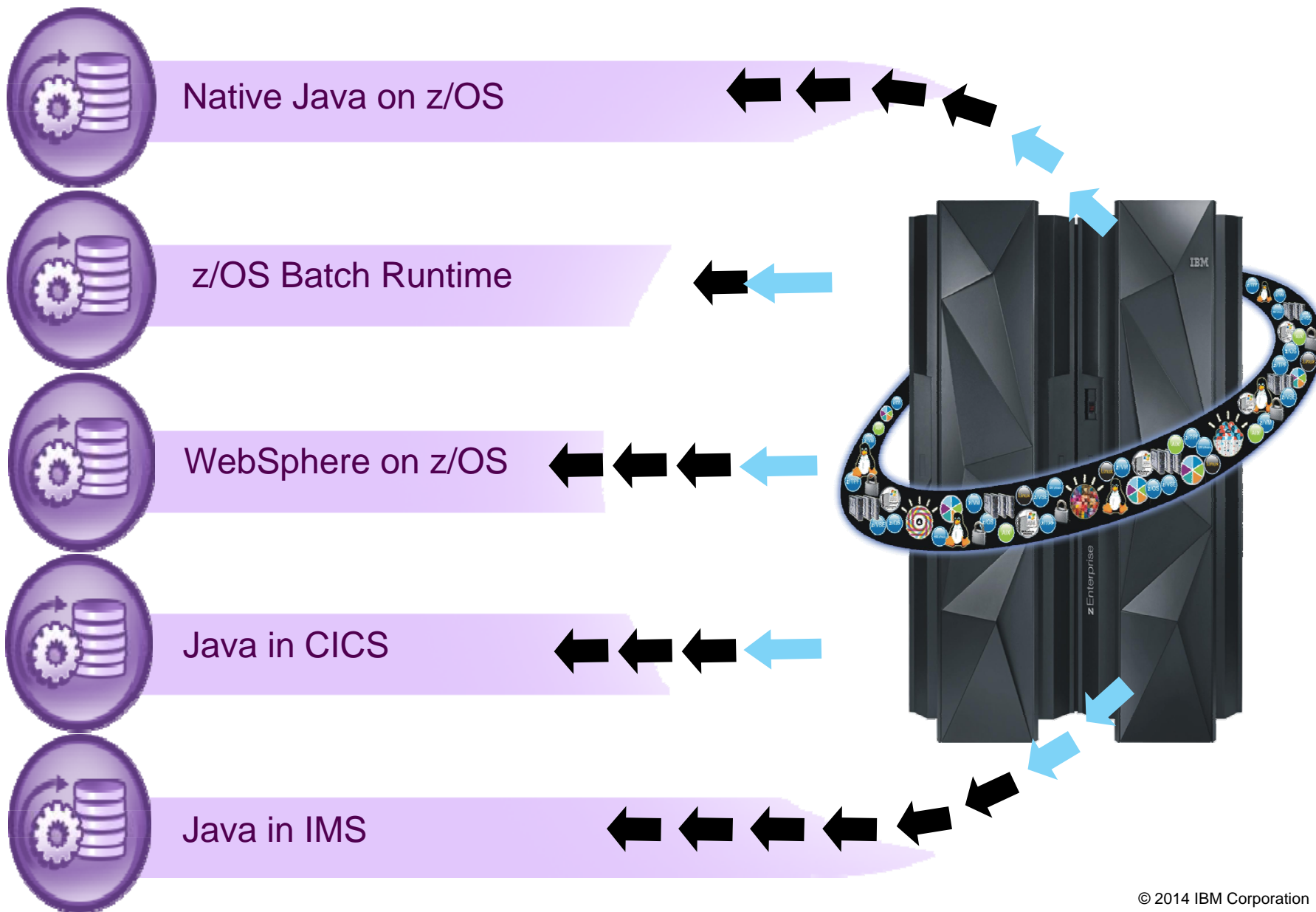
- IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).



# Legal Disclaimer

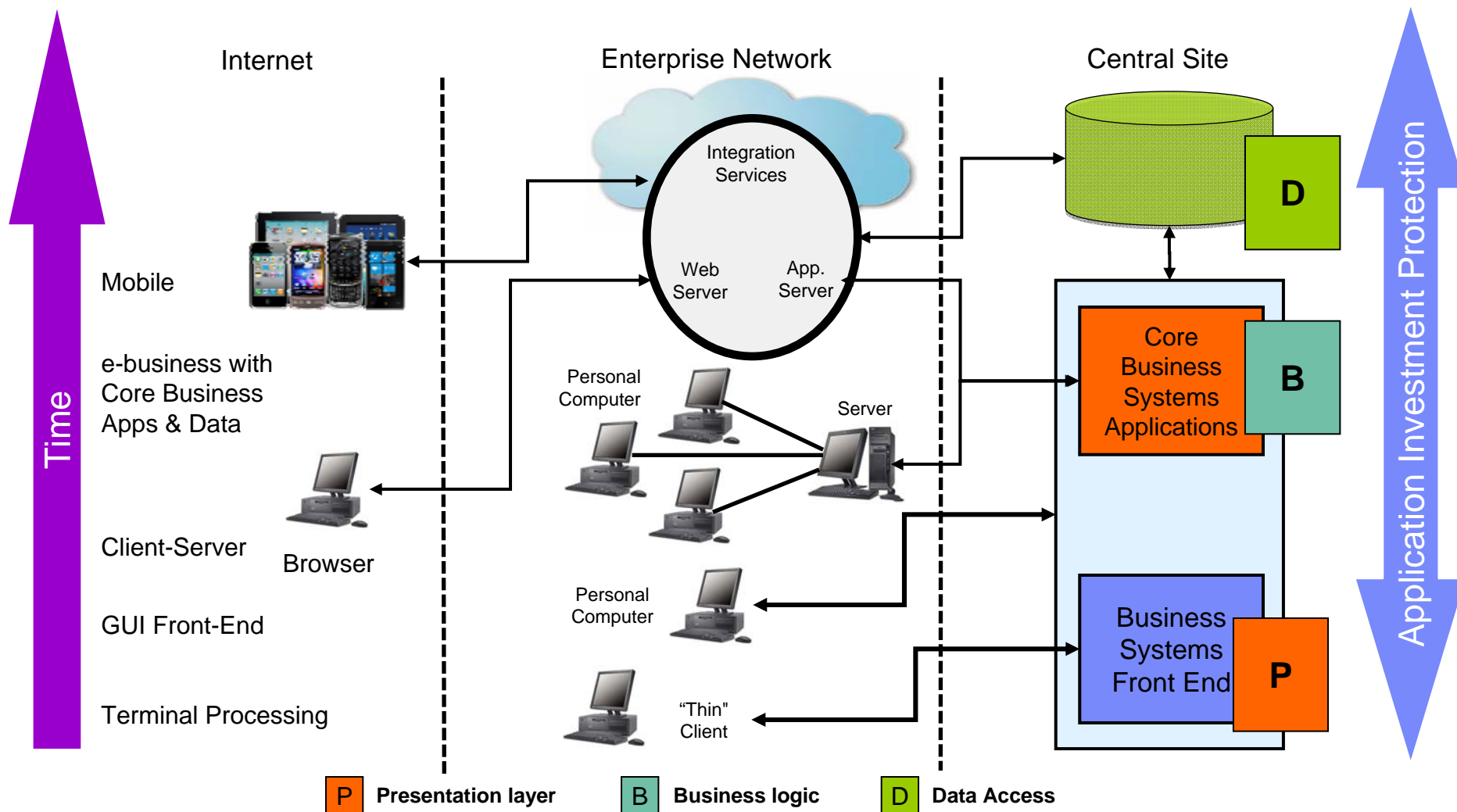
- © IBM Corporation 2013. All Rights Reserved.
- The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, 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 publication or any other materials. Nothing contained in this publication 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 IBM software.
- References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.
- If the text contains performance statistics or references to benchmarks, insert the following language; otherwise delete:  
Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including 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 results similar to those stated here.
- If the text includes any customer examples, please confirm we have prior written approval from such customer and insert the following language; otherwise delete:  
All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.
- Please review text for proper trademark attribution of IBM products. At first use, each product name must be the full name and include appropriate trademark symbols (e.g., IBM Lotus® Sametime® Unyte™). Subsequent references can drop "IBM" but should include the proper branding (e.g., Lotus Sametime Gateway, or WebSphere Application Server). Please refer to <http://www.ibm.com/legal/copytrade.shtml> for guidance on which trademarks require the ® or ™ symbol. Do not use abbreviations for IBM product names in your presentation. All product names must be used as adjectives rather than nouns. Please list all of the trademarks that you use in your presentation as follows; delete any not included in your presentation. IBM, the IBM logo, Lotus, Lotus Notes, Notes, Domino, Quickr, Sametime, WebSphere, UC2, PartnerWorld and Lotusphere are trademarks of International Business Machines Corporation in the United States, other countries, or both. Unyte is a trademark of WebDialogs, Inc., in the United States, other countries, or both.
- If you reference Adobe® in the text, please mark the first use and include the following; otherwise delete:  
Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- If you reference Java™ in the text, please mark the first use and include the following; otherwise delete:  
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- If you reference Microsoft® and/or Windows® in the text, please mark the first use and include the following, as applicable; otherwise delete:  
Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.
- If you reference Intel® and/or any of the following Intel products in the text, please mark the first use and include those that you use as follows; otherwise delete:  
Intel, Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- If you reference UNIX® in the text, please mark the first use and include the following; otherwise delete:  
UNIX is a registered trademark of The Open Group in the United States and other countries.
- If you reference Linux® in your presentation, please mark the first use and include the following; otherwise delete:  
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Other company, product, or service names may be trademarks or service marks of others.
- If the text/graphics include screenshots, no actual IBM employee names may be used (even your own), if your screenshots include fictitious company names (e.g., Renovations, Zeta Bank, Acme) please update and insert the following; otherwise delete: All references to [insert fictitious company name] refer to a fictitious company and are used for illustration purposes only.

# Java on zEnterprise



# Mainframe Assets – Why?

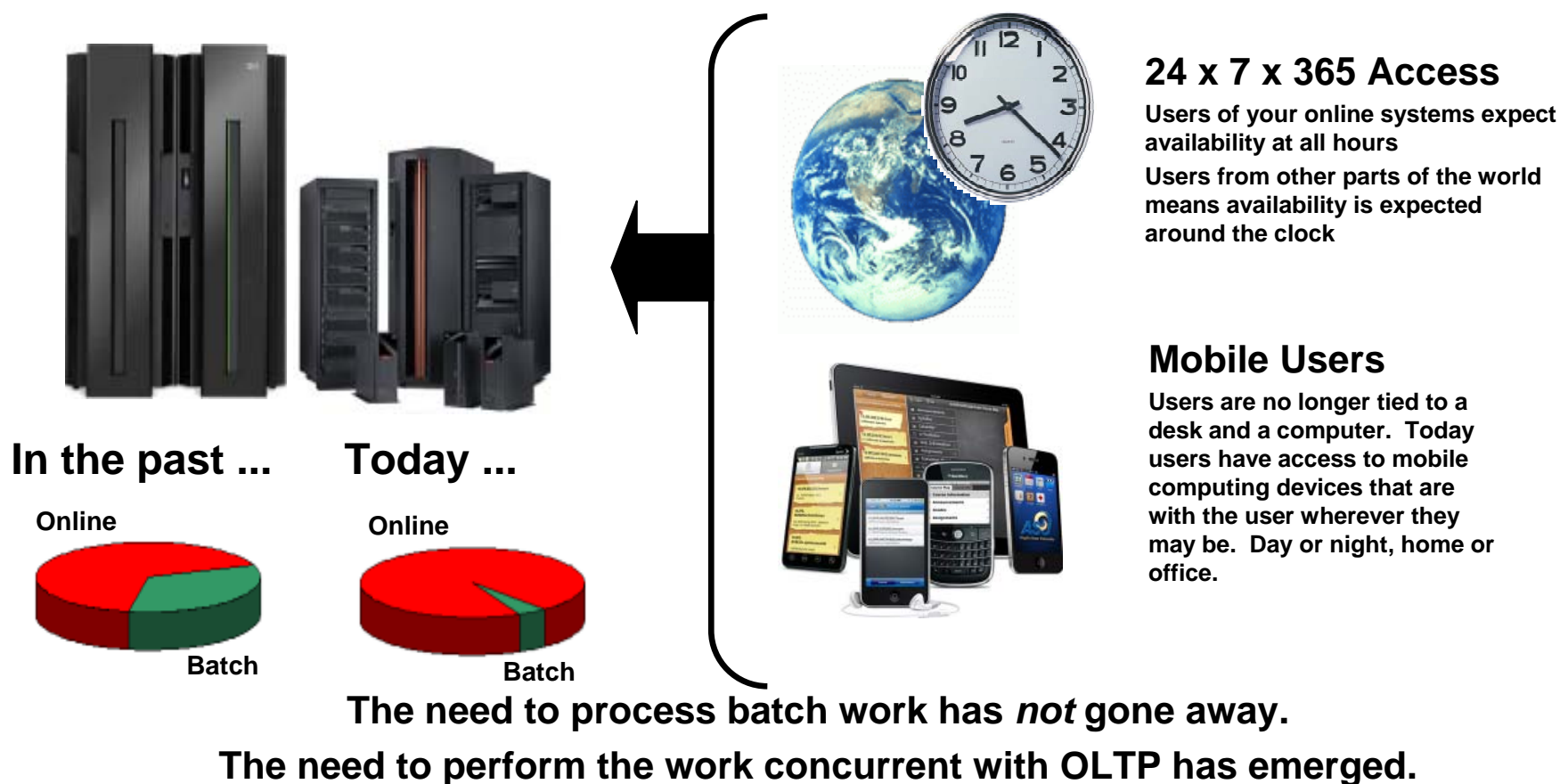
*(R)evolution in z/OS Core Middleware provides Unparalleled Scalability and Availability while protecting Investment*





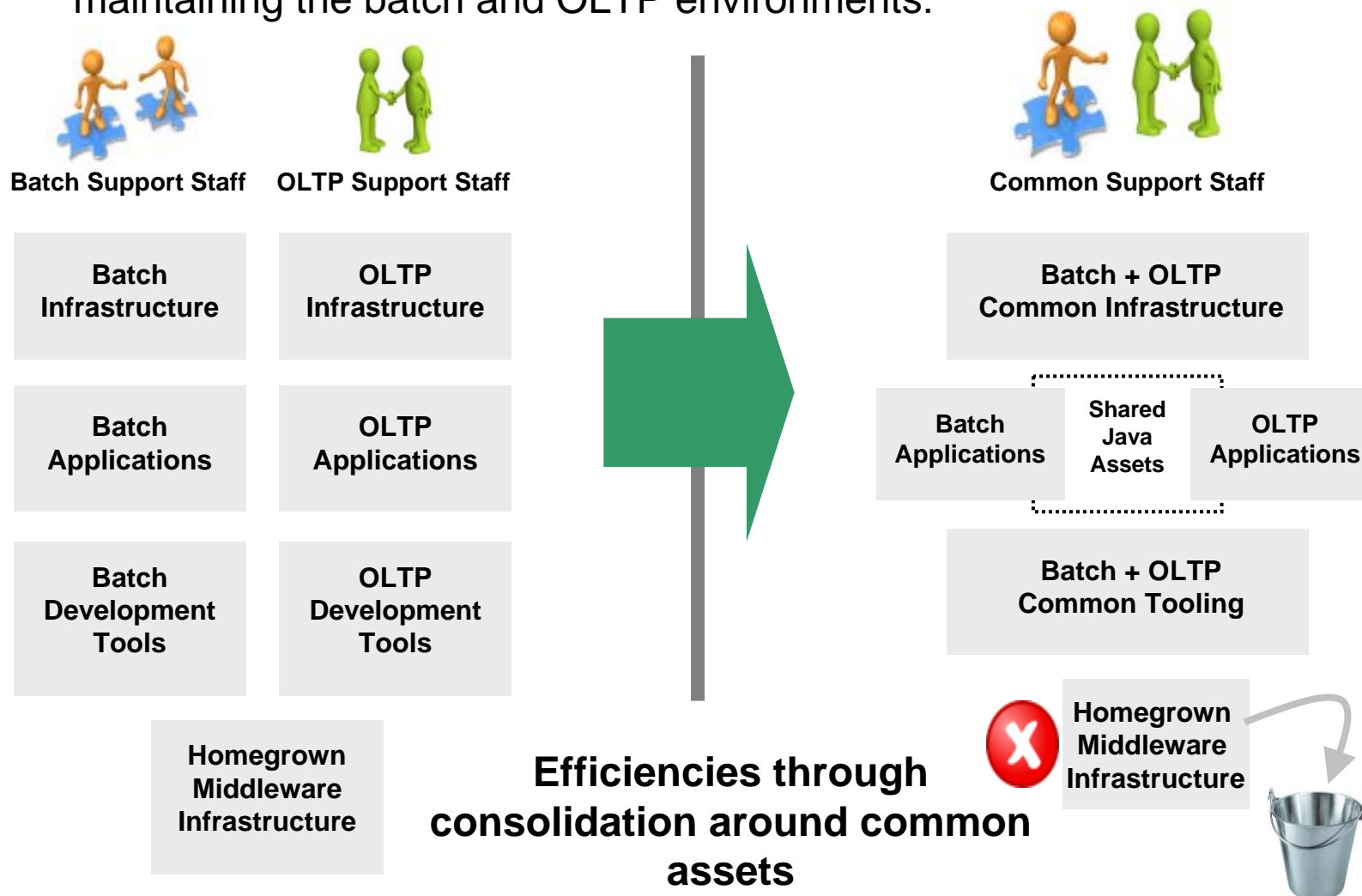
## Concept of "Dedicated Batch" Window Going Away

- Windows of time which used to be dedicated to batch processing are shrinking. The demands of online processing require more and more ...



## The Value of Shared Services

- It's not just that the window is shrinking ... it's also the cost pressures on maintaining the batch and OLTP environments:





## Java for Batch Processing?

- Yes ... for many very good reasons:

### Availability of Skills

Java is a programming language with wide adoption in the industry. Skills for Java programming are common and affordable.



### Tooling Support

Development tooling for Java has advanced to the point where some tools (IBM Rational Application Developer) are very powerful and sophisticated.

*This also provides an opportunity to consolidate to a common tooling environment for both OLTP and batch development.*

### z/OS Specialty Engines

Pressures on cost containment often dictate greater use of z/OS specialty engines. Java offloads to zAAP. Java batch does as well.

### Processing in OLTP Runtime

Running Java batch in the same execution runtime as Java OLTP provides an opportunity to mix and manage the two processing types together under the same management model.

**And don't forget development agility as well ...**

## "Agility"

- A word with several valuable contexts:

### **Business Agility**

- An application's ability to handle change without significant risk of introducing regressions

### **Developer Agility**

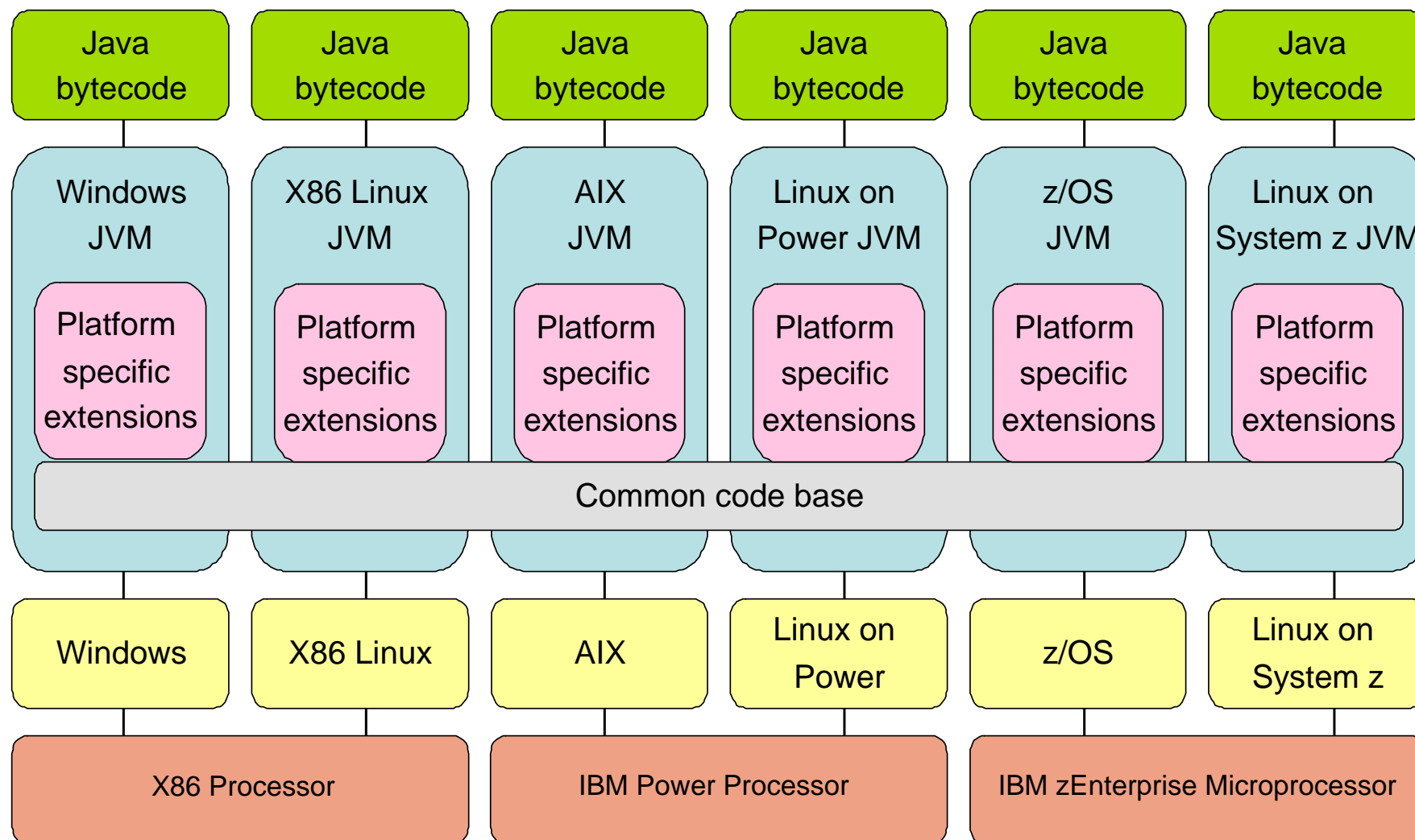
- The ease with which change can be implemented in an application, decreasing the development / test / GA cycle, moving changes into production more rapidly

### **Runtime Agility**

- The ease with which system programmers can identify and respond to errors when they occur.

