#SHAREorg

SHARE
Technology · Connections · Results

WebSphere Application Server for z/OS Version 8 Overview and Update

(Session 10560)

David Follis
IBM
z/OS Runtime Architect

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
10560	Monday	9:30	International Ballroom F	Version 8 – Overview and Update	David Follis
10580	Monday	11:00	Cottonwood A/B	Back to Basics	Mike Loos
10633	Wednesday	1:30	International Ballroom C	Installation Manager – The Cross Platform Installer for WAS	Mike Loos
10561	Wednesday	3:00	Cottonwood A/B	Version 8 – New z/OS Exploitation Features	David Follis
10562	Thursday	11:00	Cottonwood A/B	Batch Update	John Hutchinson
10581	Thursday	1:30	Cottonwood A/B	Getting Started with Version 8 – Part Zero!	Mike Loos
10518	Thursday	6:00	Cottonwood A/B	Potpourri	Anybody
10516	Friday	8:00	Dogwood B	Level 2 Update	Mike Stephen
10563	Friday	9:30	Pine	Hands on Lab	Mike Stephen, David Follis, Ken Irwin

#SHAREorg

SHARE
Technology · Connections · Results

Version 8 Overview

Versions, Dates, and Service Levels...

	GA	End of Marketing	End of Support
Version 6.0	3/25/2005	2/23/2009	9/30/2010
Version 6.1	6/30/2006	7/25/2011	**9/30/2012**
Version 7	9/26/2008		
Version 8	6/17/2011		

- End of Service announced for V6.1
- GA of Version 8
- Service level naming convention change:
  - V6.1 and V7 – even numbers z/OS only, odd numbers common
  - V8 – all levels are common (no more z/OS only levels)

The Key Technical Messages

Further refinement of proven base

- Updated specifications (Java EE 6, EJB 3.1, Servlet 3.0, JSP 3.0, etc)
- Enhanced web services: JAXB, JAX-RS, JAX-WS
- Feature packs rolled in: SCA, XML, OSGi, JPA, WOLA, Batch
- Continued focus on tightening and refining programming

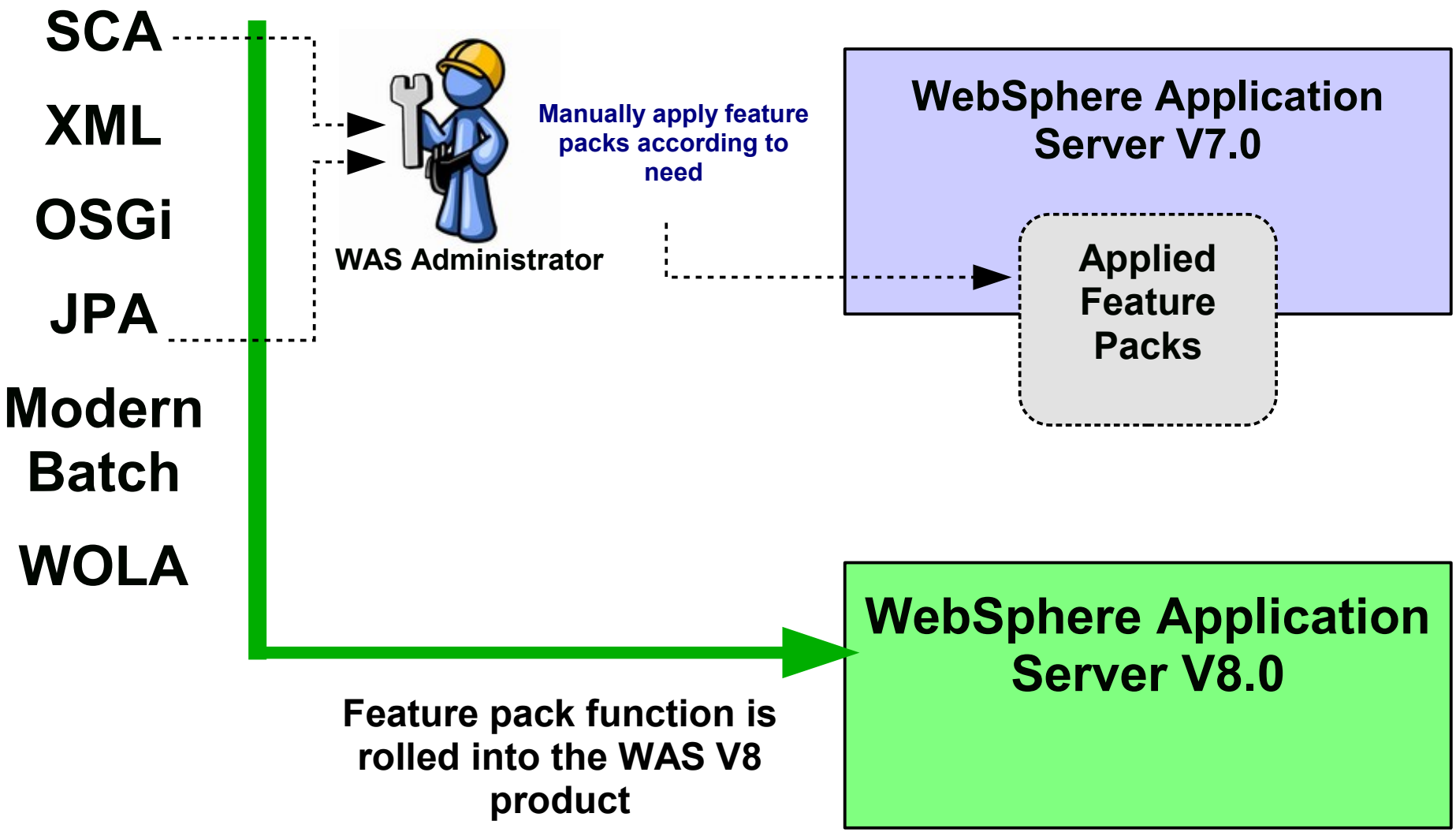
Enhanced commonality across platforms

- Common installation process using Installation Manager
- Common high-performance logging function