# Why running Java Batch Applications on z/OS?

*Platform independency concept of the IBM Java Virtual Machine*



## zEnterprise EC12 – More Hardware for Java

**Continued aggressive investment in Java on Z**  
**Significant set of new hardware features tailored and co-designed with Java**

### **Hardware Transaction Memory (HTM)**

Better concurrency for multi-threaded applications  
 eg. ~2X improvement to `java.util.concurrent.ConcurrentLinkedQueue`

### **Run-time Instrumentation (RI)**

Innovation new h/w facility designed for managed runtimes  
 Enables new expanse of JRE optimizations

### **2GB page frames**

Improved performance targeting 64-bit heaps

### **Pageable 1MB large pages using flash**

Better versatility of managing memory

### **New software hints/directives**

Data usage intent improves cache management  
 Branch pre-load improves branch prediction

### **New trap instructions**

Reduce over-head of implicit bounds/null checks

New **5.5 GHz** 6-Core Processor Chip  
**Large caches** to optimize data serving  
 Second generation **OOO design**

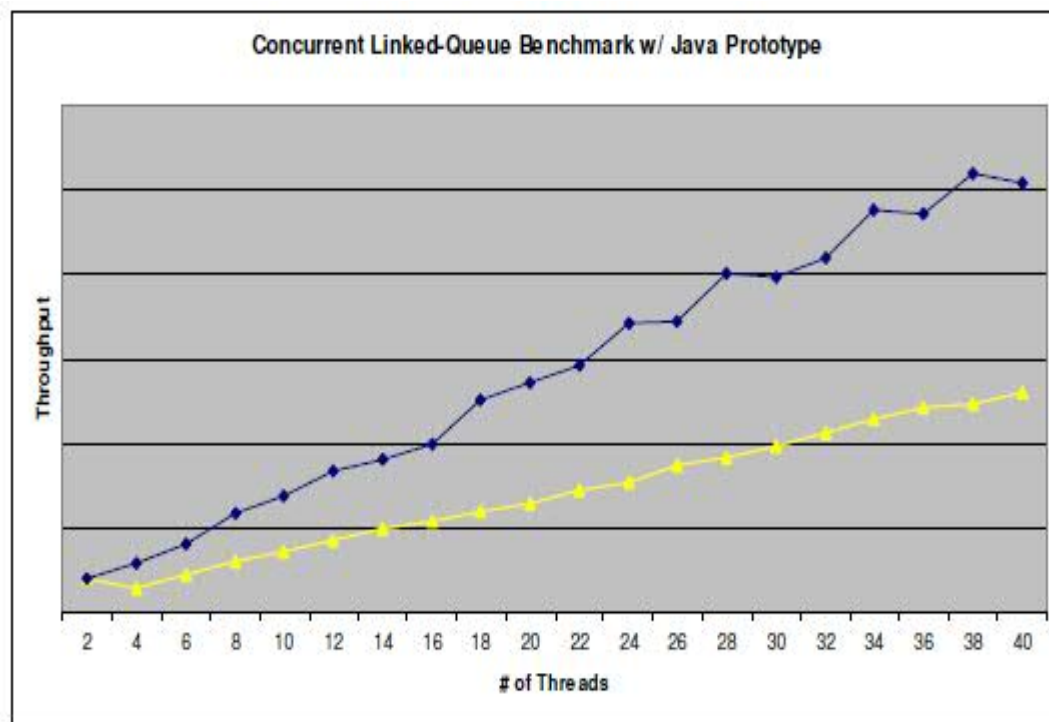


Engineered Together—IBM Java and zEC12 Boost Workload Performance  
[http://www.ibm.systemsmag.com/mainframe/trends/whatsnew/java\\_compiler/](http://www.ibm.systemsmag.com/mainframe/trends/whatsnew/java_compiler/)

Up-to 60% improvement in throughput amongst Java workloads  
 measured with zEC12 and Java7SR3

# Transactional Execution: Concurrent Linked Queue

- *~2x improved scalability of `juc.ConcurrentLinkedQueue`*
- *Unbound Thread-Safe `LinkedList`*
  - First-in-first-out (FIFO)
    - Insert elements into tail (en-queue)
    - Poll elements from head (de-queue)
  - No explicit locking required
- *Example usage: a multi-threaded work queue*
  - Tasks are inserted into a concurrent linked queue as multiple worker threads poll work from it concurrently



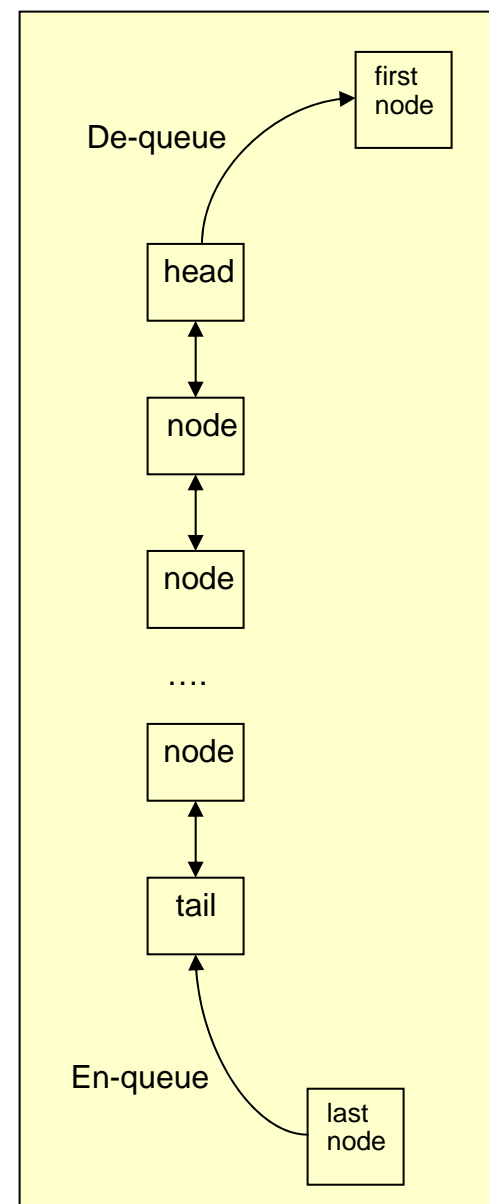
**New TX-base  
implementation**



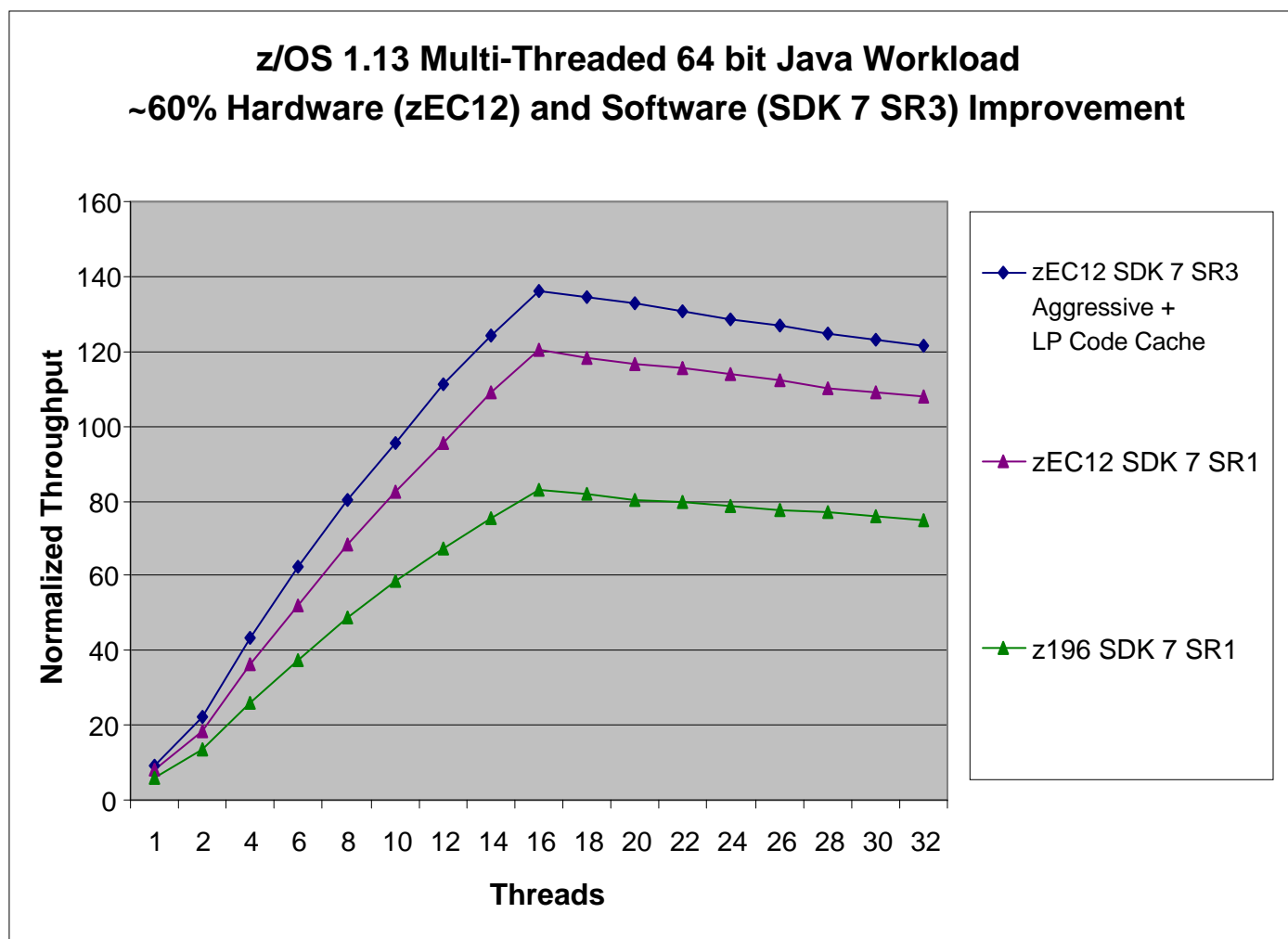
**Traditional CAS-base  
implementation**

(Controlled measurement environment, results may vary)

© 2014 IBM Corporation



## z/OS Java SDK 7: 64-bit Java Multi-threaded Benchmark on 16-Way

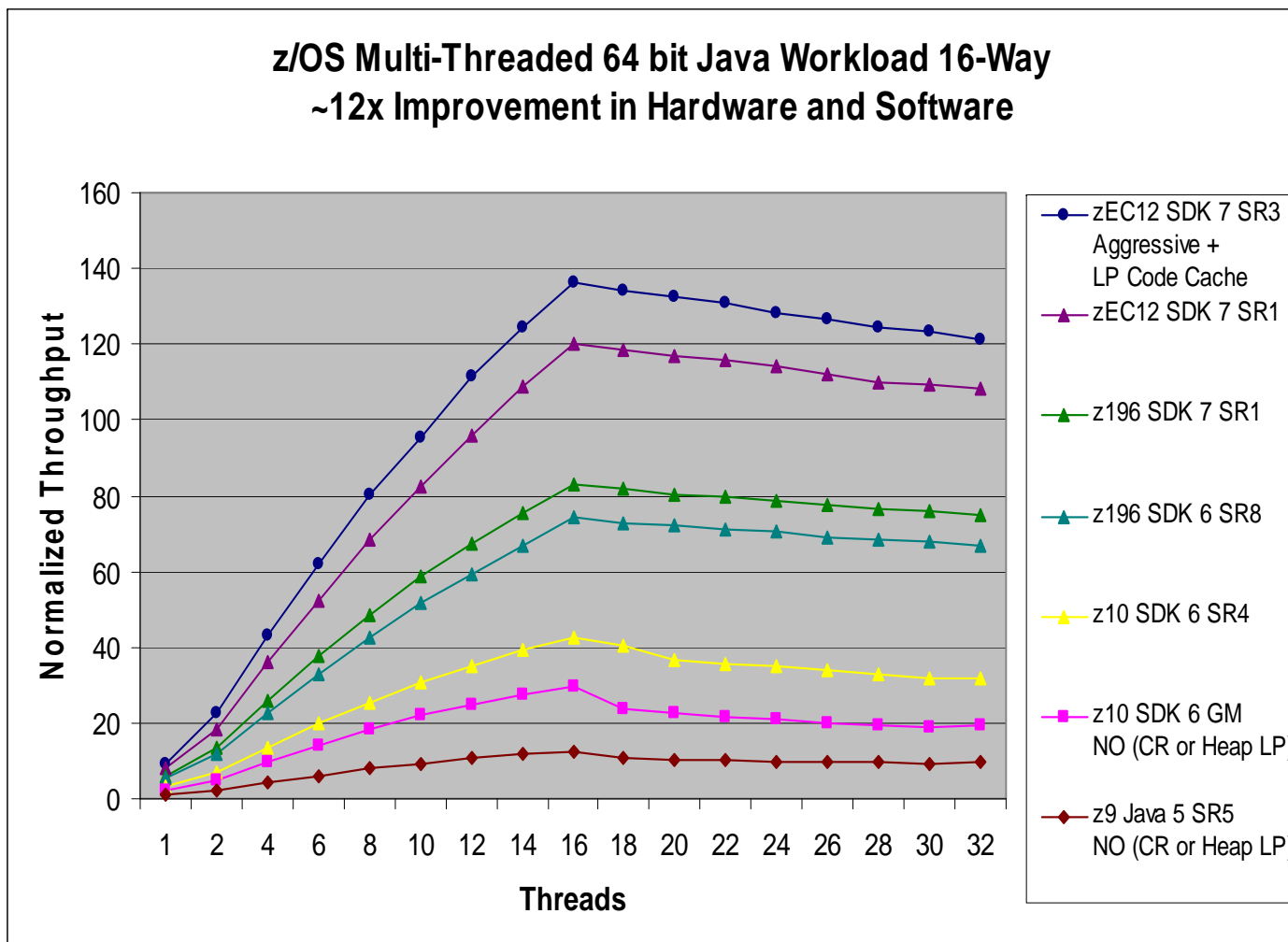


### Aggregate 60% improvement from zEC12 and Java7SR3

- zEC12 offers a ~45% improvement over z196 running the Java Multi-Threaded Benchmark
- Java7SR3 offers an additional ~13% improvement (-Xaggressive + Flash Express pageable 1Meg large pages)
- (Controlled measurement environment, results may vary)



## Aggregate HW and SDK Improvement z9 Java 5 SR5 to zEC12 Java 7SR3



~12x aggregate hardware and software improvement comparing Java5SR5 on z9 to Java7SR3 on zEC12

- LP=Large Pages for Java heap CR= Java compressed references
- Java7SR3 using -Xaggressive + Flash Express pageable 1Meg large pages
- (Controlled measurement environment, results may vary)

## New and existing supported Java products – z/OS

### Latest News:

- **Java Technology Edition, Version 7 Release 1**
  - IBM 31-bit SDK, for z/OS, Build Level: December 13, 2013
  - IBM 64-bit SDK, for z/OS, Build Level: December 13, 2013

### Earlier Deliveries

- Java Technology Edition, V7.0.0
  - IBM 31-bit SDK for z/OS, Build Level: January 08, 2014 - SDK7 SR6 FP1 (PTF UI14490 / APARs PI08841 and PI08842)
  - IBM 64-bit SDK for z/OS, Build Level: January 08, 2014 - SDK7 SR6 FP1, (PTF UI14491 / APARs PI08843 and PI08844 )
- Java Technology Edition, V6.0.1
  - IBM 31-bit SDK for z/OS, Build Level: January 10, 2014 - SDK601 SR7 FP1 (PTF UI14484 / APARs PI08836 and PI08837)
  - IBM 64-bit SDK for z/OS, Build Level: January 10, 2014 - SDK601 SR7 FP1 (PTF UI14485/ APARs PI08839 and PI08840)
- Java Technology Edition, V6.0.0
  - IBM 31-bit SDK for z/OS, Build Level: January 10, 2014 - SDK6 SR15 FP1 (PTF UI14482 / APARs PI08831 and PI08832)
  - IBM 64-bit SDK for z/OS, Build Level: January 10, 2014 - SDK6 SR15 FP1 (PTF UI14483/ APARs PI08833 and PI08834)
- Java 2 Technology Edition, V5
  - Effective September 30, 2013, neither IBM 31-bit SDK for z/OS Java 2 Technology Edition, V5 (5655-N98) nor IBM 64-bit SDK for z/OS Java 2 Technology Edition, V5 (5655-N99) is no longer in support.
  - IBM 31-bit SDK for z/OS, Build Level: July 2, 2013 - SDK5 SR16 FP3 (PTF UK95622 / APARs PM92062 and PM92063)
  - IBM 64-bit SDK for z/OS, Build Level: July 2, 2013 - SDK5 SR16 FP3 (PTF UK95623 / APARs PM92064 and PM92065)
- All products are delivered via the z/OS Java website in non-SMP/E format and via ShopIBM in SMP/E format
- All products are independently orderable and serviceable and follow the z/OS RFA rules for Withdrawal from Marketing and End of Service

## IBM and Java



- **Java is critically important to IBM**
  - Fundamental infrastructure for IBM's software portfolio
  - WebSphere, Lotus, Tivoli, Rational, Information Management (IM), PD Tools
- **IBM is investing strategically for Java in virtual machines**
  - As of Java 5.0, single JVM support, JME, JSE, JEE, J9/TR Compiler
- **IBM is massively investing System z HW optimization, integration & exploitation for Java**
  - zAAP, zAAP on zIIP, Accelerators (zEDC, Crypto, Flash etc), JIT, System Structure & Instructions, HW assisted profiling, Large Pages, Transactional Execution
- **IBM is investing in JAVA z/OS Platform integration and enablement**
  - IMS, CICS, DB2, RACF, WLM, JZOS, z/OS WebSphere, zOSMF, z/OS Batch Runtime, COBOL/PL1-Java integration strategy, WOLA, Batch etc
- **IBM also invests in, and supports public innovation in Java**
  - OpenJDK, Eclipse, Apache (XML, Aries, Derby, Geronimo, Harmony, Hadoop ...)
  - Broad participation in relevant open standards (JCP, OSGi)

## Evolving Java as a Workload Optimized System (WOS) on Z

***Enable integration of Java-based applications with core z/OS backend database environment for high performance, reliability, availability, security, and lower total cost of ownership.***

### Portable and consumable

- First-class IBM Java SDK for z/OS and z/Linux
- Providing seamless portability across platforms

### Pervasive and integrated across the z eco-system

- Java business logic runs with all z middleware (IMS, CICS, WAS etc)
- Inter-operability with legacy batch and OLTP assets

### Deep System Z exploitation

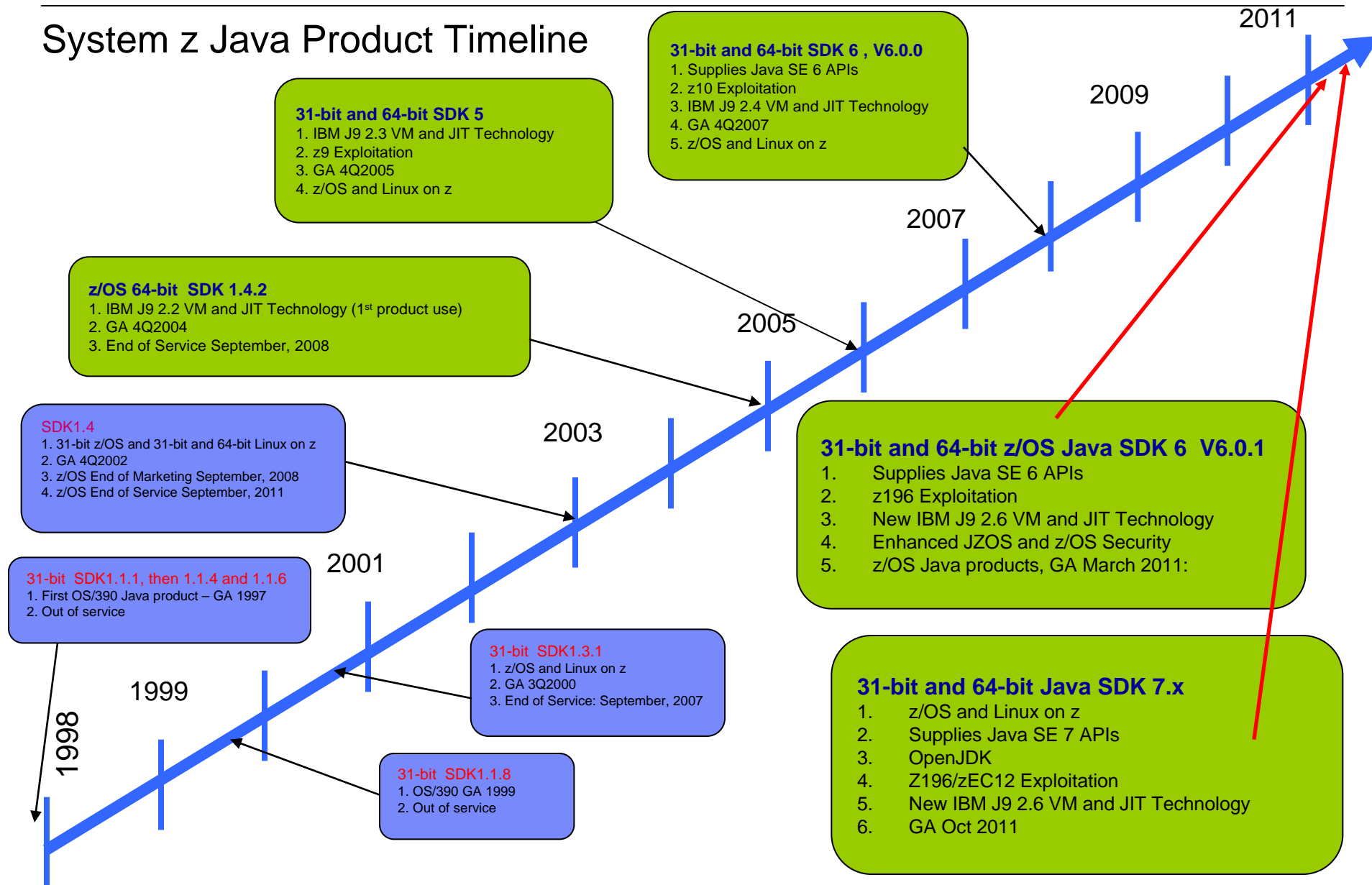
- SDK extensions enabled z QoS for full integration with z/OS
- zAAP/zIIP specialty engines provide low-cost Java capacity

### Performance

- A decade of hardware/software innovation and optimization
- Industry leading performance with IBM J9 Virtual Machine
- Enabling tight data locality for high-performance and simplified systems



# System z Java Product Timeline



IBM continues to invest aggressively in Java for System z, demonstrating a rich history of innovation and performance improvements.

# Java Road Map

\*\*Timelines and deliveries are subject to change.

## Java 5.0

New Language features:

- Autoboxing
- Enumerated types
- Generics
- Metadata

## Java 6.0

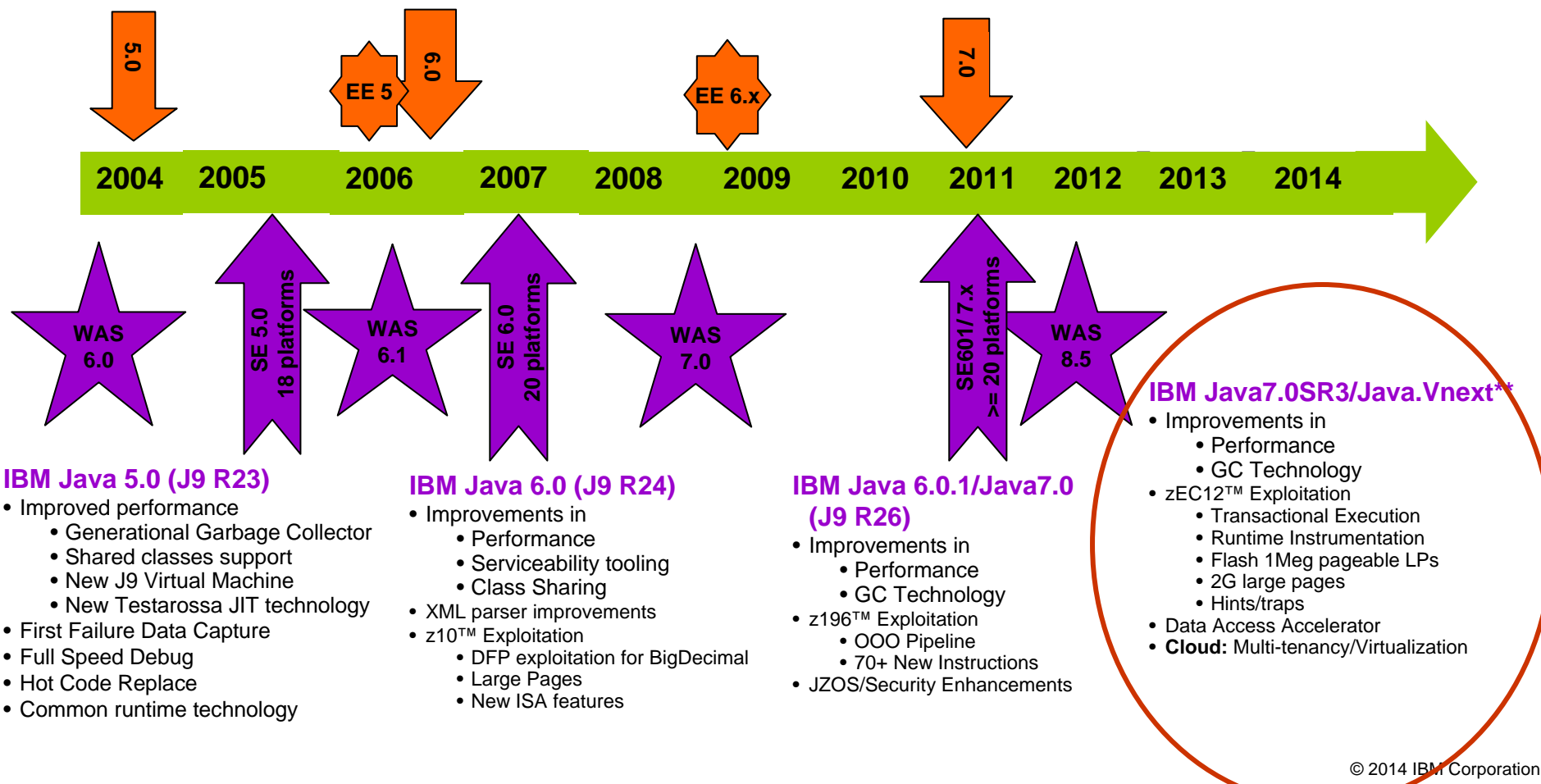
- Performance Improvements
- Client WebServices Support

## Java 7.0

- Support for dynamic languages
- Improve ease of use for SWING
- New IO APIs (NIO2)
- Java persistence API
- JMX 2.x and WS connection for JMX agents
- Language Changes

## Java 8.0\*\*

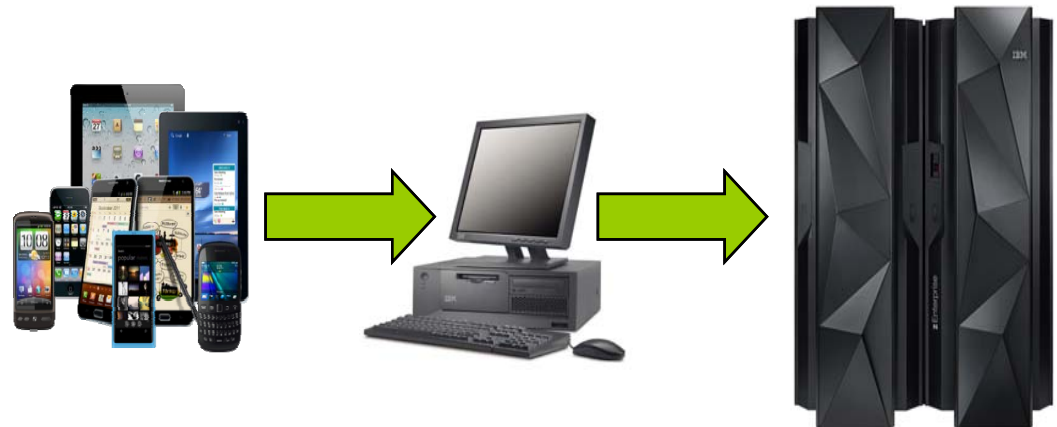
- Language improvements
- Closures for simplified fork/join



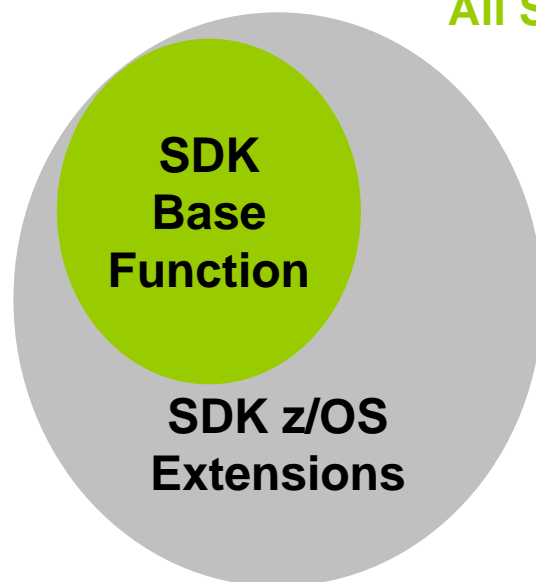


## IBM Java Runtime Environment

- IBM's implementation of Java 5 and Java 6 are built with **IBM J9 Virtual Machine and IBM Testarossa JIT Compiler** technology
  - Independent clean-room JVM runtime & JIT compiler
- Combines best-of breed from embedded, development and server environments... from a cell-phone to a mainframe!
  - Lightweight flexible/scalable technology
  - World class garbage collection – gencon, balanced GC policies
  - Startup & Footprint - Shared classes, Ahead-of-time (AOT) compilation
  - 64-bit performance - Compressed references & Large Pages
  - Deep System z exploitation – z196/z10/z9/z990 exploitation
  - Cost-effective for z - zAAP Ready!
- Millions of instances of J9/TR compiler



## z/OS – System z Java Extensions



### All SDKs support the 'standards', Java on z/OS extends the SDK

- Access to z/OS services
- Access to all types of data
- Access under control of z/OS security mechanisms
- Integration into existing operational infrastructure

Services available in JEE and JSE environments under the restrictions of the container.

### System specific extension allow you to write robust middleware and applications that integrate with traditional z/OS operating environment

- Allow for maintaining platform independent design development.
- Platform specific implementations when required
- Allows for operational and resource optimization

e.g. JAAS wrapper of SAF (RACF, ACF2, or TopSecret), Traditional OS dataset access, Cryptographic hardware (Cards and CPACF), z/OS Console (modify and messages), z/OS system logger, JES job submission, DFSORT, SMF, etc.

# Java Execution Environments and Interoperability

***Capitalize on pre-existing assets, artifacts, processes, core competencies, platform strengths***

## IBM Java Execution Offerings

- Transactional/Interactive
  - WebSphere for z/OS (WAS z/OS)
  - WebSphere Process Server for z/OS (WPS)
  - JCICS
  - IMS Java
  - DB2 Stored Procedures
  
- Batch oriented
  - WebSphere Batch (WAS-CG)  
WAS/JEE runtime extensions
  - IMS Java Batch regions (JMP)
  - JZOS component of z/OS SDK  
JES/JSE-based environment
  - Since z/OS V1R13 Java/COBOL Batch Runtime Env.\*  
JES/JSE-based, designed to inter-op with DB2 while maintaining transaction integrity

## Open Source or non-IBM vendor Application Server and Frameworks

- Tomcat, JBoss
- iBatis, Hibernate, Spring
- Ant

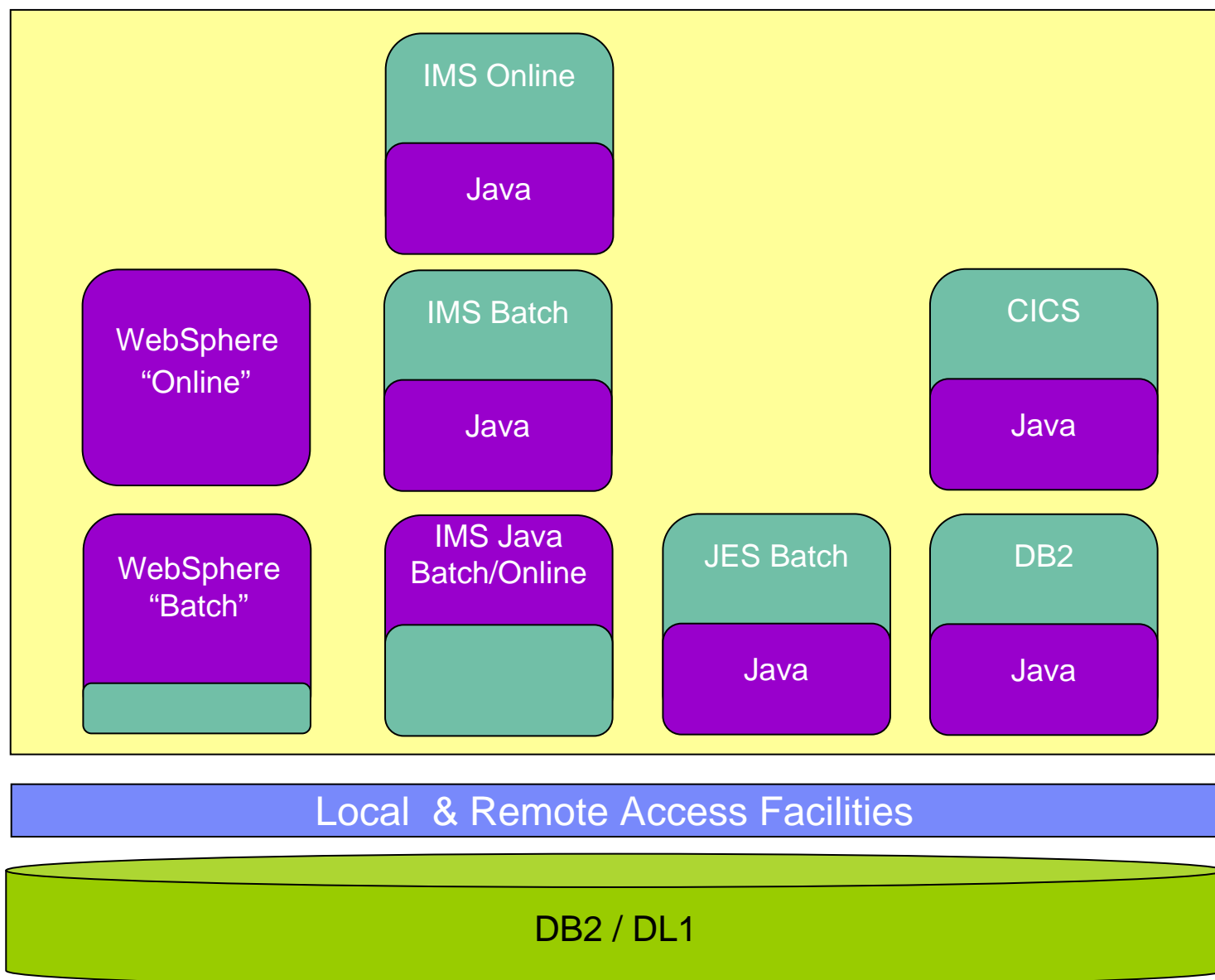
## COBOL/Native Interoperability

- COBOL Invoke maps to JNI
- RDz and JZOS\*\* have tooling to map COBOL copy books to Java classes
- JCICS
- IMS Java
- WebSphere Cobol Container, WOLA
- Etc.

\* See <http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=ca&infotype=an&supplier=897&letternum=ENUS211-252>

\*\* Alphaworks only, and hence currently un-supported

## Java on z/OS supports all classic z/OS Middleware environments



## IBM proposes two Java Batch Models on z/OS

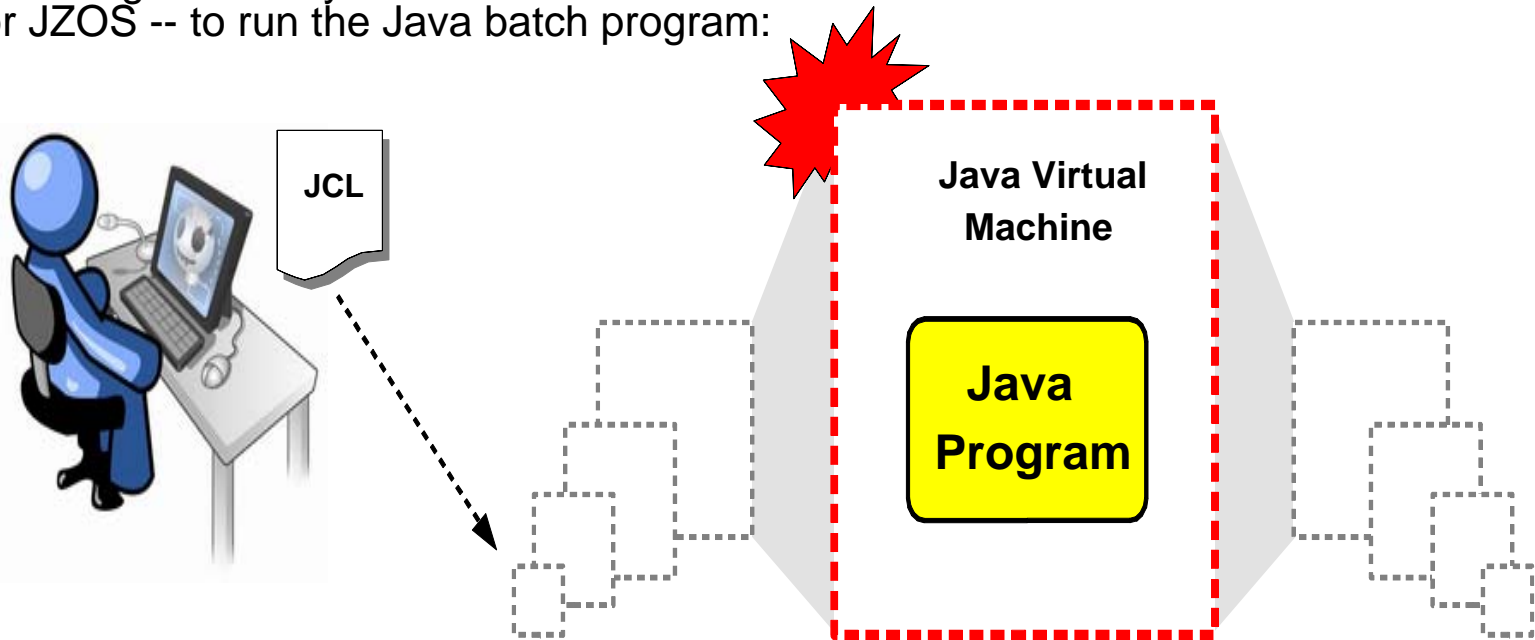
- Tools that instantiate a JVM and invoke the Java program
  - JVM terminates at completion of batch program
  - Provide a degree of batch support
  - Examples: BPXBATCH or JZOS
- 
- Provides a set of batch functionality in the form of supporting class libraries and development libraries
  - Specific integration with underlying middleware or platform
  - Examples: WAS Feature Pack for Modern Batch, WebSphere Batch aka WebSphere Compute Grid

**JVM Launcher**

**Execution  
Platform**

## Java stand-alone Batch

- The first thought of many is to use a JVM launcher – Java command line interface (CLI) or JZOS -- to run the Java batch program:



- This works, but it has its limitations:
  - There is overhead associated with repeated cycling of JVMs
  - Much is left to your application -- there are few batch services provided
- JZOS addressed a few, but not all ... and that is the next topic to touch on



## What is JZOS and how to get it?

- A launcher for the z/OS JVM

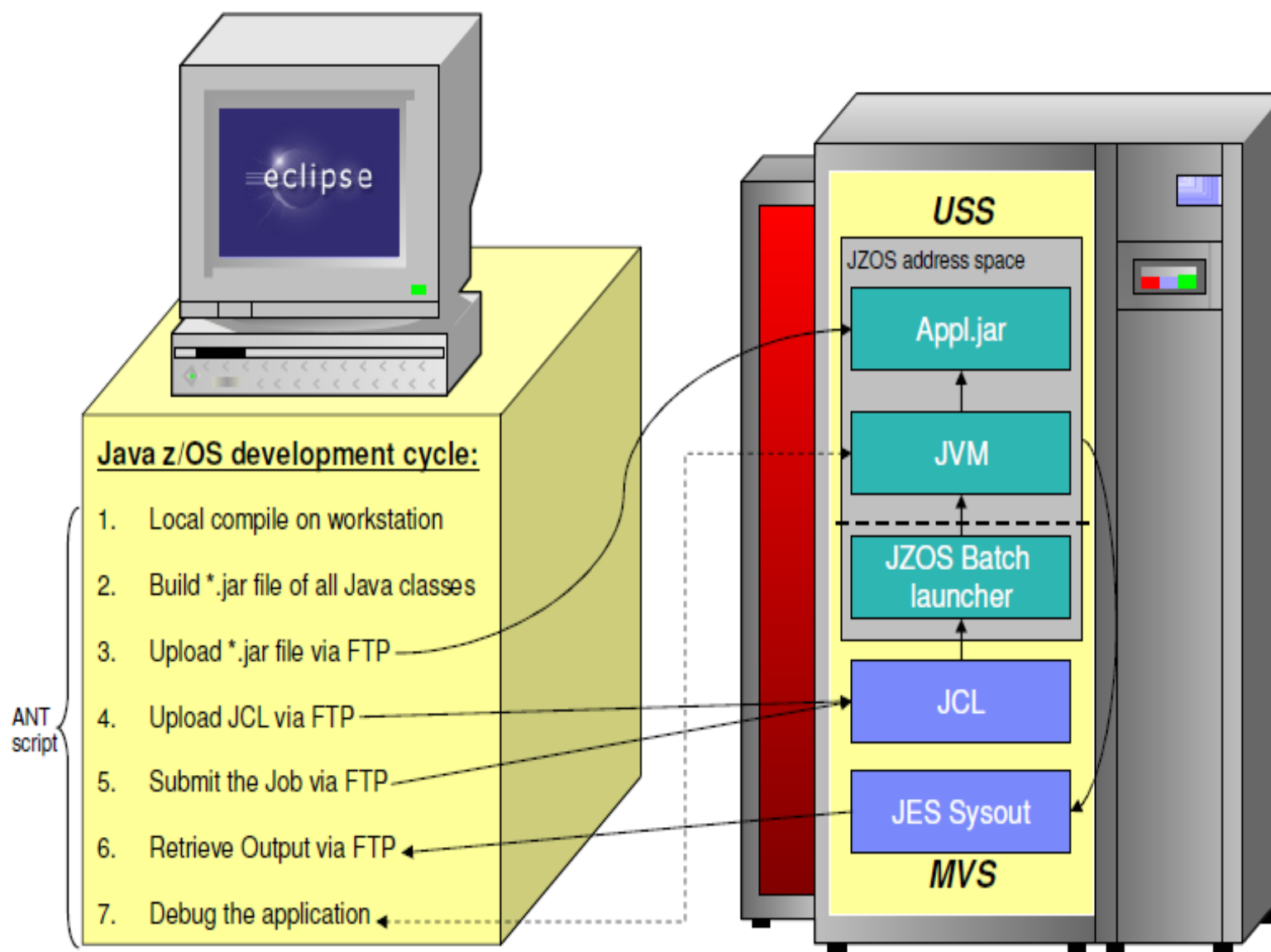


- A toolkit to use z/OS system functions from Java



- Included with all recent Java SDKs on z/OS (New versions on alphaWorks)

## Java Batch Framework on z/OS: JZOS



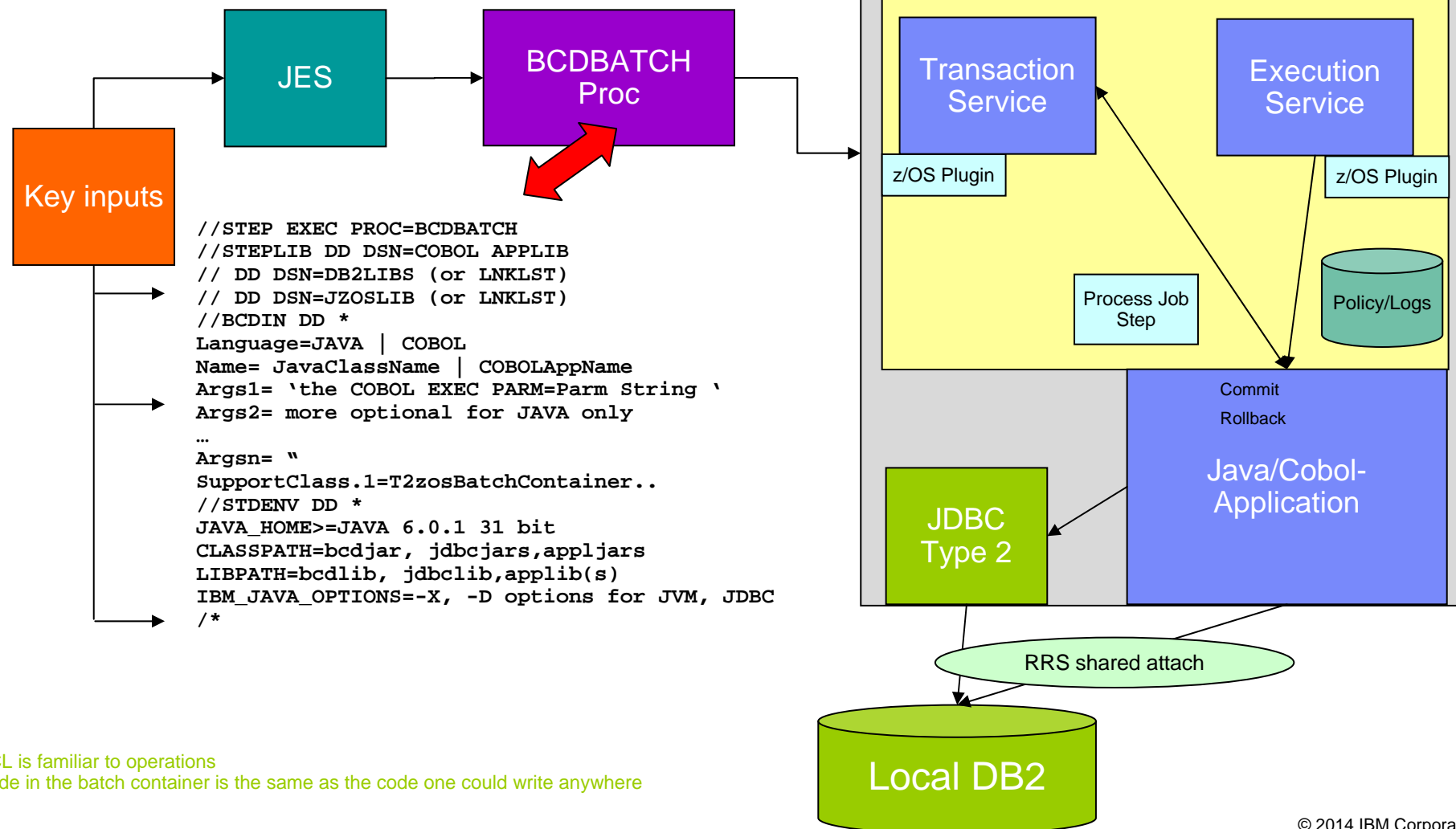
## JCL sample of a JZOS job

```
//ZUSER30A JOB
//PROCLIB JCLLIB ORDER=SYS1.PROCLIB
//JAVA EXEC PROC=JVMPRC14,
// JAVACLS='com.ibm.bank.sample.Main'
//OUTPUT DD SYSOUT=*
//STDENV DD *
# This is a shell script which configures
# any environment variables for the Java JVM.
# Variables must be exported to be seen by the launcher.
. /etc/profile
export APPL_HOME=/u/zuser30/myjava
export JAVA_HOME=/usr/lpp/java/J1.4
export PATH="$PATH":"${JAVA_HOME}"/bin:
LIBPATH="$LIBPATH":"${JAVA_HOME}"/bin
LIBPATH="$LIBPATH":"${JAVA_HOME}"/bin/classic
LIBPATH="$LIBPATH":"${JZOS_HOME}"
export LIBPATH="$LIBPATH":
# Customize your CLASSPATH here
# Add application home directory and jars to CLASSPATH
for i in "${APPL_HOME}"/*.jar; do
CLASSPATH="$CLASSPATH":"$i"
done
export CLASSPATH="$CLASSPATH":
```

```
# Set JZOS specific options
# Use this variable to specify encoding for DD STDOUT and
STDERR
#export JZOS_OUTPUT_ENCODING=Cp1047
# Use this variable to prevent JZOS from handling MVS
operator
# commands
#export JZOS_ENABLE_MVS_COMMANDS=false
# Use this variable to supply additional arguments to main
#export JZOS_MAIN_ARGS=""
# Configure JVM options
IJO="-Xms16m -Xmx128m"
# Uncomment the following line if you want to debug the
application
#IJO="$IJO -Xdebug
-Xrunjdwp:transport=dt_socket,server=y,address=8000"
IJO="$IJO -Djzos.home=${JZOS_HOME}"
# Uncomment the following if you want to run without JIT
#IJO="$IJO -Djava.compiler=NONE"
# Uncomment the following if you want to run with Ascii
file
encoding..
#IJO="$IJO -Dfile.encoding=ISO8859-1"
export IBM_JAVA_OPTIONS="$IJO "
export JAVA_DUMP_HEAP=false
export JAVA_PROPAGATE=NO
export IBM_JAVA_ZOS_TDUMP=NO
//
```

## z/OS Batch Runtime – High-level overview of the z/OS Batch Runtime

- The batch container performs the initialization that sets up the environment for COBOL, Java, and DB2 interoperability.



JCL is familiar to operations  
code in the batch container is the same as the code one could write anywhere

## Requirements for z/OS Batch Runtime

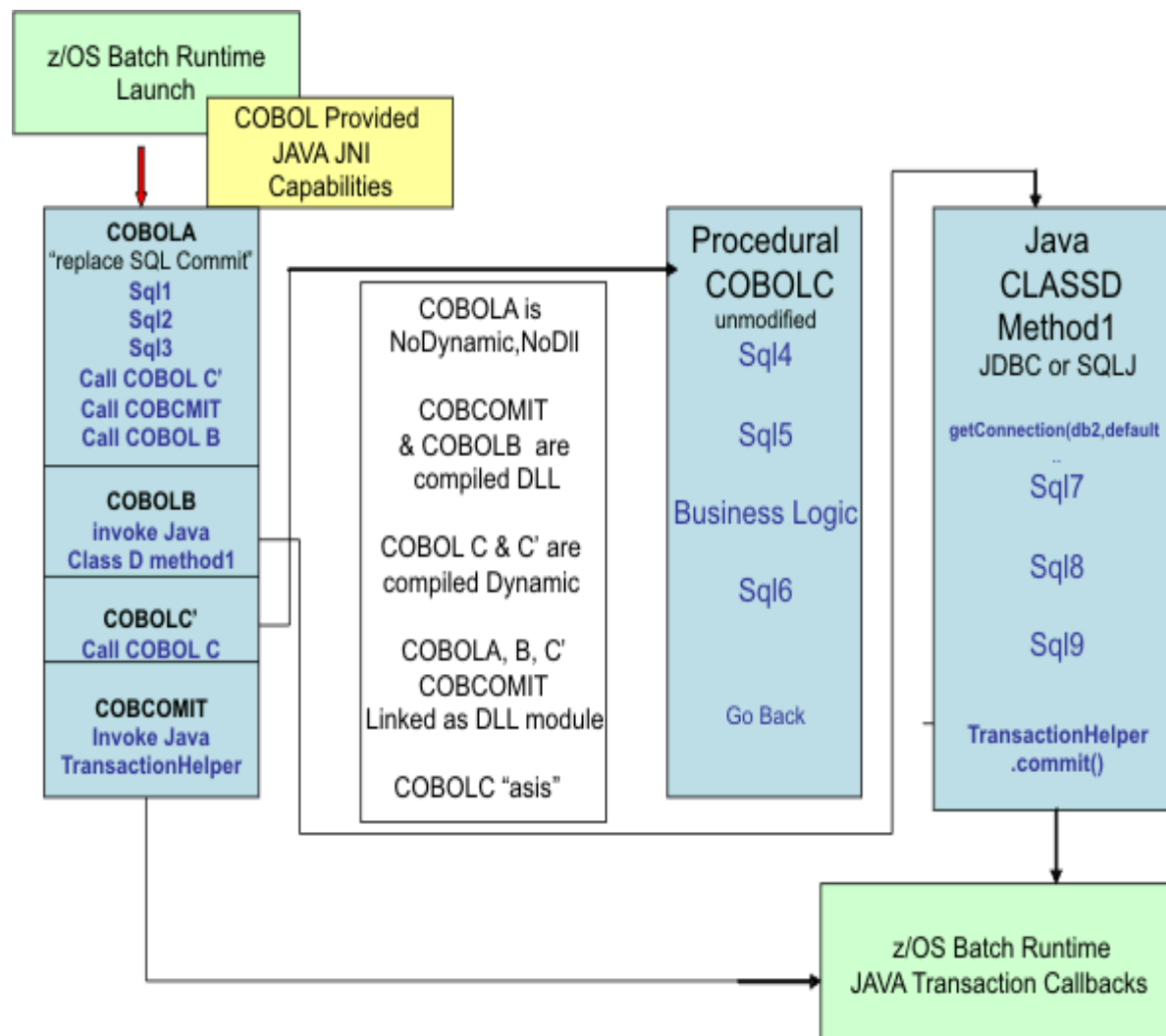
- z/OS® Batch Runtime requires the following programs:
  - IBM 31-bit SDK for z/OS, Java™ Technology Edition, V6 (5655-R31)
  - Enterprise COBOL Version 4.2 and above
  - One of the following:
    - DB2 V9 with PTF UK62190 for JDBC 3.0 specification level, or PTF UK62191 for JDBC 3.0 specification level
    - DB2 V10 with PTF UK62141 for JDBC 3.0 specification level, or PTF UK62145 for JDBC 4.0 specification level
- References:
  - z/OS Batch Runtime Planning and User's Guide (SA23-2270-00)  
<http://publibz.boulder.ibm.com/epubs/pdf/iea2n511.pdf>
  - For complete information about Enterprise COBOL Version 4 Release 2, see  
[www.ibm.com/software/awdtools/cobol/zos/library/](http://www.ibm.com/software/awdtools/cobol/zos/library/)



## z/OS Batch Runtime

*Example: COBOL program calling Java and unmodified COBOL ments for z/OS Batch Runtime*

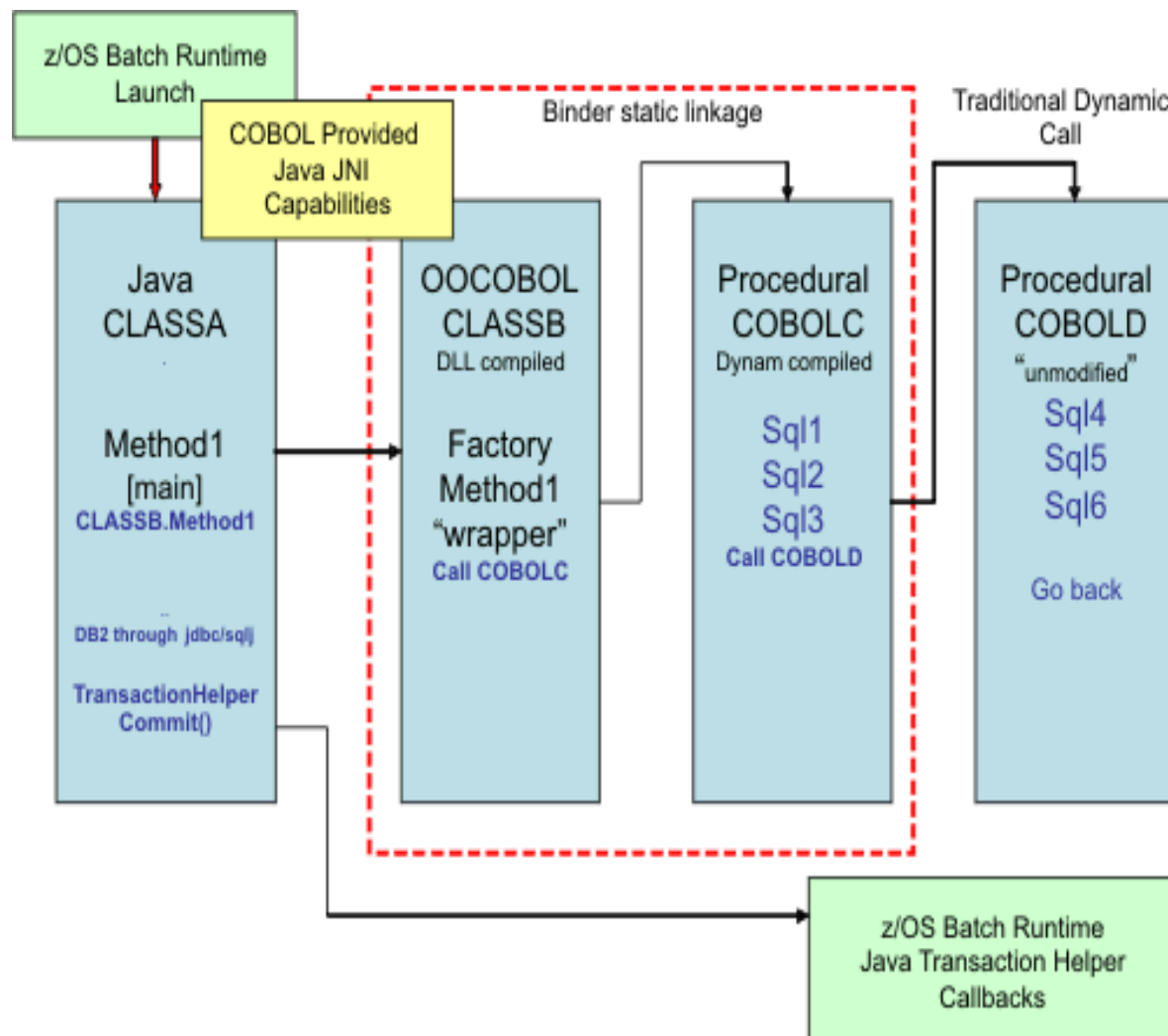
- In this example, the program flow starts in COBOL and then flows to a Java program and to another COBOL program. OOCOBOL methods are not used; however, the programs use both COBOL JNI and user JNI.



## z/OS Batch Runtime

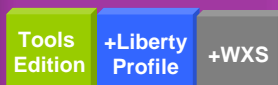
*Example: Java program using OOCOBOL to call COBOL*

- A Java program flows to a COBOL program.
- In this example, the Java program uses an OOCOBOL factory wrapper to call COBOL.



# WebSphere Application Server Family

## WAS for Developers



Enables efficient development of innovative apps that will run on WAS in production

Available as a no-charge edition for the developer desktop and includes Eclipse adapters

Provide WAS and WDT editions as freely available for dev desktops and supported under production runtime licenses

## WAS Hypervisor Edition



Optimized to instantly run in VMware, PowerVM, zVM and other server virtualization environments.

## WAS ND



Delivers near-continuous availability, with advanced performance and intelligent management capabilities, for mission-critical apps. **Full entitlement to eXtreme Scale.**

## WAS for z/OS



Takes full advantage of the z/OS Sysplex to deliver a highly secure, reliable, and resource efficient server experience. **Entitlement to eXtreme Scale z/OS client.**

## WAS



Provides secure, high performance transaction engine for moderately sized configurations with web tier clustering and failover across up to five application server profiles. **Includes entitlement to eXtreme Scale for session caching and DynaCache.**

## WAS Liberty Core

**New!**

Liberty Profile Only  
(Web Profile)

A lightweight and low-cost Liberty profile based offering (not full-profile WAS), providing the capabilities to rapidly build and deliver web apps that do not require the full Java EE stack.

## WAS Express

+Liberty  
Profile

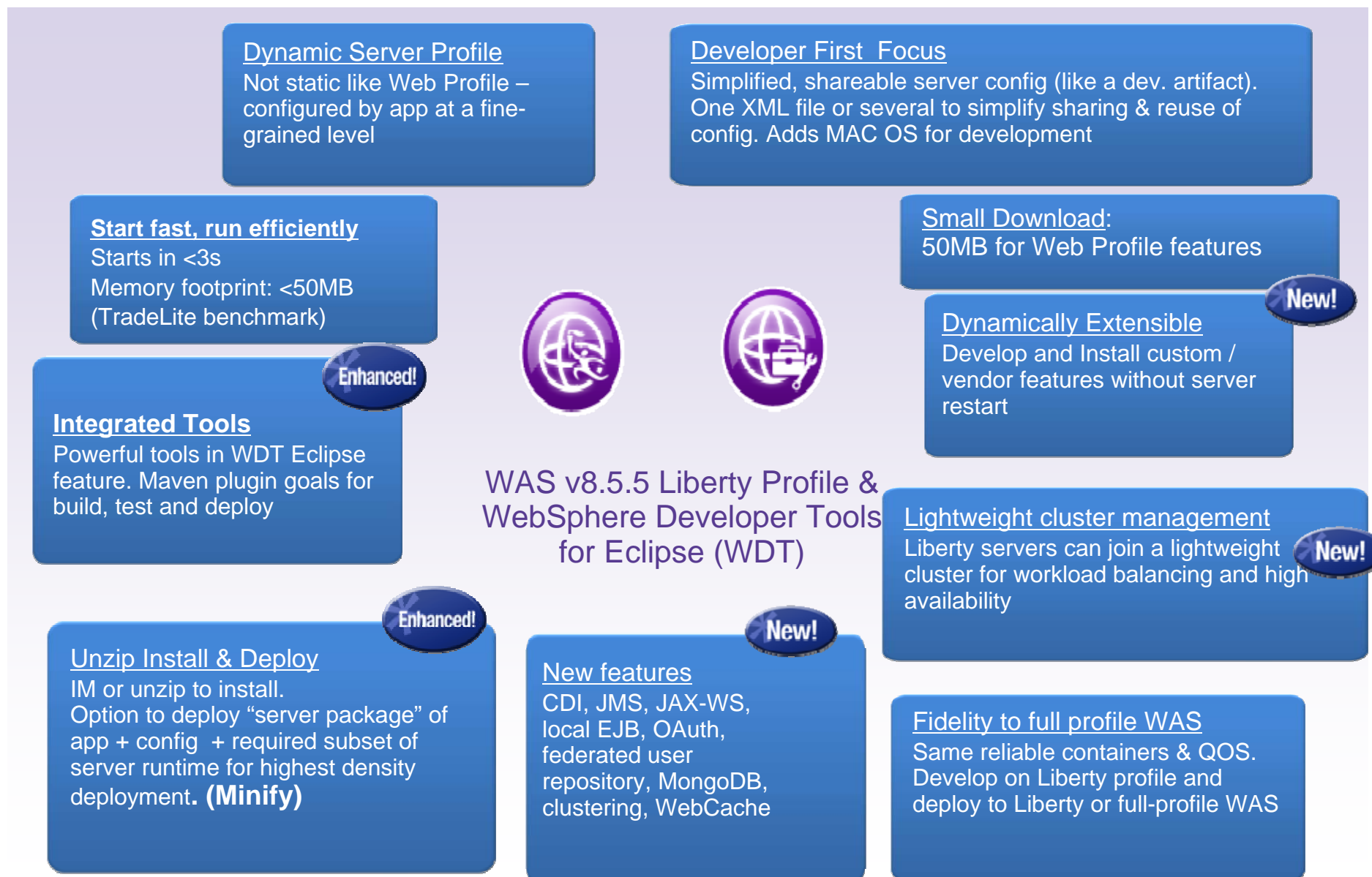
A low-cost, ready-to-go solution to build dynamic Web sites & apps, including both Liberty and full-profile WAS.

## WAS CE

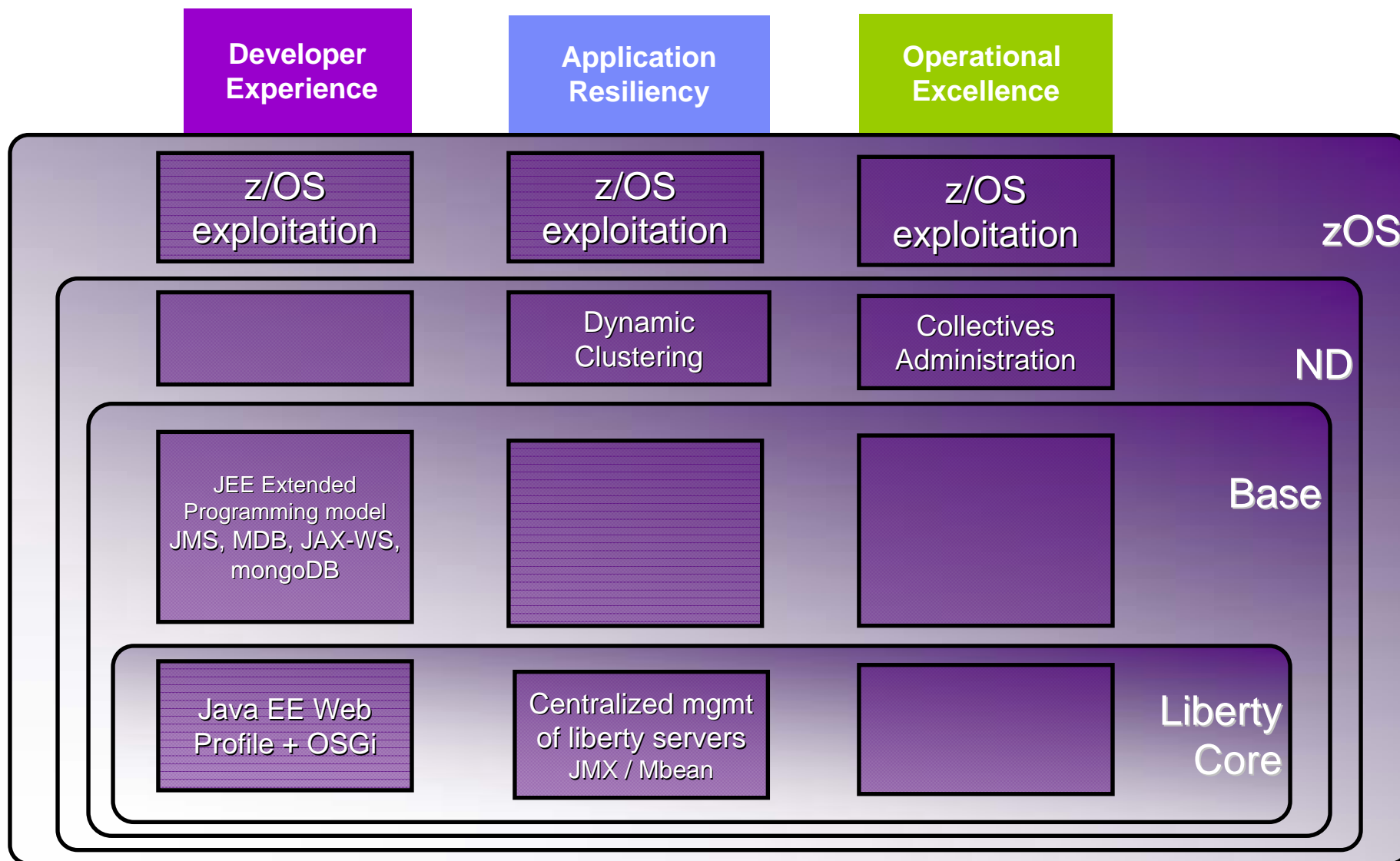
An open source-based, small footprint foundation with no up-front acquisition costs.



## WAS v8.5.5 Liberty Profile “Enhanced”

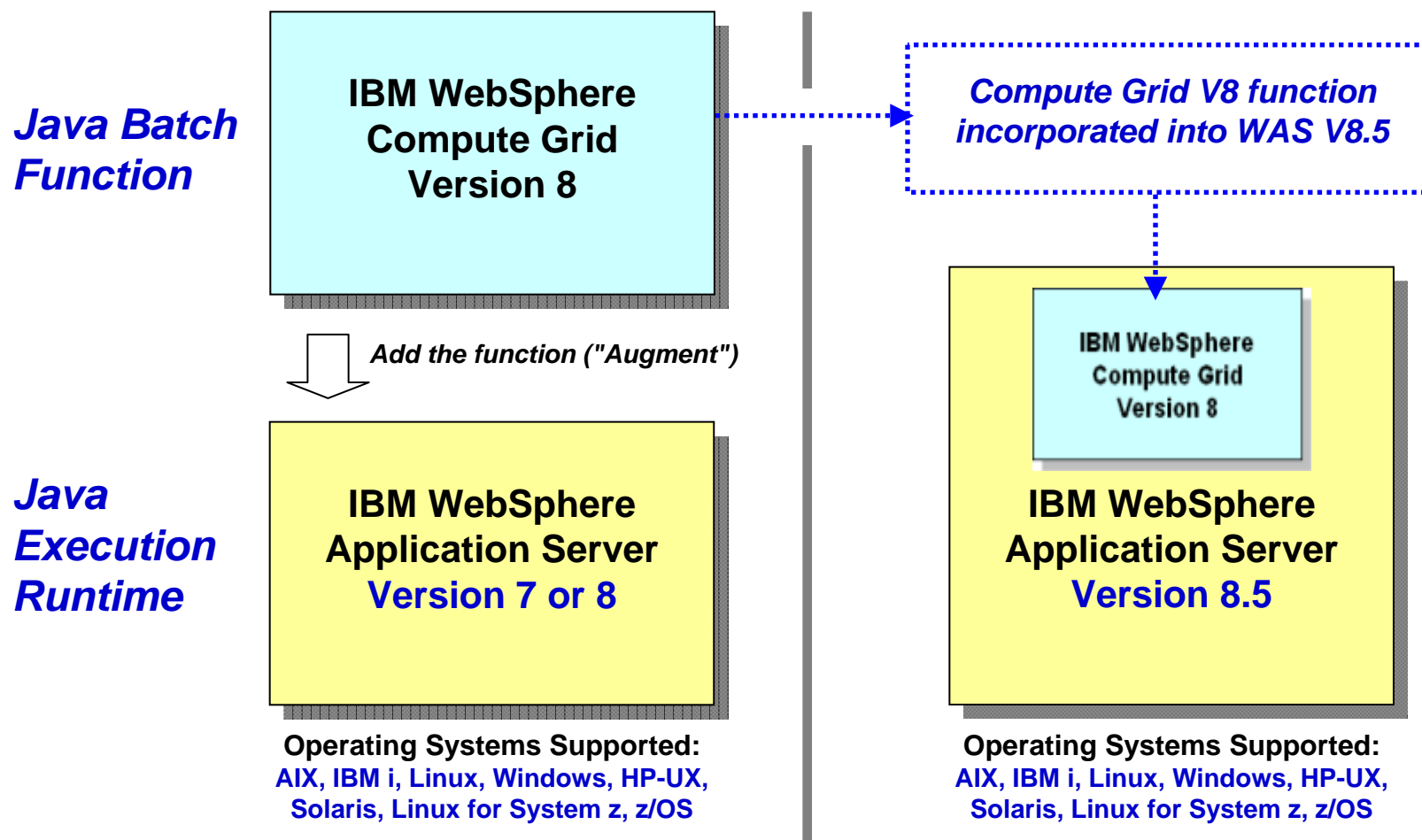


## Fit for Purpose Servers – Production Servers



## IBM Compute Grid V8 and IBM WAS V8.5

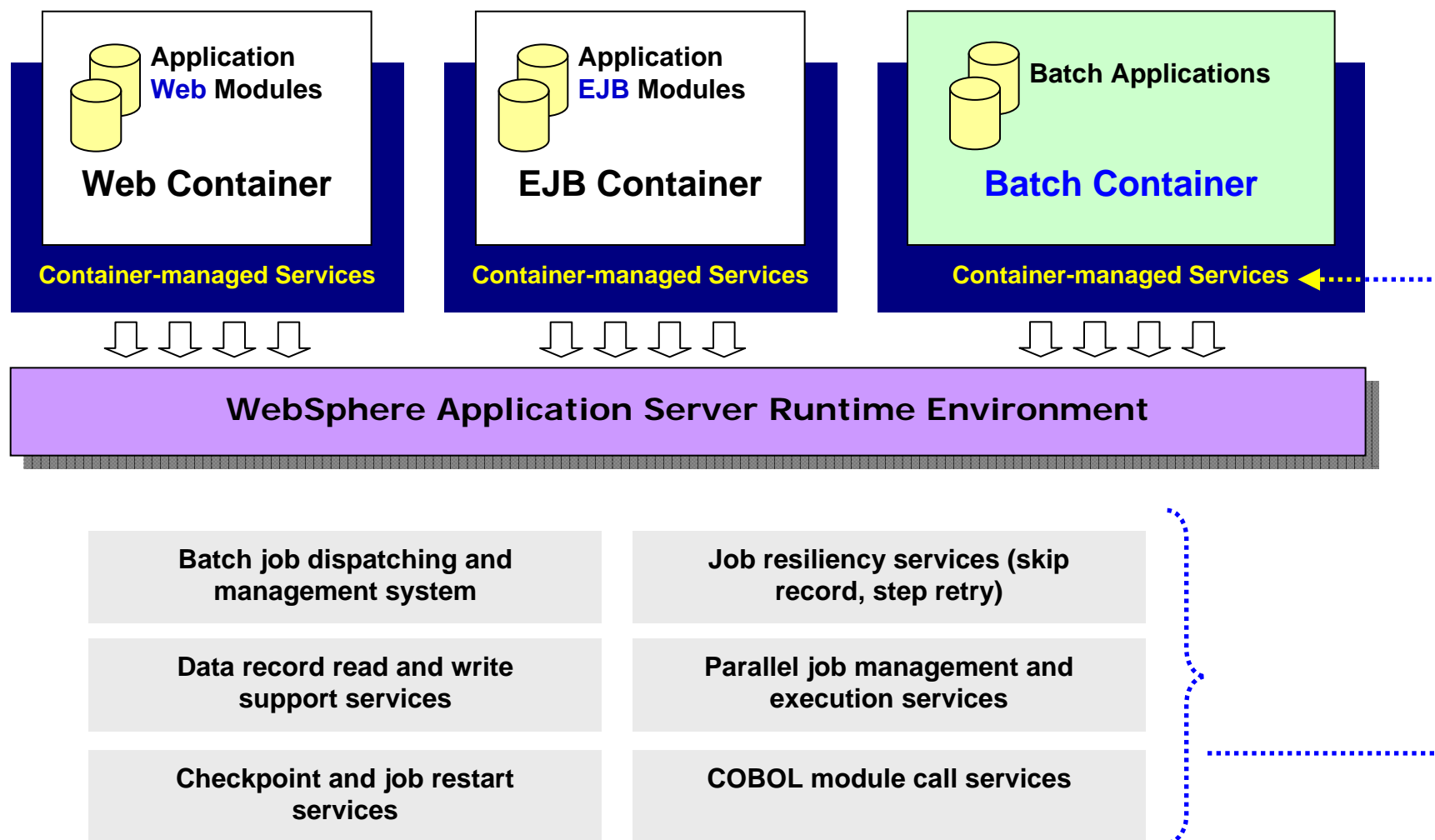
- The IBM WebSphere Java Batch function is provided in two ways today:



Function is identical between the two environments. Compute Grid V8 available for those who have not yet migrated their execution runtimes to WAS V8.5

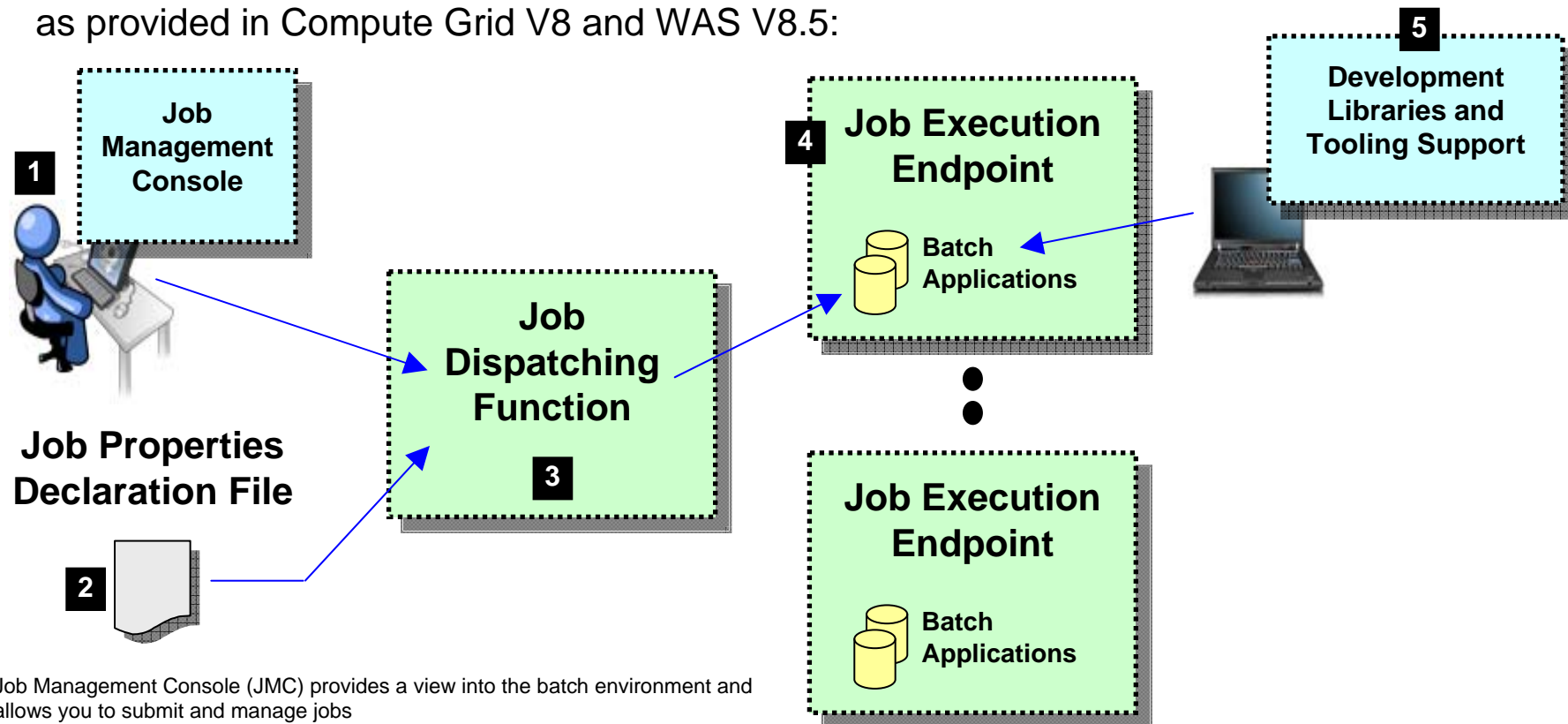
## Batch Container Added to the WAS Runtime

- At a very high-level, you may think the IBM WebSphere Java Batch function as a "batch container" operating alongside the other containers of WAS itself:



## 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:

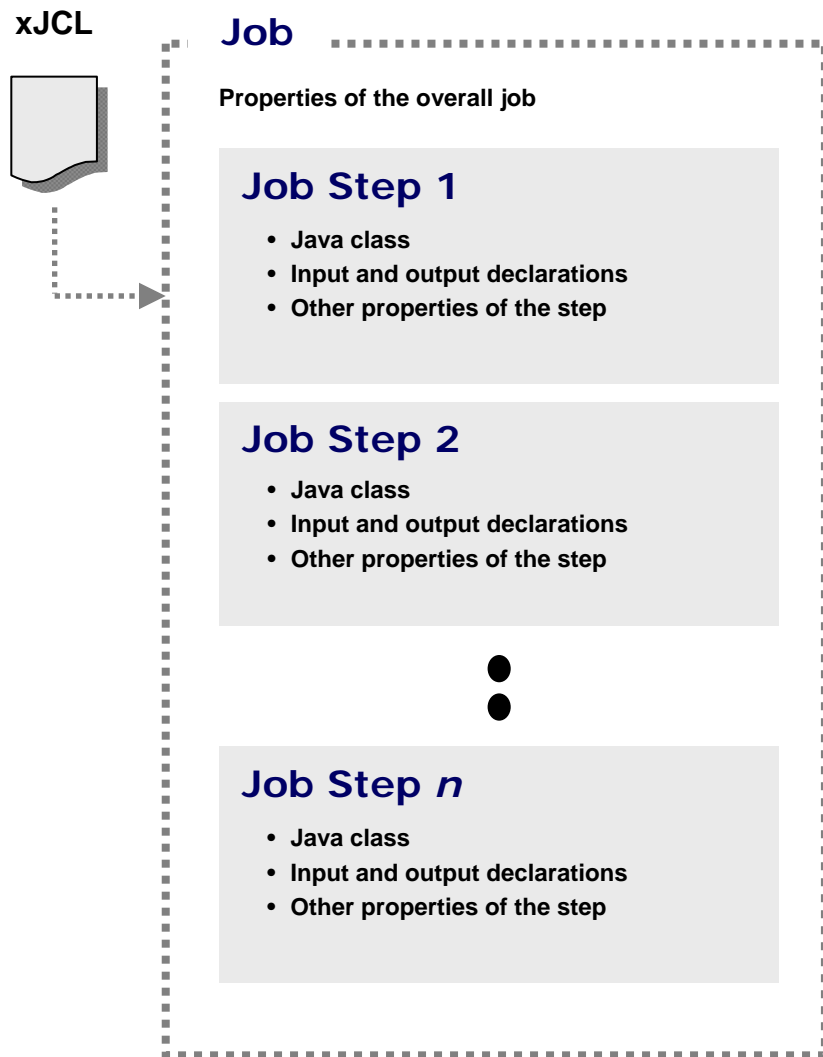


1. Job Management Console (JMC) provides a view into the batch environment and allows you to submit and manage jobs
2. Job declaration file (xJCL) provides information about the job to be run, such as the steps, the data input and output streams and the batch class files to invoke
3. The Job Dispatching 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 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

## Batch Job and Batch Job Steps

- A batch job consists of one or more steps executed in order specified in xJCL:



**The xJCL is submitted through the Job Management Console**

Interfaces provided: [HTTP browser](#), [command Line](#), [Web Services](#), [RMI](#)

**The Job Dispatching function interprets xJCL and determines which endpoint has batch application class files deployed**

**Dispatching Function invokes job and passes to the endpoint an object containing all the properties in xJCL**

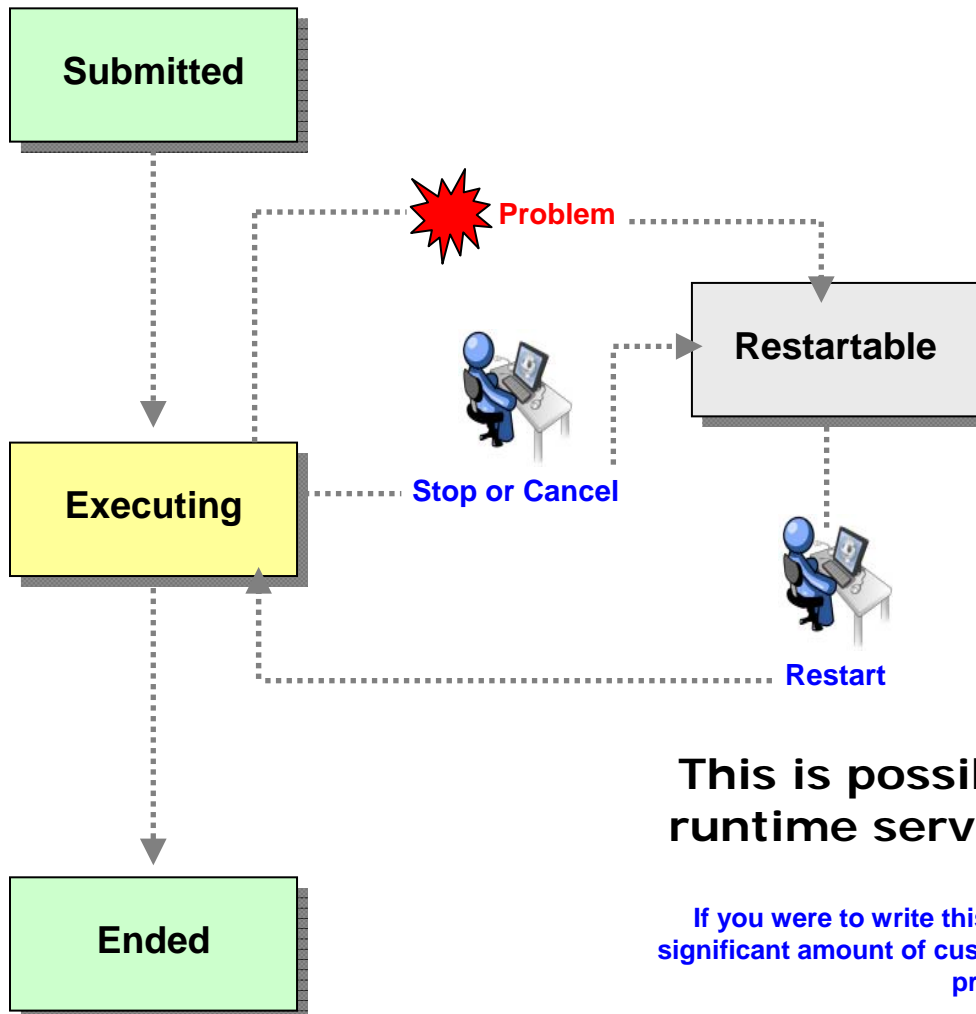
**Steps are executed in order, with conditional step processing if declared**

**Dispatching Function maintains awareness of job state**

**When job ends, job output file accessible through Job Management Console**

## Job Execution "State"

- The following picture illustrates a simplified view of the job states ... it helps illustrate a key point: executing jobs can be acted upon; failed jobs restarted.



**The Job Management Console provides you ability to act upon an executing job**

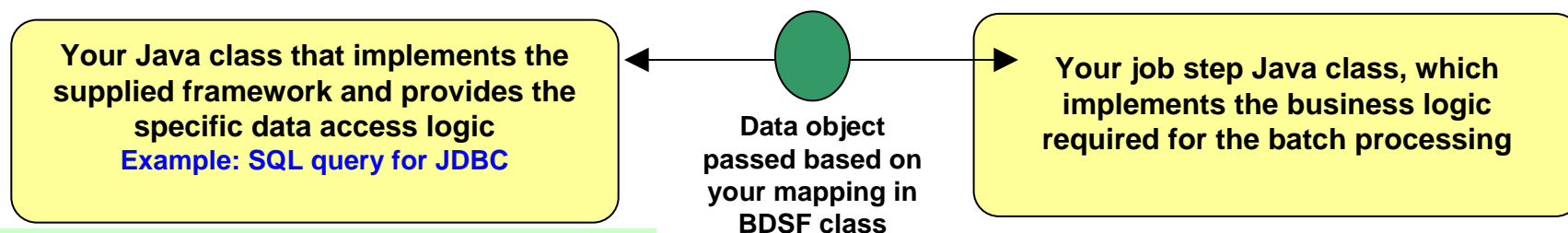
**The Batch Container is maintaining checkpoint status and will restart at the last checkpoint interval**

**This is possible because of the Java batch runtime services that are part of the batch container model**

**If you were to write this yourself then just what's shown here would require a significant amount of custom batch middleware code. IBM WebSphere Java Batch provides that as part of the product.**

## Batch Data Stream Framework (BDSF)

- This is a key function service provided by the batch container - it abstracts data read and write operations so your code may focus on the business logic:



### Batch Data Stream Framework

#### Supplied "patterns" for data access:

- JDBC read or write operations
- JPA read or write operations
- File read or write operations
- z/OS Data Set read or write operations



**Batch Data Stream retrieves result set from data persistence store (DB, file, etc.)**

**Batch Data Stream maps data fields to data object**

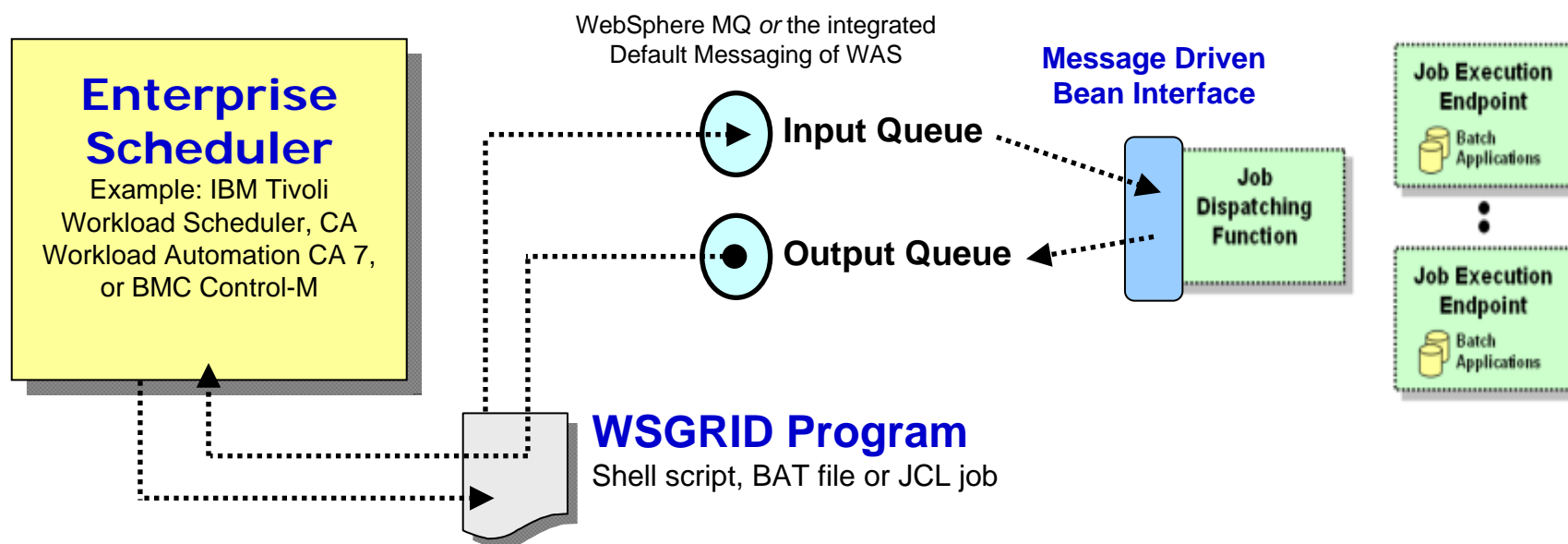
**For each record in result set, BDSF invokes your job step, passing a data object mapped to your specifications**

**Your job step code stays focused on business logic, not Java stream handling and data object formatting**



## Integration with Enterprise Scheduler Functions

- The Job Dispatching Function has a Message Driven Bean (MDB) interface. IBM supplies a program that integrates schedulers with WebSphere Java Batch:



**WSGRID is seen by Scheduler as any other batch job it starts and monitors**

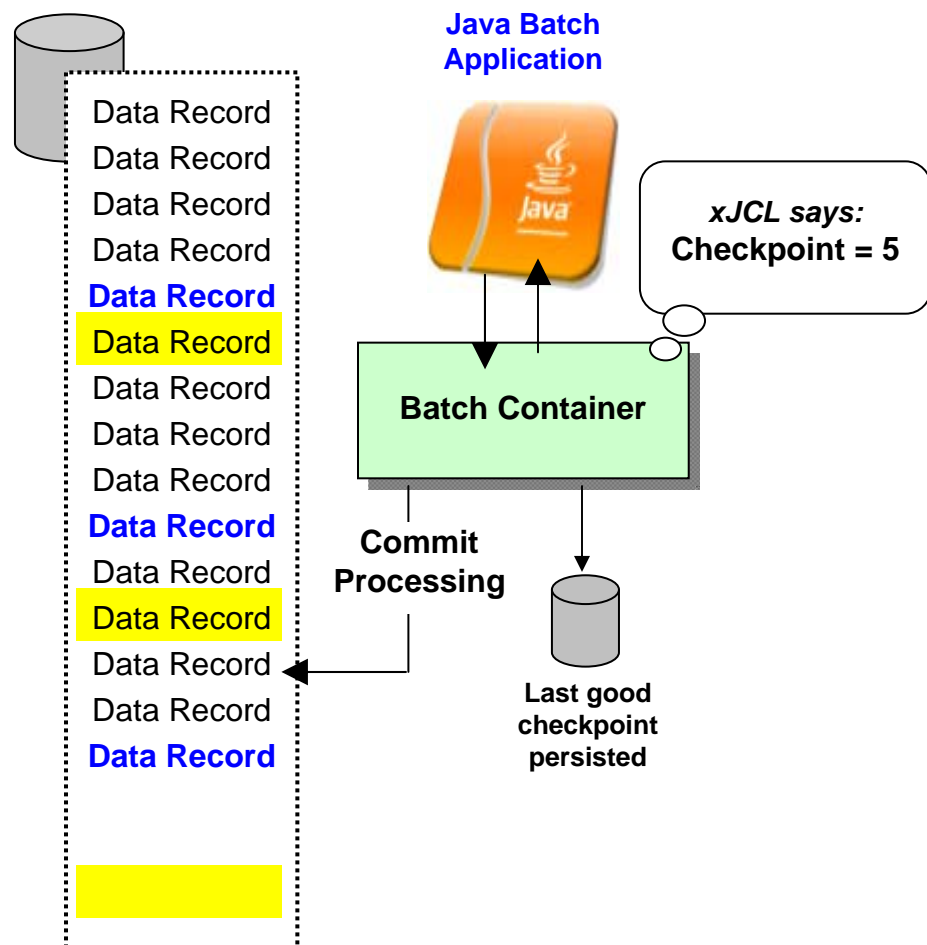
**WSGRID interacts with Job Dispatching, submitting the job and processing Java batch job output back to STDOUT or JES Spool if z/OS**

**WSGRID program stays up for life of job in WebSphere Java Batch**

**To the Scheduler, WGRID is the Java Batch job ... but behind WSGRID is all the WebSphere Java Batch function we'll discuss**

# Transactional Checkpoint Processing

- The batch container provides the ability to checkpoint at intervals based on either record count or time. The container keeps track of last checkpoint.



**Checkpoint interval (record or time) specified in the xJCL**

**This is a function of the batch container, *not* your application code**

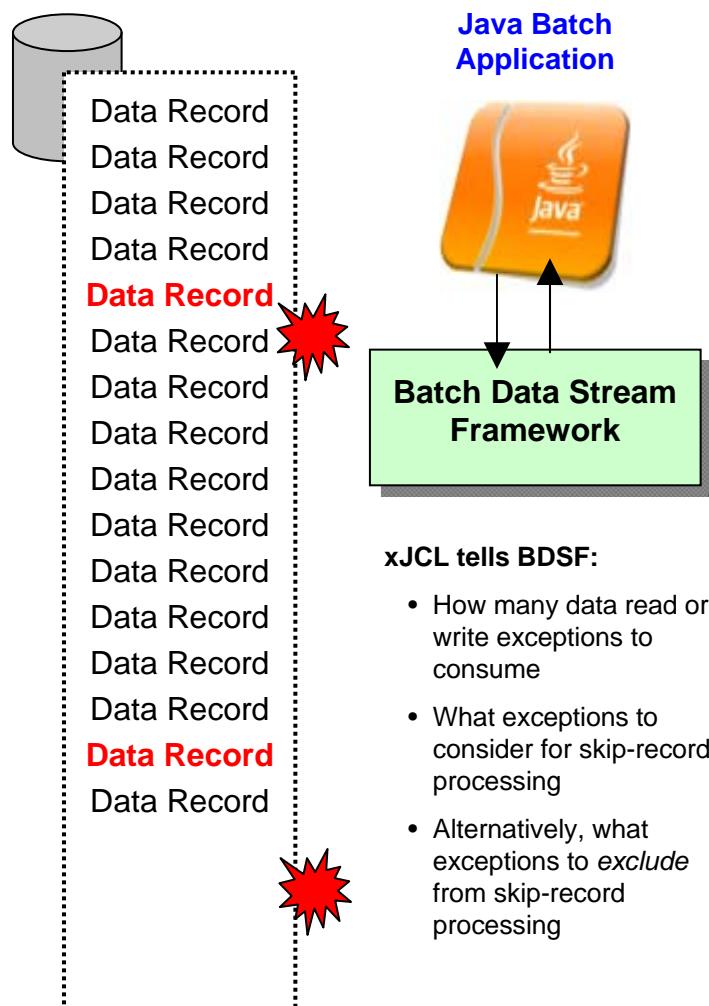
**As checkpoint intervals are reached, container commits and records the checkpoint attained**

**In the event of a failure, job may be restarted at the last good checkpoint**

**Set the checkpoint interval based on your knowledge of balance between recoverability and efficiency**

# Skip-Record Processing

- Provides a container-managed way of tolerating data read or write errors so the job itself may continue on. Information about data errors may be logged.



**Objective: allow job to continue if a data read or write exception occurs in BDSF**

Why fail a million-record job just because of one or two read or write exceptions? Better to complete the job and allow auditors to go back and investigate the few exceptions.

**Skip-Record processing allows BDSF to keep exception and *not* surface it to your application**

This takes burden off your application code to explicitly handle data read or write exceptions that may occur

**A "skip-record listener" may be called so your code may lot information about skipped record**

More on "batch listeners" coming up

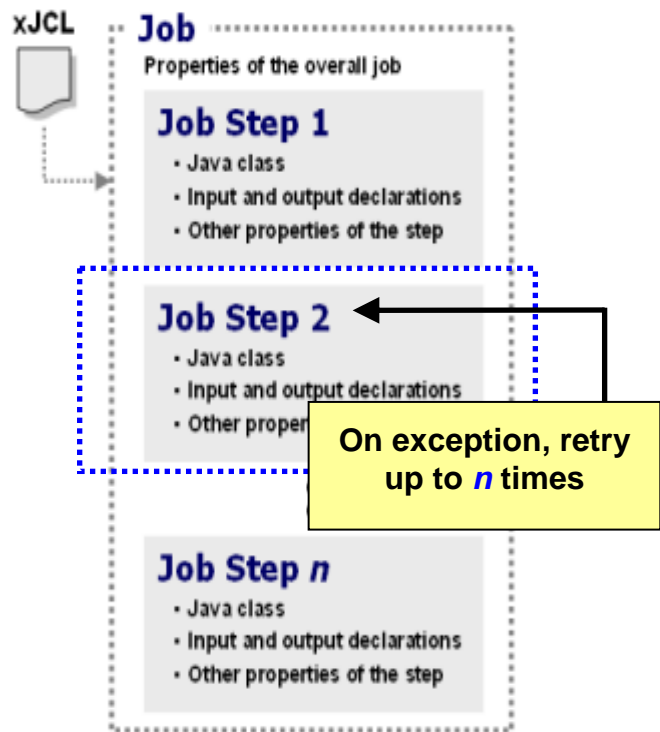
**xJCL properties allow you to specify how many records may be skipped and what exceptions to include or exclude from consideration**

**When skip limit is reached, further exceptions are surfaced to application. That may result in job failing and going into a restartable state**

Normal restart-at-checkpoint would occur

## Retry-Step Processing

- Provides a means of retrying a job step in the event of an exception thrown. If successful on retry then the job continues and your processing completes.



### xJCL tells Container:

- How many step retries may be attempted
- What exceptions to consider for retry-step processing
- Alternatively, what exceptions to *exclude* from retry-step processing
- Whether to process a delay before attempting a retry of the step

**Objective:** retry step in attempt to allow overall job to continue and complete when an unanticipated exception is thrown

This is at level higher than skip-record ... this is if an unhandled exception is thrown when the job step function is called

Batch container falls back to last good checkpoint and restarts from there

A "retry-step listener" may be called so you can perform custom action upon retry-step processing

[More on "batch listeners" coming up](#)

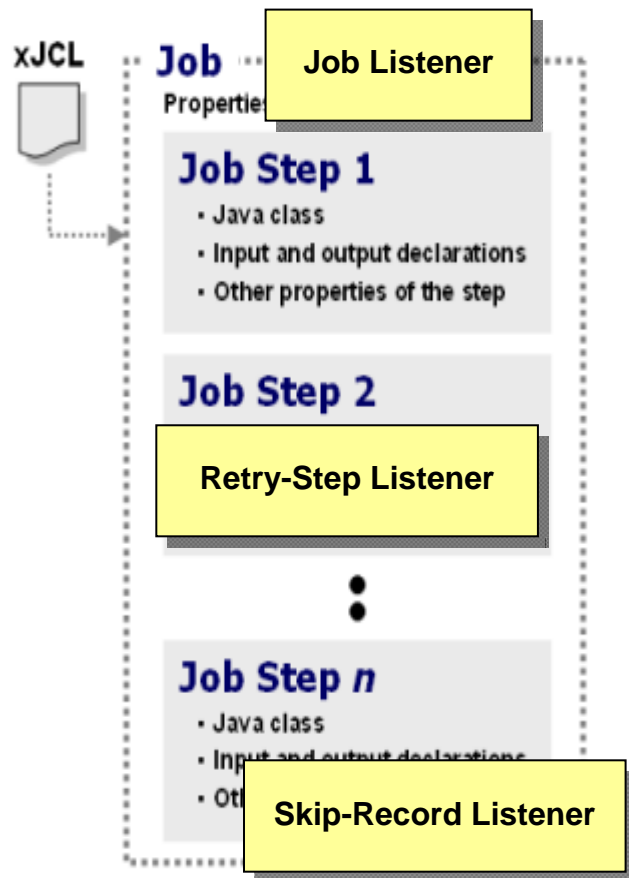
xJCL properties allow you to specify how many retry attempts will be performed and what exceptions to include or exclude from consideration

When retry limit is reached, job will go into restartable state

[Normal restart-at-checkpoint would occur](#)

## Batch "Listeners"

- These are callout points where your customer "listener" code will be called when key events occur. The callouts are managed by the batch container:



Listeners provide ability to have your code called at key points during batch job execution

### Job Listener

- Callouts occur:  
*Start of the job; Start of each step; End of each step; End of job*
- Register your code to container with property in xJCL
- Use this to perform any special setup or cleanup actions at those points in the lifecycle of a batch job

### Retry-Step Listener

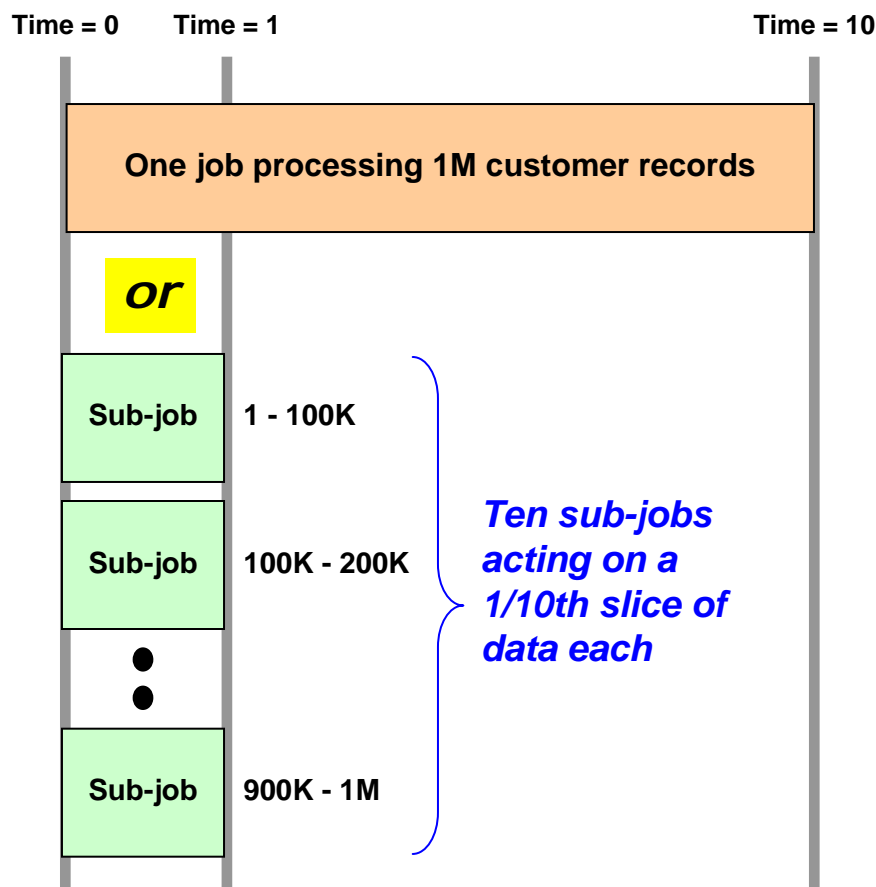
- Callouts occur:  
*When the exception is thrown; When the retry is attempted*
- Register your listener with code in application `createJobStep()` method
- Use this to take action at these points, such as logging information about the exception and the point in the processing where it occurred

### Skip-Record Listener

- Callouts occur:  
*On skipped read or skipped write operation*
- Register your listener with code in application `createJobStep()` method
- Use this to take action at these points, such as logging information about the exception and the record skipped

# Parallel Job Manager

- The Parallel Job Manager (PJM) provides a way to "parameterize" logic so parallel sub-jobs may act on a slice of the overall batch job data:



**Objective is reduction in overall job completion time**

*Which shortens overall batch window if other jobs are dependent on this job for completion*

**xJCL specifies whether job is to be run in parallel, and if so how:**

- One JVM, multiple threads
- Multiple JVMs

**Your "parameterizer" code is called at start so data range may be segmented into sub-job slices**

**Job is submitted, then PJM dispatches "sub-jobs" to act on each data range**  
*"Parameterizer" code constructs data range query strings to be used by each sub-job*

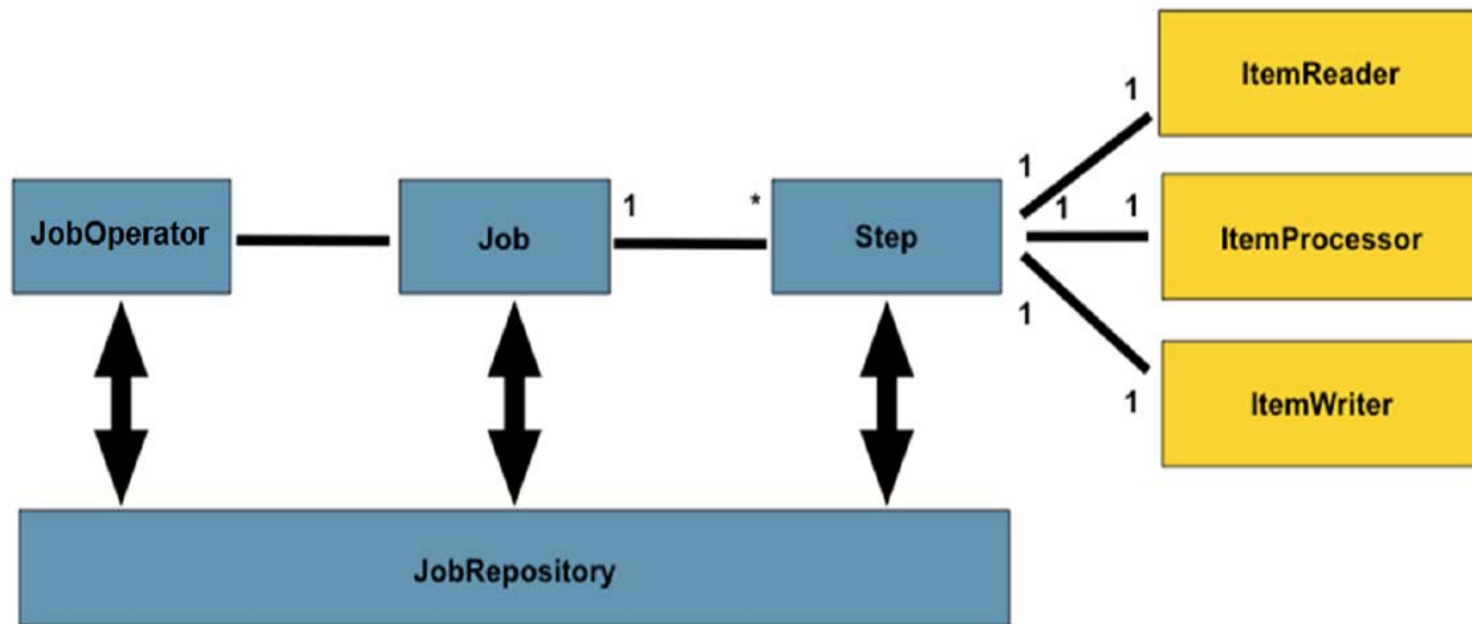
**PJM manages "top-job" and all subordinate "sub-jobs" to completion**

---

## JSR 352 Batch Applications for the Java Platform

- Adds new application type (Batch) to Java
- Expert Group (operation April 2013-present) :
  - IBM (Lead)
  - Credit Suisse
  - Oracle
  - Clarkson University
  - Redhat
  - Industry Consultants (2)
  - VMWare
- Specification approved on 8 April 2013.
- Part of Java EE 7 (Reference Implementation is GlassFish)
- Separate Reference Implementation available for Java SE

## Batch for Java Landscape



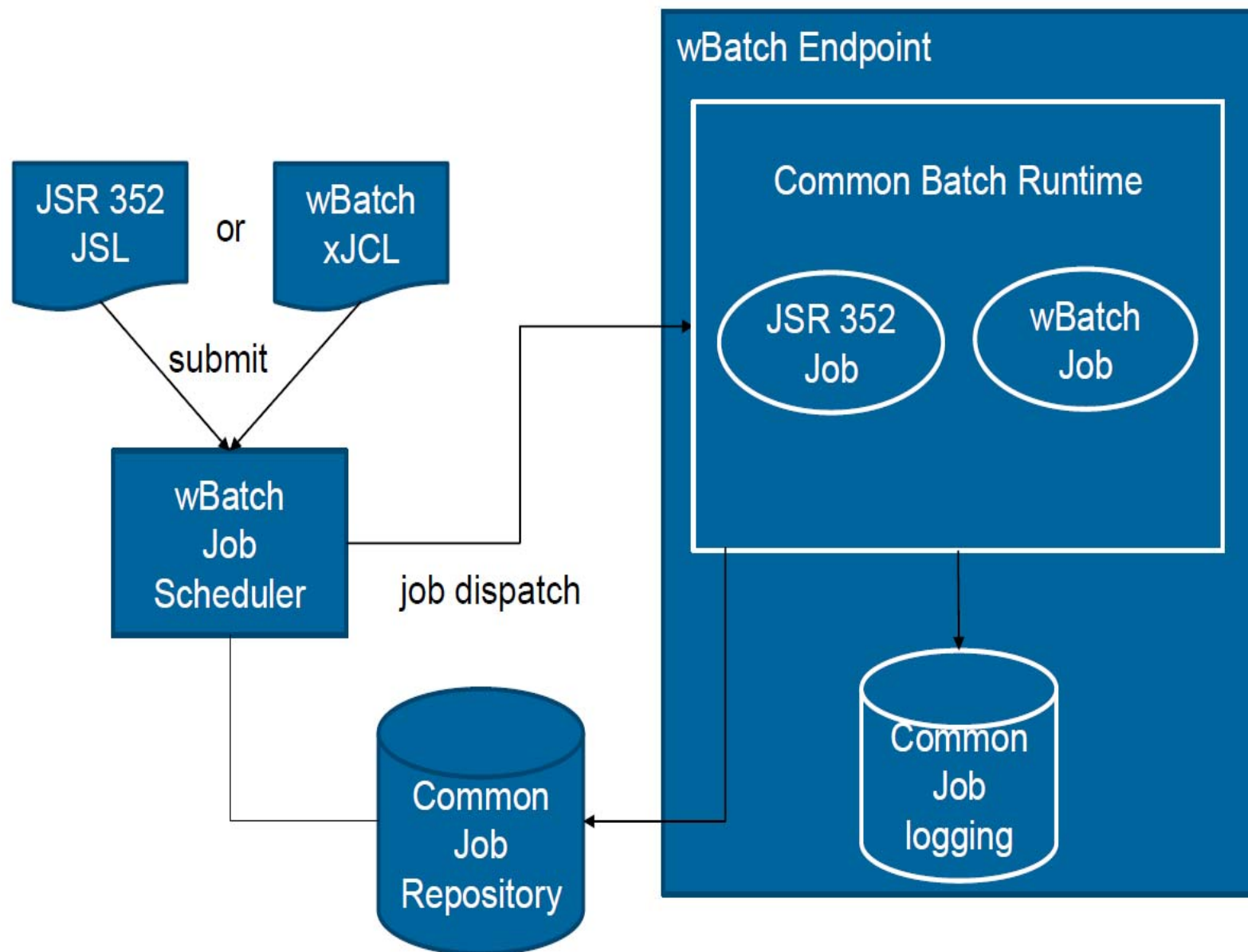
- Jobs (JSL) are started through the JobOperator interface.
- A typical job is a series of one or more steps, where each step is comprised of a reader, processor, and writer.
- Typical job steps read and process a series of input records and produce output records in a long running loop with periodic checkpoints to support restart.
- Job history is stored in a repository and is available through the JobOperator interface.



## JSR 352 Comparison to WebSphere (CG) Batch

Feature	JSR 352	WAS Batch
Transactional Batch Model	Yes	Yes
Compute Intensive Model (Batchlet)	Yes	Yes
Partitioned Step Execution	Yes	Yes
Job and Step Contexts	Yes	Yes
Job and Step Listeners	Yes	Yes
Job Specification Language (JSL)	Yes	Yes
JSL Substitution	Yes	Yes
Operations Interface	Yes	Yes
Concurrent Steps	Yes	NO
Restart Substitution	Yes	NO
CDI Integration	Yes	NO
Job Dispatcher	NO	Yes
Job Logs	NO	Yes
Distributed Parallelization	NO	Yes

## Theoretical JSR 352/WebSphere Batch Integration



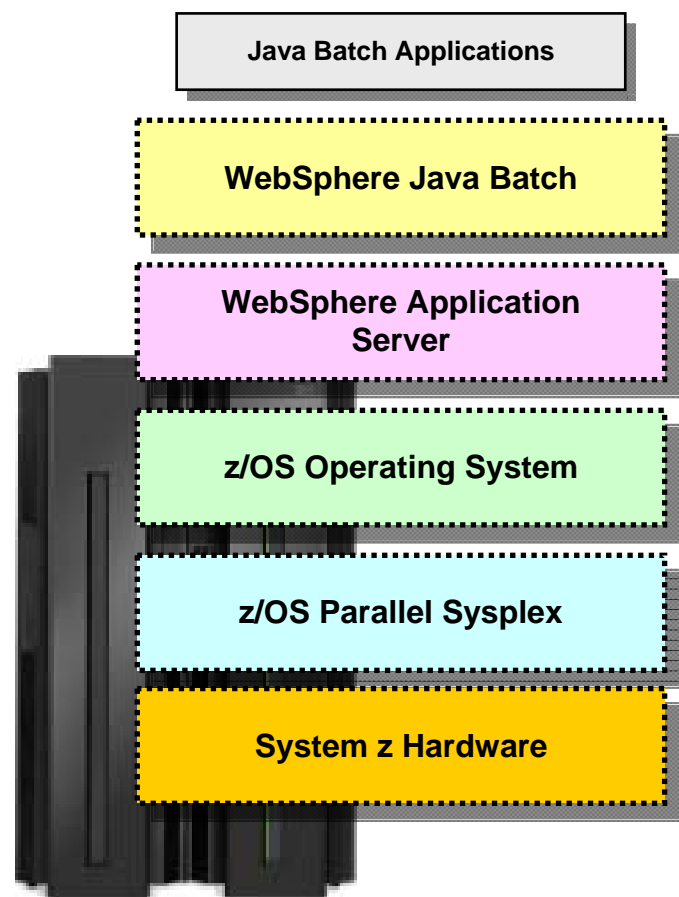
---

## Summary

- WAS v8.5 delivers enhanced value for batch workloads.
- Full WebSphere Compute Grid feature-set now part of WAS
  - Increases WAS value
  - Simplifies ownership
- Proven function encourages customer adoption where warranted by business need.
- JSR 352 delivers industry standardization for Java batch applications.

## The Value Statements of WebSphere Batch on z/OS

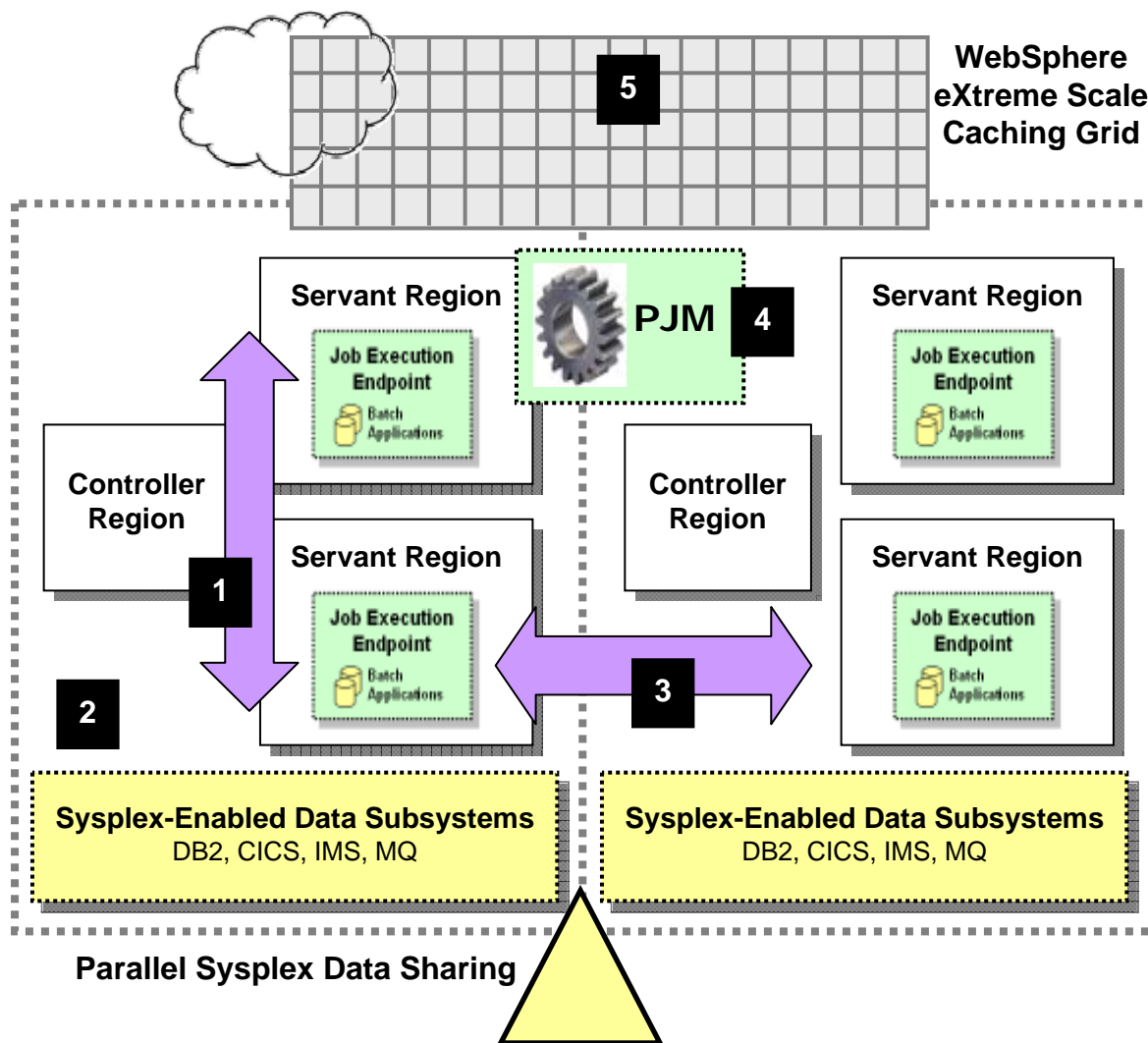
- If we start from a high level, we see the following platform benefits that accrue up to Java batch running on the platform:



- Batch runtime services
- Batch development tooling
- Proven Java runtime environment
- WAS deployment and management model
- WAS Qualities of Service
- Decades of maturity, stability and reliability
- Consolidated operation and management model
- Rich set of system facilities: WLM, SMF, RMF, SAF
- z/OS instance clustering with central data sharing
- Elimination of single points of failure for availability
- Near linear scalability up to 32 nodes
- Engineered from beginning for reliability and stability
- Engineered for high levels of I/O
- Extremely long mean time between failure
- Speciality engines for specific work offload
- Dynamic capacity expansion
- Logical partitioning using PR/SM hypervisor

## Scaling Up the Java Batch Solution on z/OS

- There are several ways in which a WebSphere Java batch solution can be scaled up to provide greater batch throughput and shorter execution windows:



### 1. Vertical

WAS z/OS servant regions provide a type of "vertical cluster," giving you additional batch compute resources

### 2. Capacity on Demand

CPU processors may be dynamically added to a z/OS LPAR, increasing the capacity for processing work

### 3. Horizontal

WAS z/OS clustering on top Parallel Sysplex provides near-linear scalability up to 32 nodes with a central data sharing model

### 4. Parallel Processing

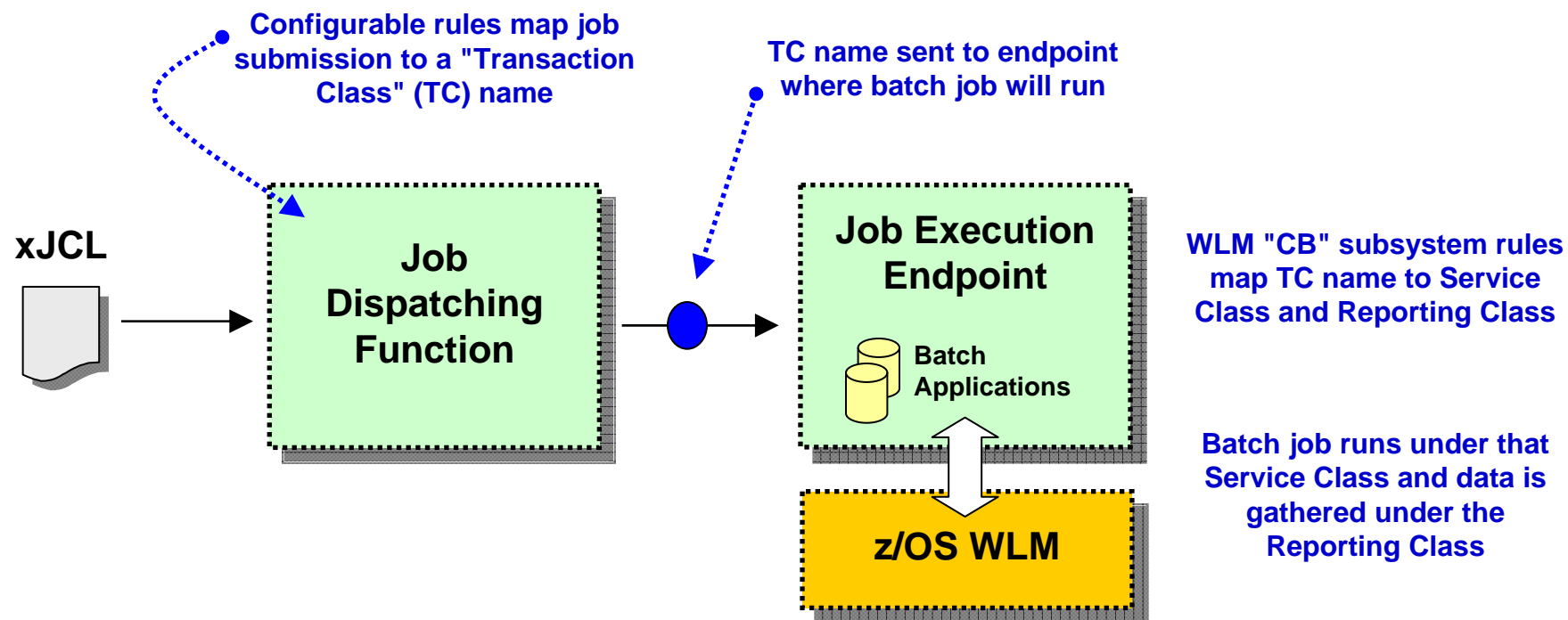
The Parallel Job Manager may be used to partition data into sub-jobs, which may then be run on multiple threads, different servants, or different servers on other LPARs.

### 5. Data Caching

WebSphere eXtreme Scale provides a data caching grid from which Java batch may fetch and store data

## WLM Classification

- The submitted job can be tagged with a WLM "transaction class," which may then be used to map the job to a WLM Service Class or Reporting Class:



**Classifying to a Service Class allows WAS z/OS to place work into separate servant regions based on Service Class**

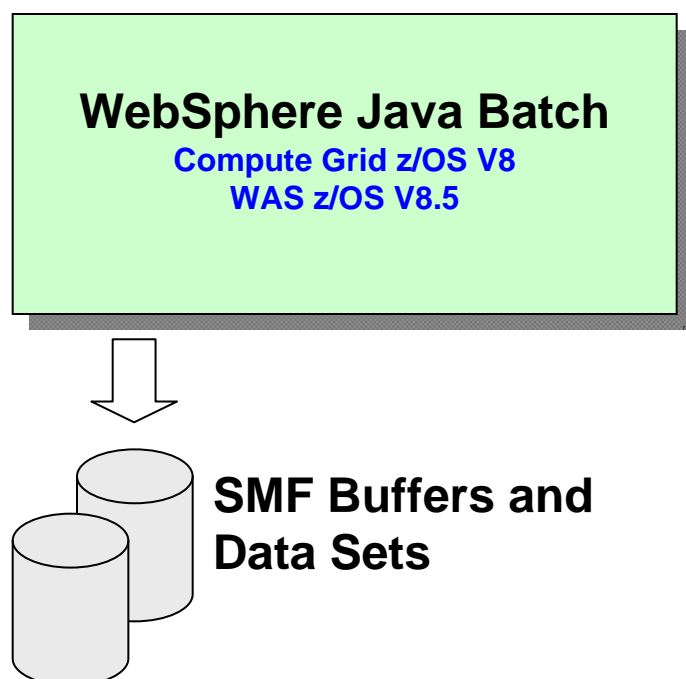
*A somewhat sophisticated practice not widely used*

**Classifying to a Reporting Class allows WLM to gather system information for all work running under that Class**

*A much more common practice that is very useful for understanding usage patterns and for capacity planning*

## SMF 120.9 Activity Recording

- WAS z/OS supports the use of activity recording using the SMF 120.9 record. WebSphere Java Batch extends the record with batch activity information:



**Job activity records allow you to understand how your system is being used and to provide chargeback data**

**Activity recording available on all platforms, but only z/OS uses SMF, which is an extremely efficient logging mechanism**

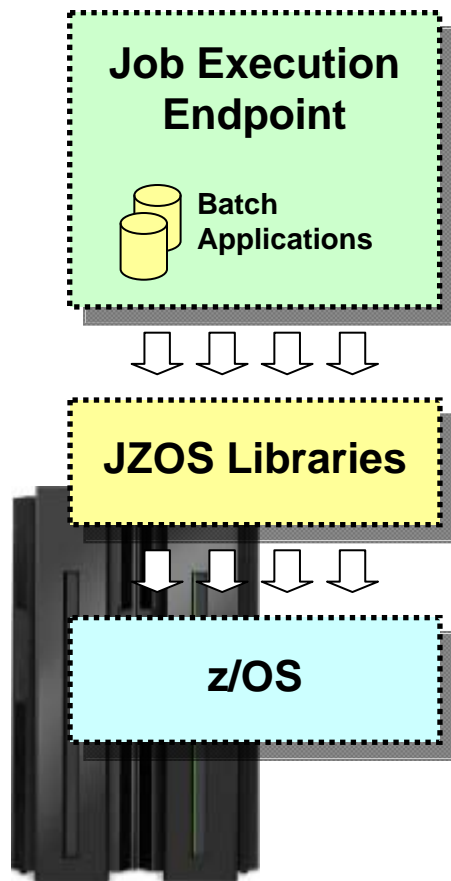
**Provides historical records for usage analysis and batch capacity planning**

### **Information captured:**

- Job submitter
- Date and time of submission
- Final job state
- Total CPU used for job
- General processor used for job
- zAAP usage derived:  $\text{Total} - \text{GP} = \text{zAAP}$

## Use of JZOS Services

- JZOS is a set of functions that make using Java on z/OS much easier and useful. The JZOS class libraries may be used in batch application development:



**JZOS is technology acquired by IBM from Dovetail Technologies\* and incorporated into z/OS\*\***

### Examples of some z/OS services available:

**DfSort** - interface for invoking DFSORT

**MvsConsole** - class with static methods to interface with the MVS console.

**MvsJobSubmitter** - class for submitting batch jobs to JES2 or JES3 from a Java program

**PdsDirectory** - class for opening a PDS directory and iterating over its members.

**WtoMessage** - simple data object/bean for holding a WTO message and its parameters.

**ZUtil** - static interface to various z/OS native library calls other than I/O.

**WebSphere Java Batch and JZOS are not mutually exclusive ... the JZOS class libraries may provide exactly what you need for your batch application to access z/OS functions and services**

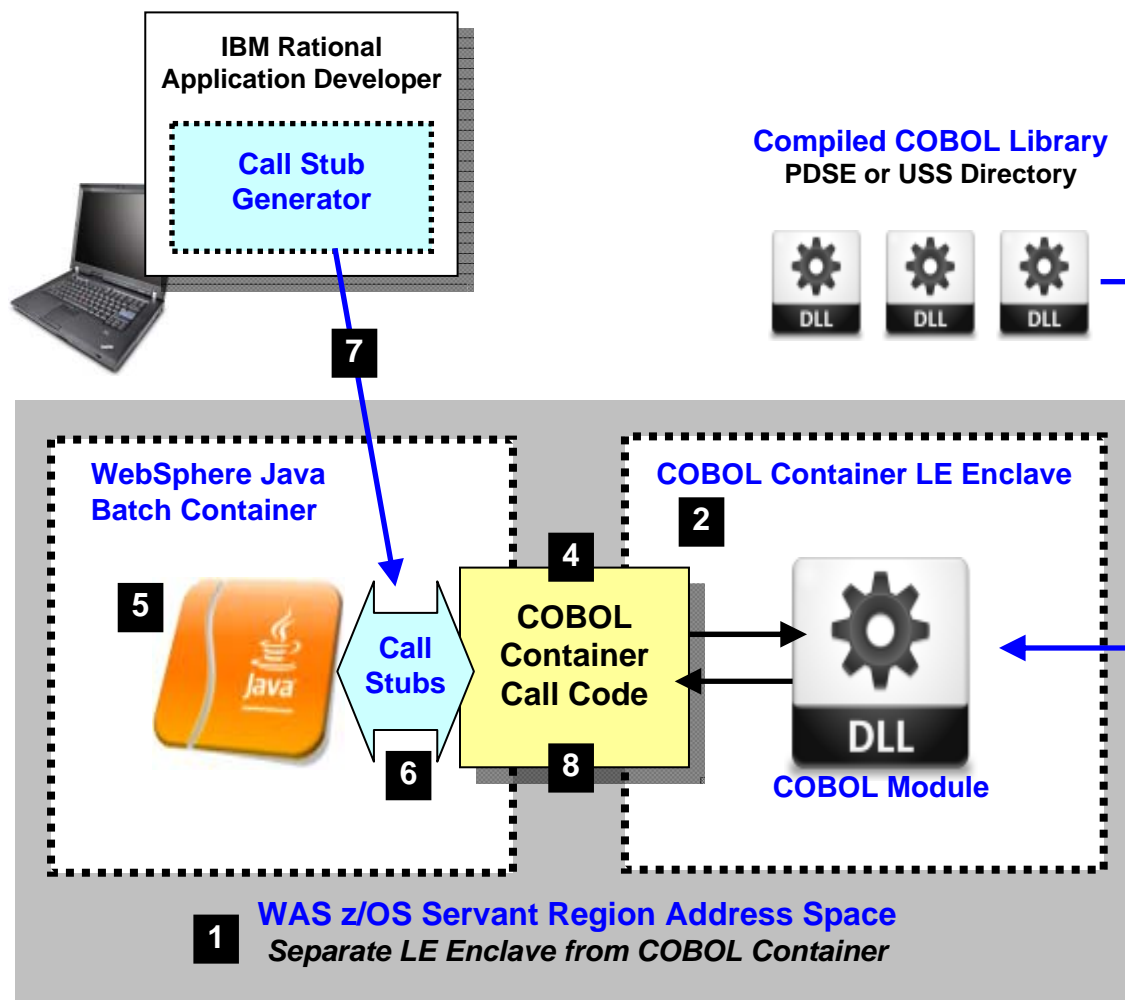
\* [www.dovetail.com](http://www.dovetail.com)

\*\* <http://www-03.ibm.com/systems/z/os/zos/tools/java/products/jzos/overview.html>



# COBOL Container

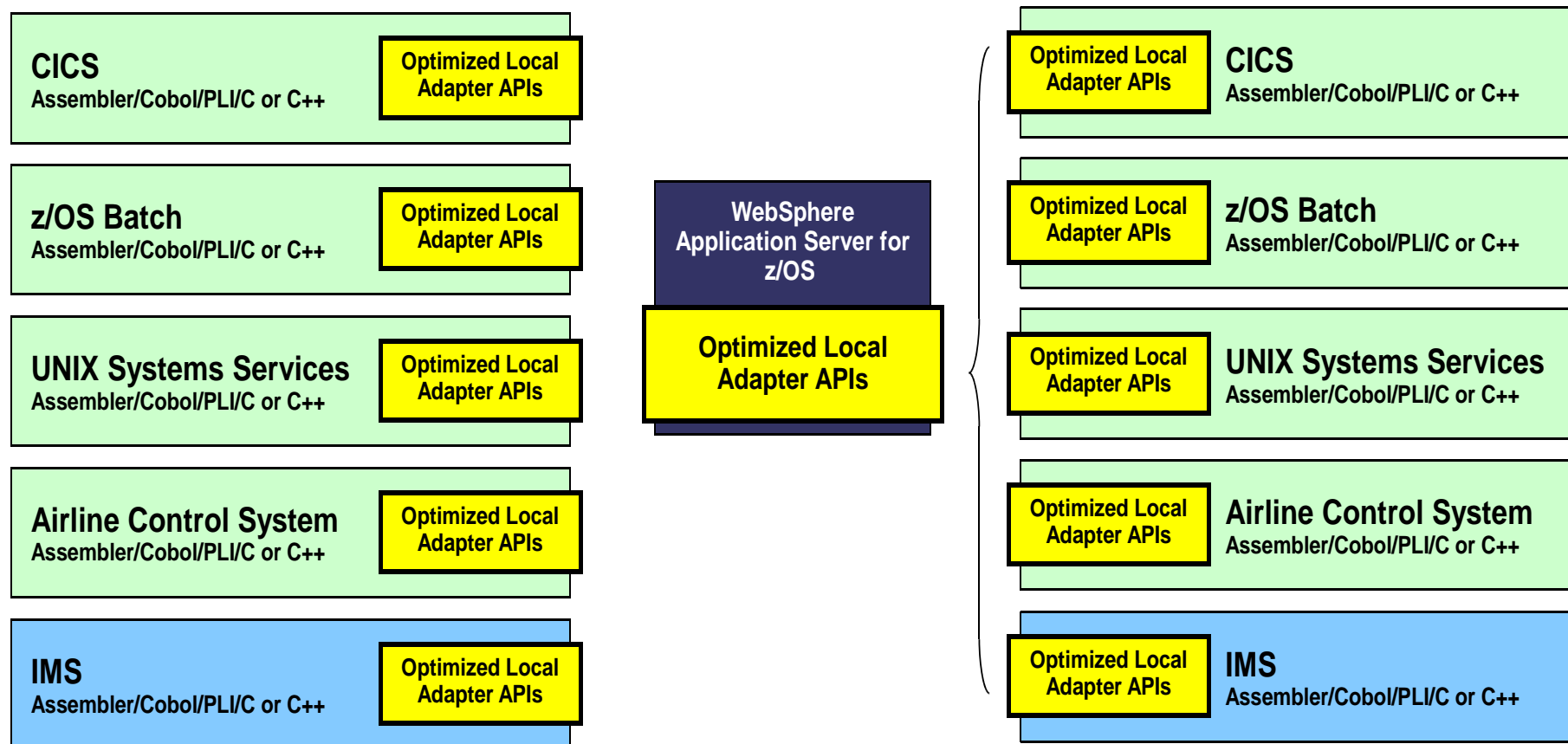
- The COBOL Container provides a way to call and execute COBOL modules in the WAS z/OS server address space ... a very efficient way to call COBOL



**Lines of code needed to invoke COBOL many times less than other means of calling COBOL from Java**

# WebSphere Optimized Local Adapters

- The new WOLA Adapters allow a bi-directional Cross-Memory communication between WAS and CICS, z/OS Batch and USS.



\* Tuesday 16:15-17:30 Petori: zAI13 WebSphere Application Server for z/OS Optimized Local Adapters (WOLA) - Real World Scenarios by Dave Follis

## Enterprise Modernization scope: COBOL << >> Java

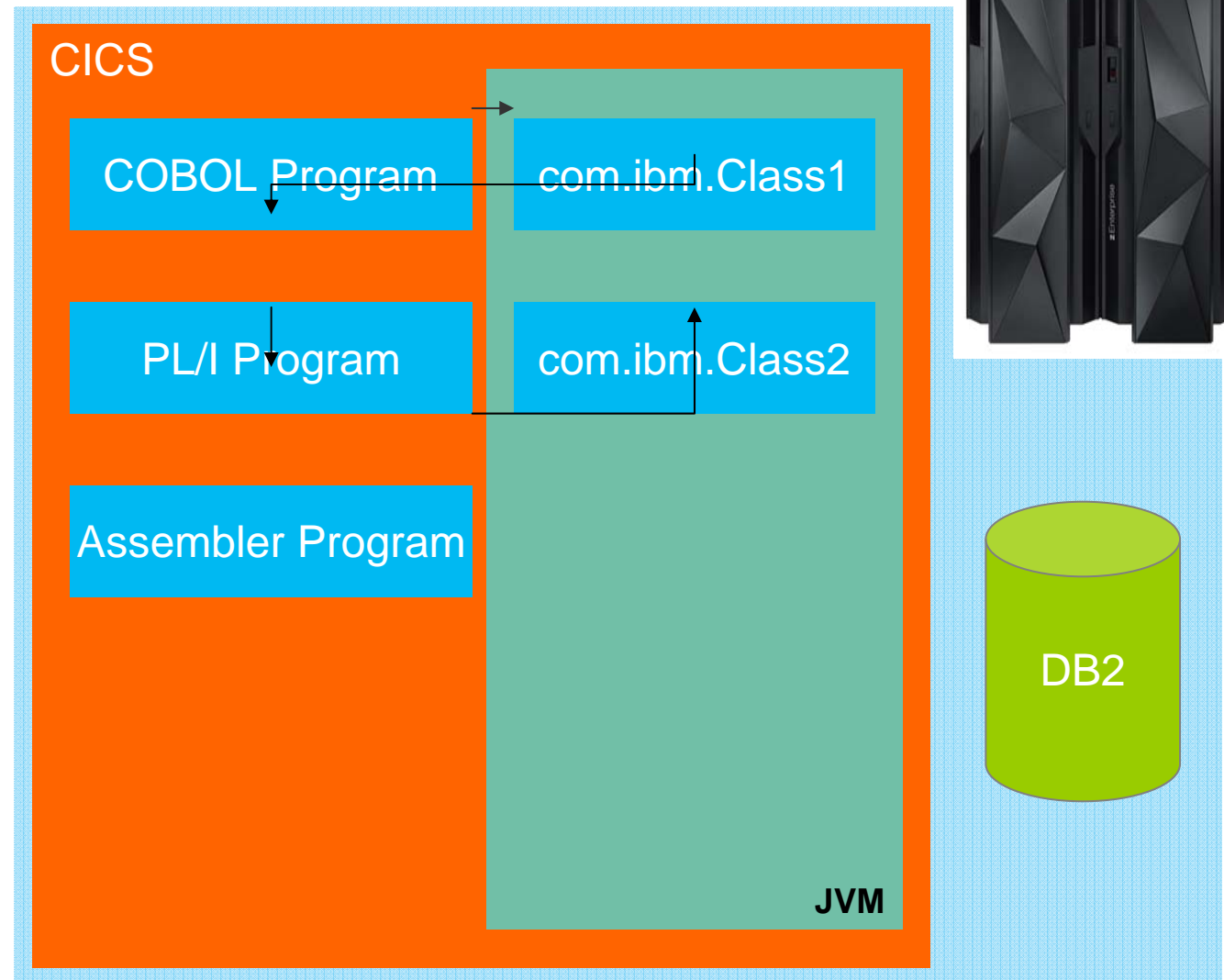
- Core Mainframe applications written in Cobol, PL/I and Assembler calling Java applications
- Customers with the requirement **to integrate Java based standard software** in existing mainframe applications
- Customers decided to **re-write their core application in Java**
- Customers decided to **write new business logic in Java** instead of traditional mainframe languages
- Customers with the need to **integrate business logic in Java** in their existing mainframe applications
- Customers able to **generate Java instead of Cobol from a 3GL language**
- Customers **introducing Java Batch** in their existing mainframe batch environment
- Customers with the requirement **to integrate the mainframe with the SAP**



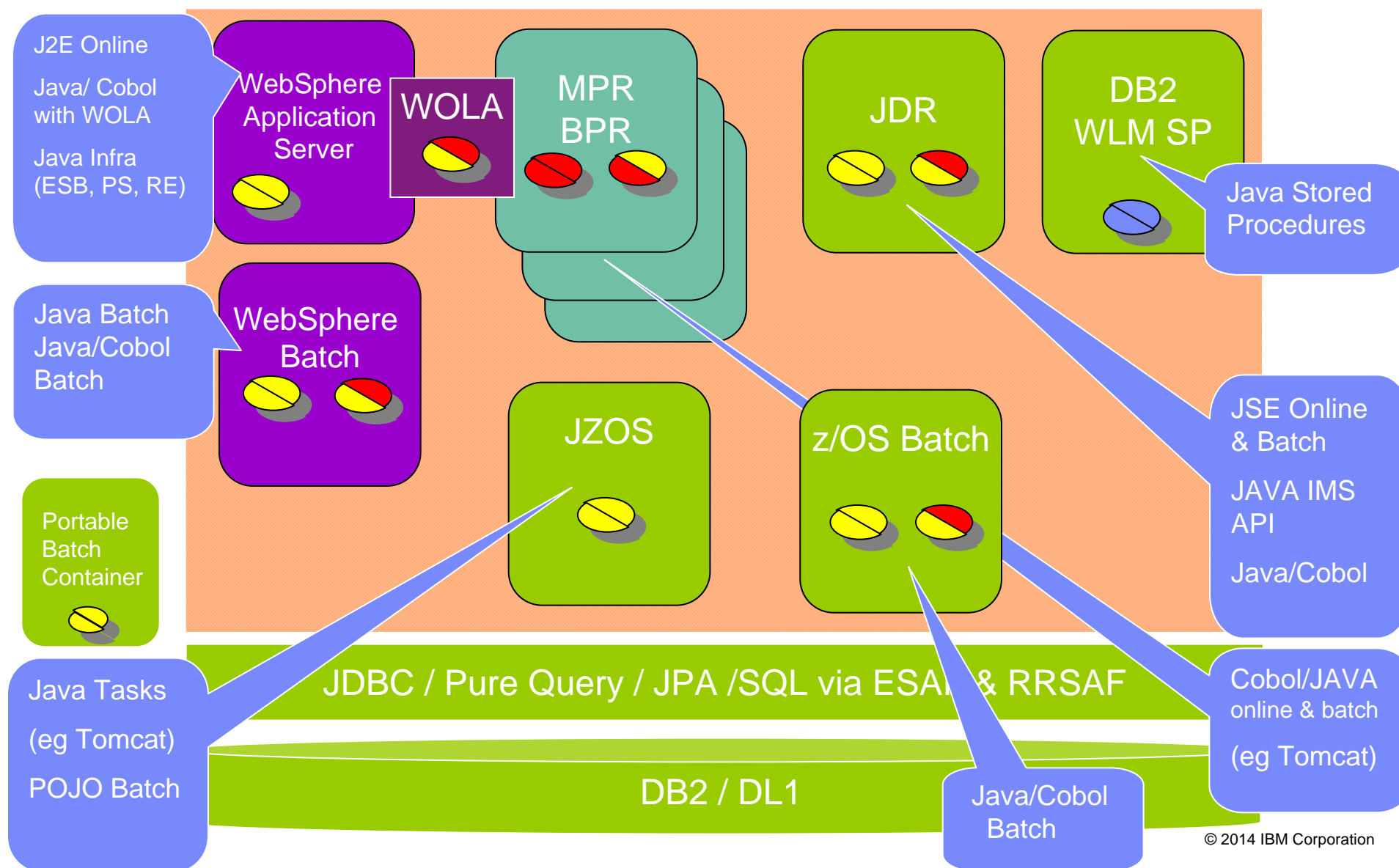
## Java and CICS

- Java is a first class Language in CICS
- The Communication is done via the same API as with all other languages (XCTL/Link with Channel / Container or Commarea)
- JDK 6.0.1 - 64bit
- OSGi Equinox FW
- JCICS API available

I ❤️ CICS



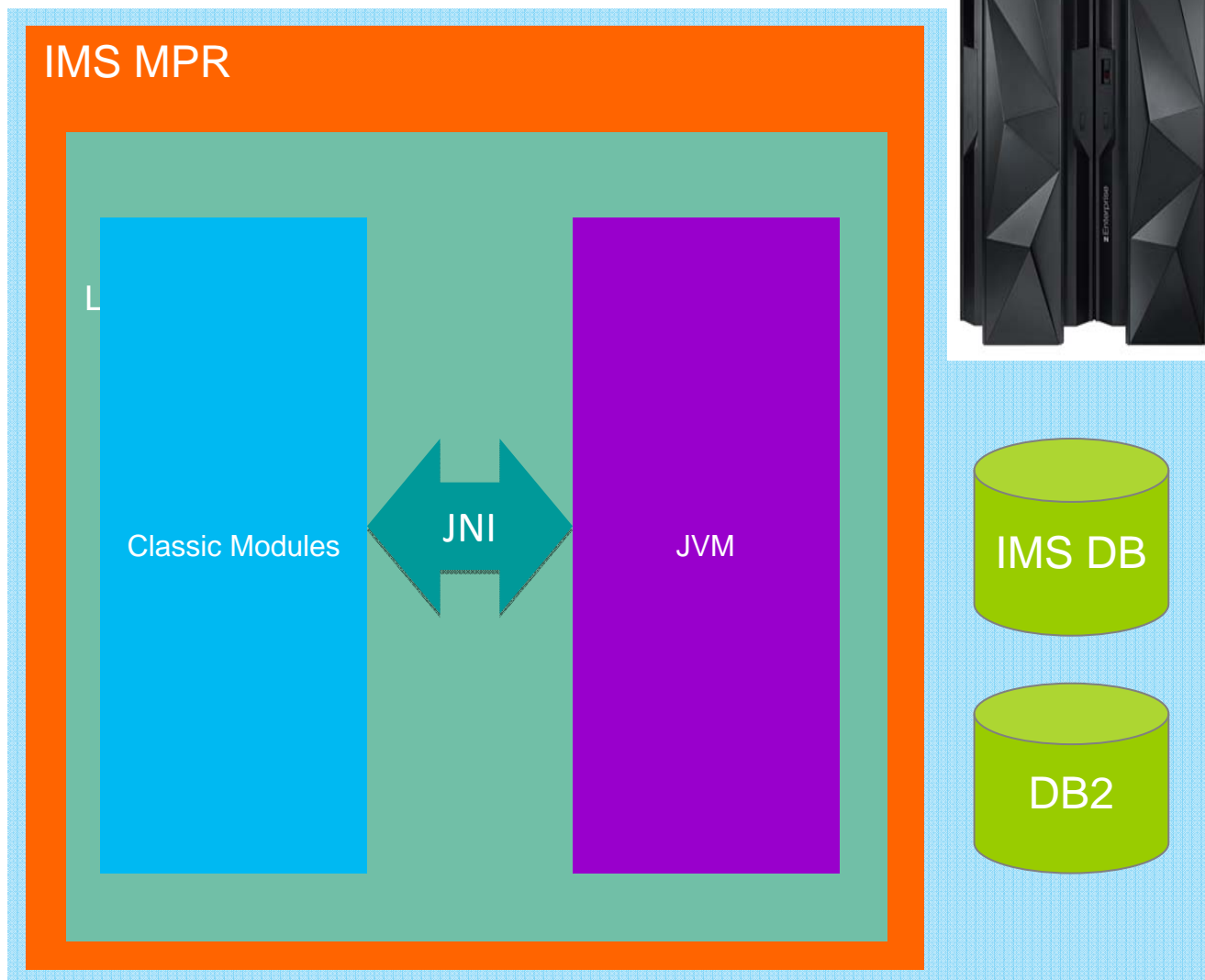
# Java and IMS



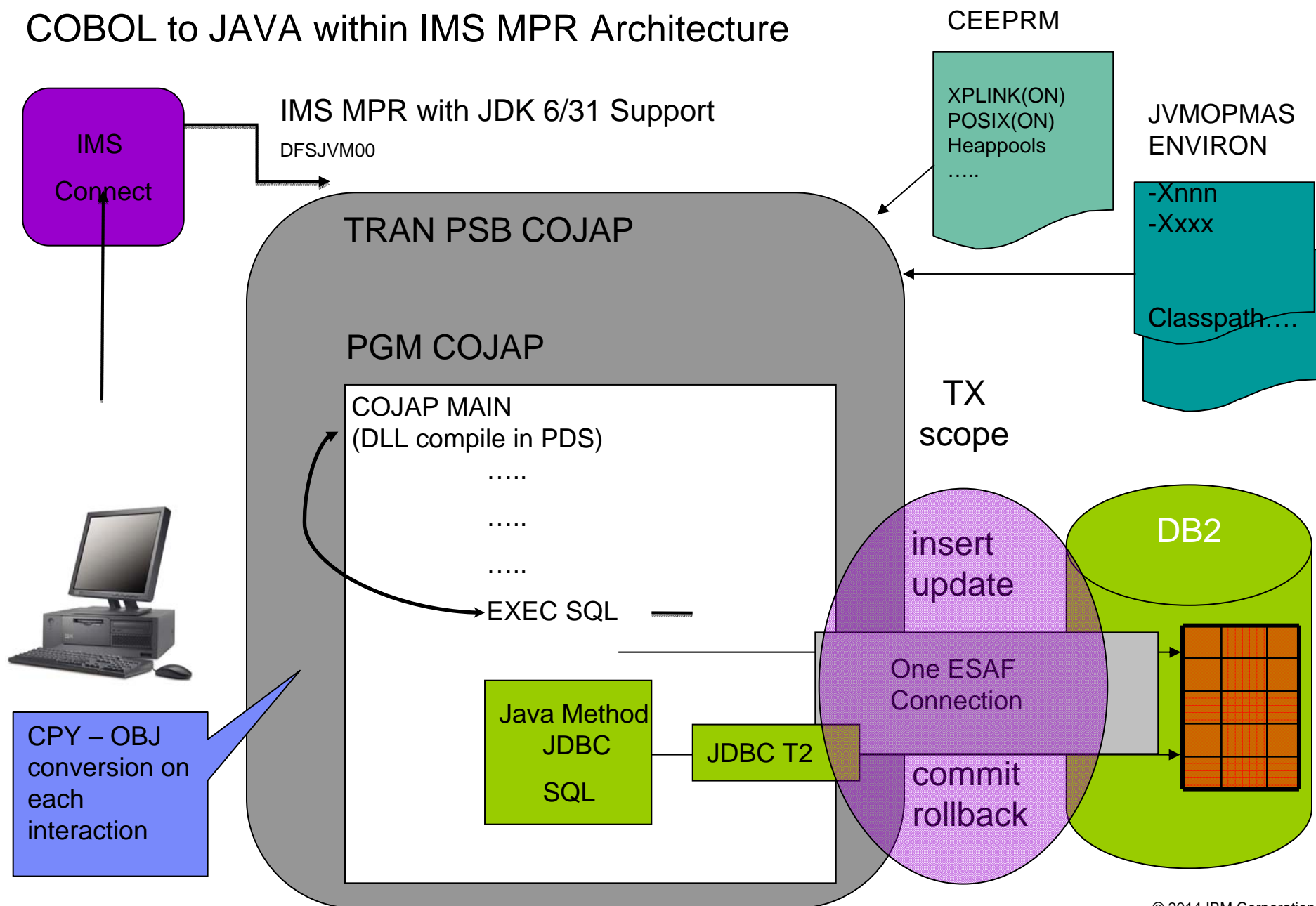
# Java and IMS on z/OS

- Java is a first class Language in IMS
- IMS guarantees a single UOR over Cobol-Java-Mix in an MPR (Message Processing Region)
  - One ESAF (External Subsystem Attach Facility) connection for both Cobol static SQL and JDBC based database accesses

**I ♥ IMS**

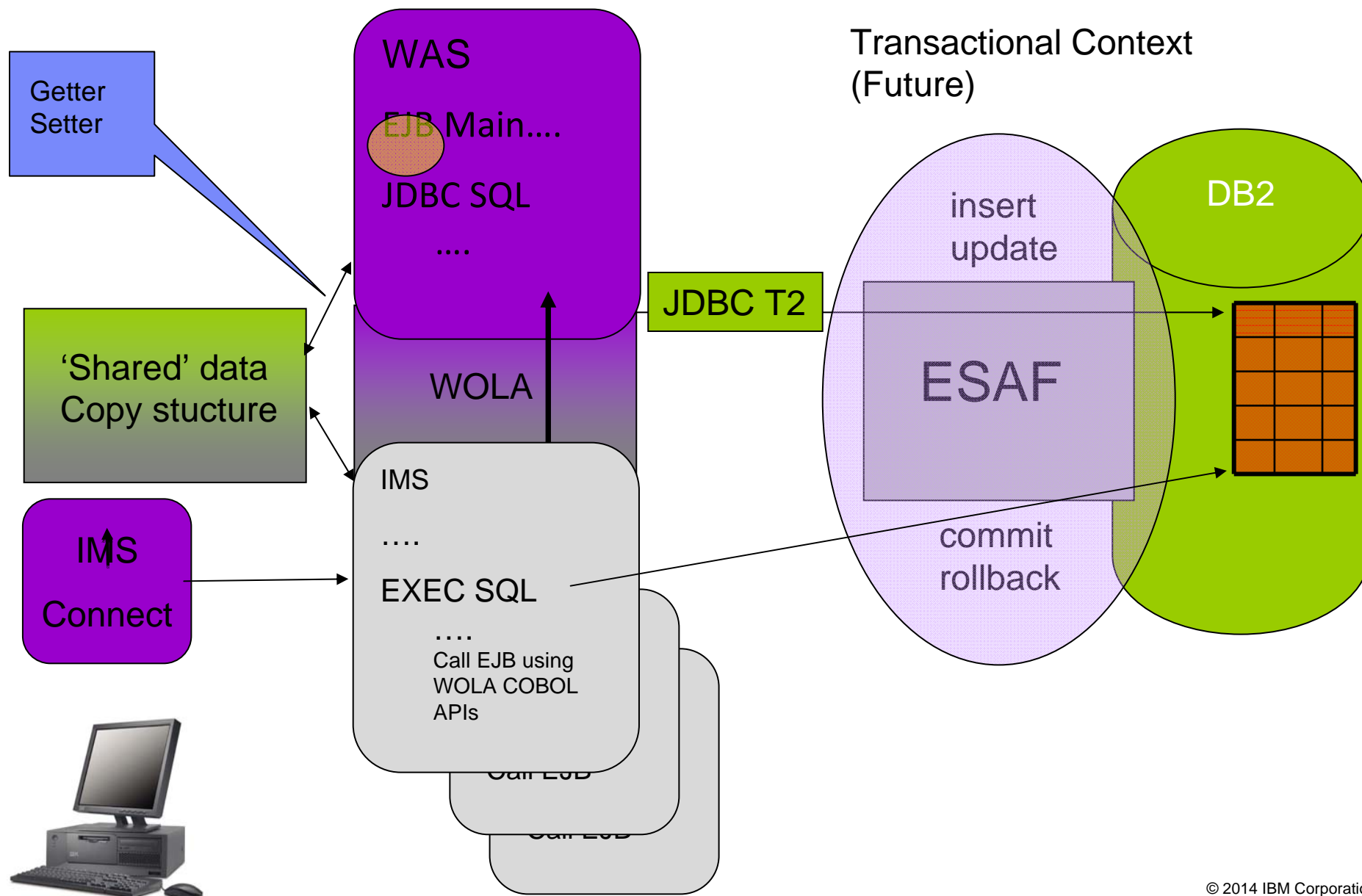


# COBOL to JAVA within IMS MPR Architecture





## Alternative: Accessing IMS thru WebSphere optimized Local Adapters





# JAVA (TM) auf System z – Notes

## Reasons for Java on z/OS

- Mainframe qualities out of the box
- Hardware, utilization & integration, z/OS, isolation, availability,
- 2 Phase Commit / support of transactions in IMS/CICS/zWAS
- Centralism & Governance
- Disaster Recovery
- Automation
- Sysplex Routing
- WLM support and Advisory
- Monitoring, Accounting, Reporting
- Shared Data architecture
- Address space and cross system Workload management (ENCLAVE)
- Modernization and vitalization of the Mainframe Platform

## Reasons for Java

- De facto Standard for open solutions
- Availability of
  - Experts, Skills, Education
  - Tools & Development environments
  - Frameworks
- Portable code
- Cost optimization

## Added value for JAVA on z/OS

- Best Security
  - Integrated security & auditing
  - Robust UNIX, no Admin/SU rights for Java
- Transactional robustness & efficiency with RRS
- Performance
  - Linear scalability, path length, response times
- Batch & Started Task abilities
  - Support of UNIX and MVS data
  - Whole JES & WebSphere Integration, console integration
- Integration / Extension/ Modernization of z/OS assets
  - f.e. Cobol/CICS/IMS etc Modernization
- Simple & robust deployment topologies
- Negotiating position & Price / Performance improvements

## More information on Java z/OS and zEnterprise

- z/OS Java web site
  - <http://www.ibm.com/systems/z/os/zos/tools/java/>
- IBM SDK Java Technology Edition Version 7 Information Center
  - <http://publib.boulder.ibm.com/infocenter/java7sdk/v7r0/index.jsp>
- IBM SDK Java Technology Edition Version 6 Supplement
  - <http://public.dhe.ibm.com/common/ssi/ecm/en/zsl03118usen/ZSL03118USEN.PDF>
- JZOS Batch Launcher and Toolkit Installation and User's Guide (SA38-0696-00)
- For JZOS function included in IBM Java SE 7 SDKs for z/OS
  - <http://publibz.boulder.ibm.com/epubs/pdf/ajvc0110.pdf>
- JZOS Batch Launcher and Toolkit Installation and User's Guide (SA23-2245-03)
- For JZOS function included in IBM Java SE 6 and SE 5 SDKs for z/OS
  - <http://publibfi.boulder.ibm.com/epubs/pdf/ajvc0103.pdf>
- IBM zEnterprise / System z Redbooks Portal: <http://www.redbooks.ibm.com/portals/systemz>
- IBM zEnterprise Announcement Landing Page: [ibm.com/systems/zenterprise196](http://ibm.com/systems/zenterprise196)
- IBM zEnterprise HW Landing Page: [ibm.com/systems/zenterprise196](http://ibm.com/systems/zenterprise196)
- IBM zEnterprise Events Landing Page: [ibm.com/systems/breakthrough](http://ibm.com/systems/breakthrough)
- IBM Software: [ibm.com/software/os/systemz/announcements](http://ibm.com/software/os/systemz/announcements)
- IBM System Storage: [ibm.com/systems/storage/product/z.html](http://ibm.com/systems/storage/product/z.html)
- IBM Global Financing: [ibm.com/financing/us/lifecycle/acquire/zenterprise/](http://ibm.com/financing/us/lifecycle/acquire/zenterprise/)
- Global Technology Services: <http://www.ibm.com/services/us/index.wss/offerfamily/gts/a1027714>

Grazie ITALIAN धन्यवाद HINDI *Merci* FRENCH ありがとうございます JAPANESE *Obrigado* BRAZILIAN PORTUGUESE 多谢 SIMPLIFIED CHINESE

Thank You

多謝 TRADITIONAL CHINESE Gracias SPANISH Спасибо RUSSIAN நன்றி TAMIL โอบพระคุณ THAI *Danke* GERMAN شكراً ARABIC

We appreciate your feedback.  
 Please fill out the survey form in order to improve this educational event.

---

## Backup

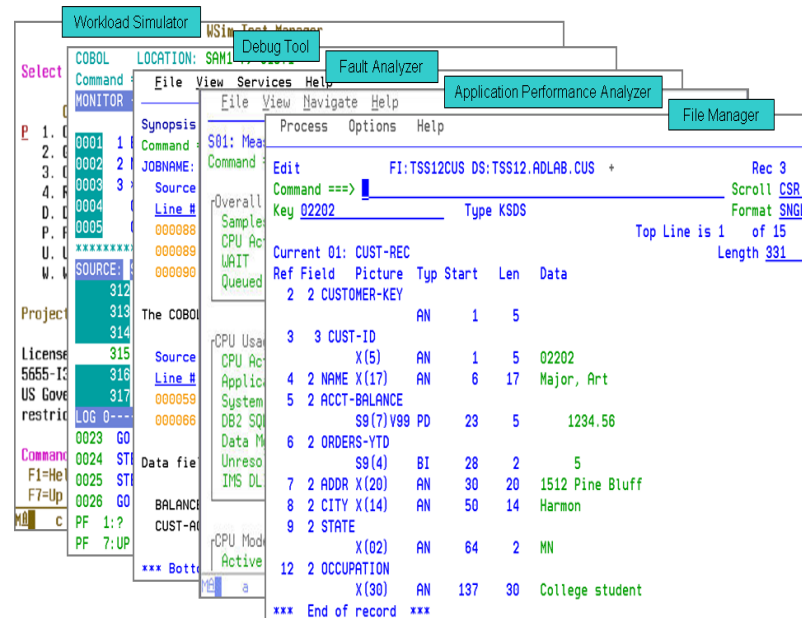
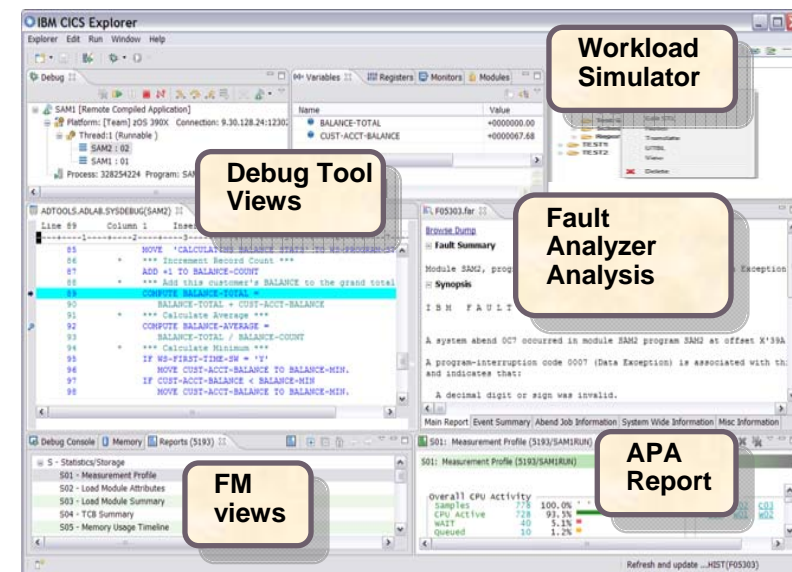
# Problem Determination Tools V13 - Five Major One Time Charge Products

## KEY PD TOOLS and Functions:

- **File Manager for z/OS** Data management tool supporting key file structures like VSAM, DB2, CICS, and IMS
  - New – DB2 and CICS Eclipse GUI Plug-Ins
- **Debug Tool for z/OS** Source code debugging to improve development productivity
  - New - Smooth debugging mixed apps JAVA/COBOL/PLI
- **Fault Analyzer for z/OS** Helps you rapidly pinpoint cause of failed application (program abend)
  - New – JAVA JVM Improvements
- **Application Performance Analyzer for z/OS** Monitor performance at the application level
  - New - Expanded JAVA Support (JVM, WAS, CICS, IMS)
- **Workload Simulator for z/OS and OS/390** Regression and Load testing of interactive z/OS applications
  - New – Initial Eclipse GUI Plug-Ins

## Other Related Tools:

- Common components via CICS Explorer or IMS Explorer – including common login facilities and much more...
- **Rational Developer for System z** – PD Tools GUI front end plugs into RDz for a complete developer workbench for Cobol and Java
- **Hour Glass** – date time manipulation
- **ISPF Data Set Commander** – less key strokes for ISPF
- **Optim TDM** – build of relational test data, subsets of production data, scrambling

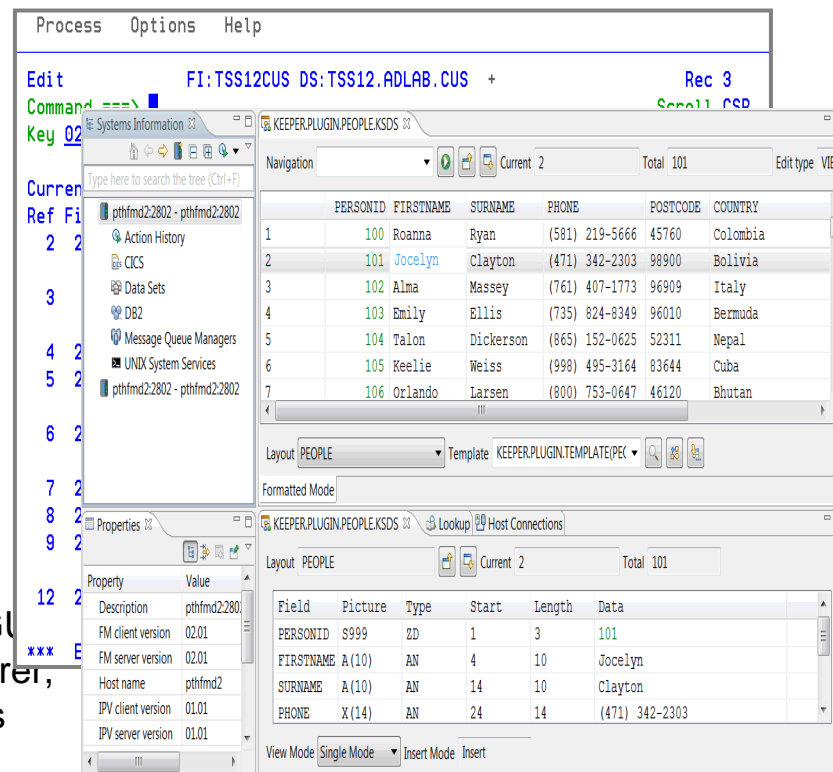


I ♥ PD Tools

## Manage a variety of enterprise data file structures

### IBM File Manager Version 13!

- **IBM File Manager** allows you to manage production, test, and development data across multiple formats and storage media.
- Create, edit, copy, browse, extract, print, and compare enterprise data.
- Proven 3270-based interface and free graphical user interface.
- GUI plug-in support for File Manager,DB2,IMS,MQ
- GUI plug-in for File Manager/CICS
- Support for IMS V12, DB2 V11. MQ V8, COBOL V5
- Formatted KSDS key
- Functional, usability, and security enhancements to the GUI plug-in that include improved integration with CICS Explorer, IMS Explorer, and the other Problem Determination Tools family of plug-ins
- A range of other enhancements to address key customer requirements

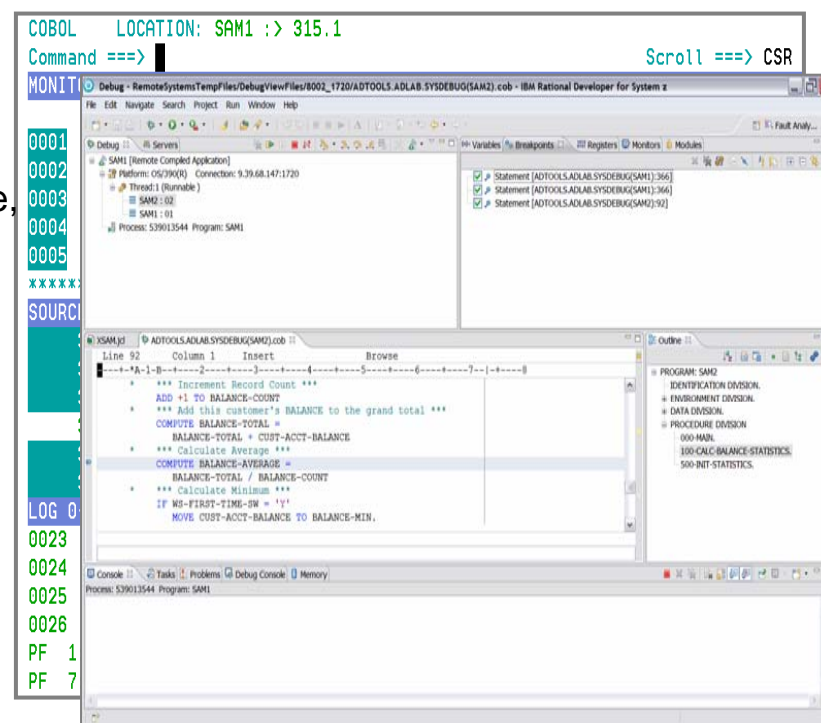


*Data management tool  
supporting key file structures  
like VSAM, DB2, CICS, and  
IMS*

## Debugging of enterprise applications

### IBM Debug Version 13!

- **IBM Debug Tool** can help you increase debugging efficiencies and reduce application development cycle times.
- Program testing and analysis aid that helps you examine, monitor, and control the execution of application programs on z/OS.
- Proven 3270-based interface and free graphical user interface.
- Smooth debugging mixed apps JAVA/COBOL/PLI
- Enhanced performance for automonitor
- Support for CICS TS V5.1, IMS V13, DB2 V11
- Functional, usability, and security enhancements to the GUI plug-in that include improved integration with CICS Explorer, IMS Explorer, and the other Problem Determination Tools family of plug-ins
- A range of other enhancements to address key customer requirements
- COBOL V5



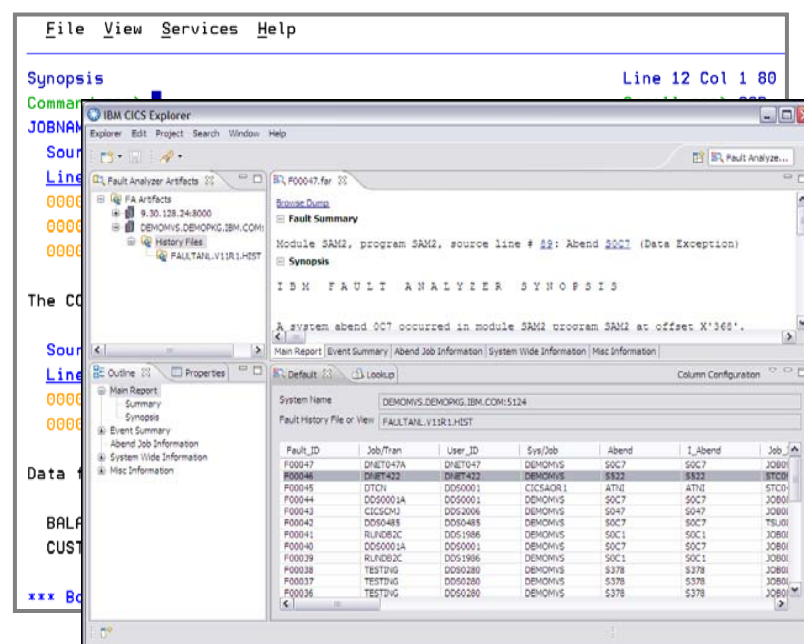
*Source code debugging to  
improve development  
productivity*



## Pinpoint the cause of enterprise application failures

### IBM Fault Analyzer Version 13!

- **IBM Fault Analyzer** improves developer productivity and decreases deployment costs by helping to analyze and correct application failures quickly.
- Develop and test new and existing applications more productively, helping to reduce costs along the way.
- Proven 3270-based interface and free graphical user interface.
- Enhanced Java support
- Support for CICS TS V5.1
- Functional, usability, and security enhancements to the GUI plug-in that include improved integration with CICS Explorer, IMS Explorer, and the other Problem Determination Tools family of plug-ins
- A range of other enhancements to address key customer requirements
- COBOL V5

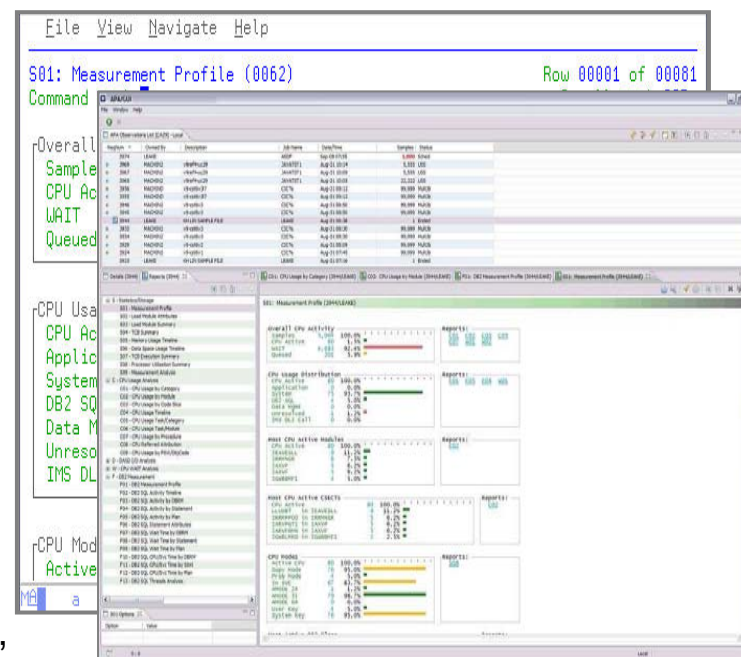


*Helps to identify the cause, analyze the failure, and fix the problem*



## Pinpoint enterprise application bottlenecks *IBM Application Performance Analyzer Version 13!*

- **IBM Application Performance Analyzer** helps maximize the performance of your applications and improve the response time of your online transactions and batch turnaround.
- Identify constraints and improve the entire application's performance no matter where the problem resides (CICS, IMS, DB2, MQ, COBOL, PLI, ASM, JAVA, WAS, COBOL5)
- Proven 3270-based interface and free graphical user interface.
- Enhanced support for WebSphere and Java
- Sampling enhancements to improve accuracy of reports
- Support for CICS TS V5.1, IMS V13, Java V7.0, MQ V7
- Functional, usability, and security enhancements to the GUI plug-in that include improved integration with CICS Explorer, IMS Explorer, and the other Problem Determination Tools family of plug-ins

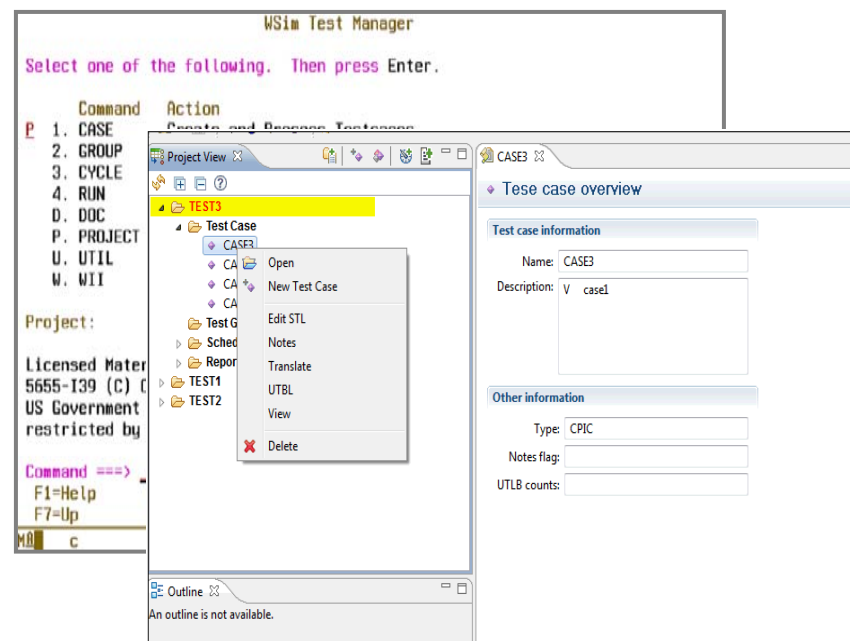


*Monitor and optimize  
performance at the application  
level*

## Regression and load test interactive z/OS applications

### **IBM Workload Simulator Version 1.1.0.1 GUI Changes!**

- **IBM Workload Simulator** helps eliminate the need for large amounts of terminal hardware and terminal operator time for testing.
- Ideal for stress, performance, regression, function, and capacity planning tests.
- Proven 3270-based interface and new free graphical user interface.
- Initial release of free graphical user interface (GUI) provides access to the power of the host product



*Simulates a network of terminals  
and the associated messages*