Greater z/OS exploitation

- More granual settings for timeouts and tracing w/ dynamic control
- Resource routing for data source and connection factories

Rollup of Prior Feature Packs



New Version = Updated/New Specs

Best place to understand the new specs is to go to the InfoCenter and search on `rovr_specs`

Network Deployment (z/OS), Version 8.0 > Reference > Supported configurations and limitations

Specifications and API documentation

The WebSphere® Application Server product supports various ir
specifications and application programming interface (API) docu
previous product releases.

Components

- Any application type
- Web applications
- Portlet applications
- SIP applications
- EJB applications
- Client applications
- Web services
- Service Component Architecture
- Service integration
- Data access resources
- Messaging resources
- Mail, URLs, and other Java EE resources
- Security
- Web Services Security
- Naming and directory
- Object Request Broker
- Transactions
- WebSphere extensions
- Administration

Any application type

Table 1. Supported specifications for any application type. The product supports the specifications or APIs in this table.

Specification or API	Version 8.0	Version 7.0	Version 6.1	Version 6.0
Java™ Platform, Enterprise Edition (Java EE) specification Prior to Java EE 5, the specification name was Java 2 Platform, Enterprise Edition (J2EE).	Java EE 6 (JSR 316) New Java EE 5 J2EE 1.4 J2EE 1.3	Java EE 5 New J2EE 1.4 J2EE 1.3	J2EE 1.4 J2EE 1.3 J2EE 1.2	J2EE 1.4 New J2EE 1.3 J2EE 1.2
Java Platform, Standard Edition (Java SE) specification Prior to Java SE 6, the specification name was Java 2 Platform, Standard Edition (J2SE).	Java SE 6	Java SE 6 New	J2SE 5	J2SE 1.4.2
ISO 8859 specifications	ISO 8859 applies to these versions.			

Java EE 6 (JSR 316)

Continues trend towards increased function and a simpler development model

Java Servlet 3.0 (JSR 315)

Enhancements to support modern web development

EJB 3.1 (JSR 318)

Further simplifies development of EJBs

JCA 1.6 (JSR 322)

Update specification architecture based on feedback from experts and users

What's new in WAS V8 Performance

- Improved performance with Java 6
  - JVM runtime enhancements
  - JIT optimizations
- JPA Improvements
  - Optimizations to ObjectCache
  - Fine-grained locking mechanisms
  - Optimized Query cache
  - Overall pathlength reduction across JPA 2.0 engine
- Improved Web Services performance
  - Improved parser performance for JAXB marshalling
- Startup time and memory footprint improvements
  - Java 6 updates and increased shared class cache reduces memory footprint
  - Class loading improvements reduce startup time

Java, JVM, and Platform-Awareness

Version 8 still provides Java 6 (though called “6.0.1”). What’s new is the JVM inside the supplied Java, and the z196-awareness:

Application

Application still “sees” Java 6, but benefits from the performance enhancements below

Java 6 Specification Definition

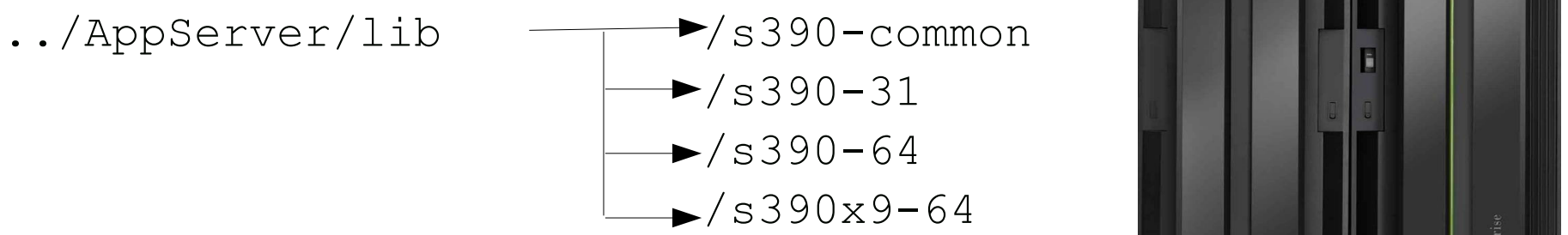
Java Virtual Machine

z-Aware Native Layer

- Significant enhancements to JIT optimization technology
- New Balanced GC policy to reduce max pause times
- z196 exploitation of instructions and new Out-of-Order-Execution pipeline

z196 processor has faster chips, larger and better cache, an Out-of-Order-Execution pipeline and additional instructions

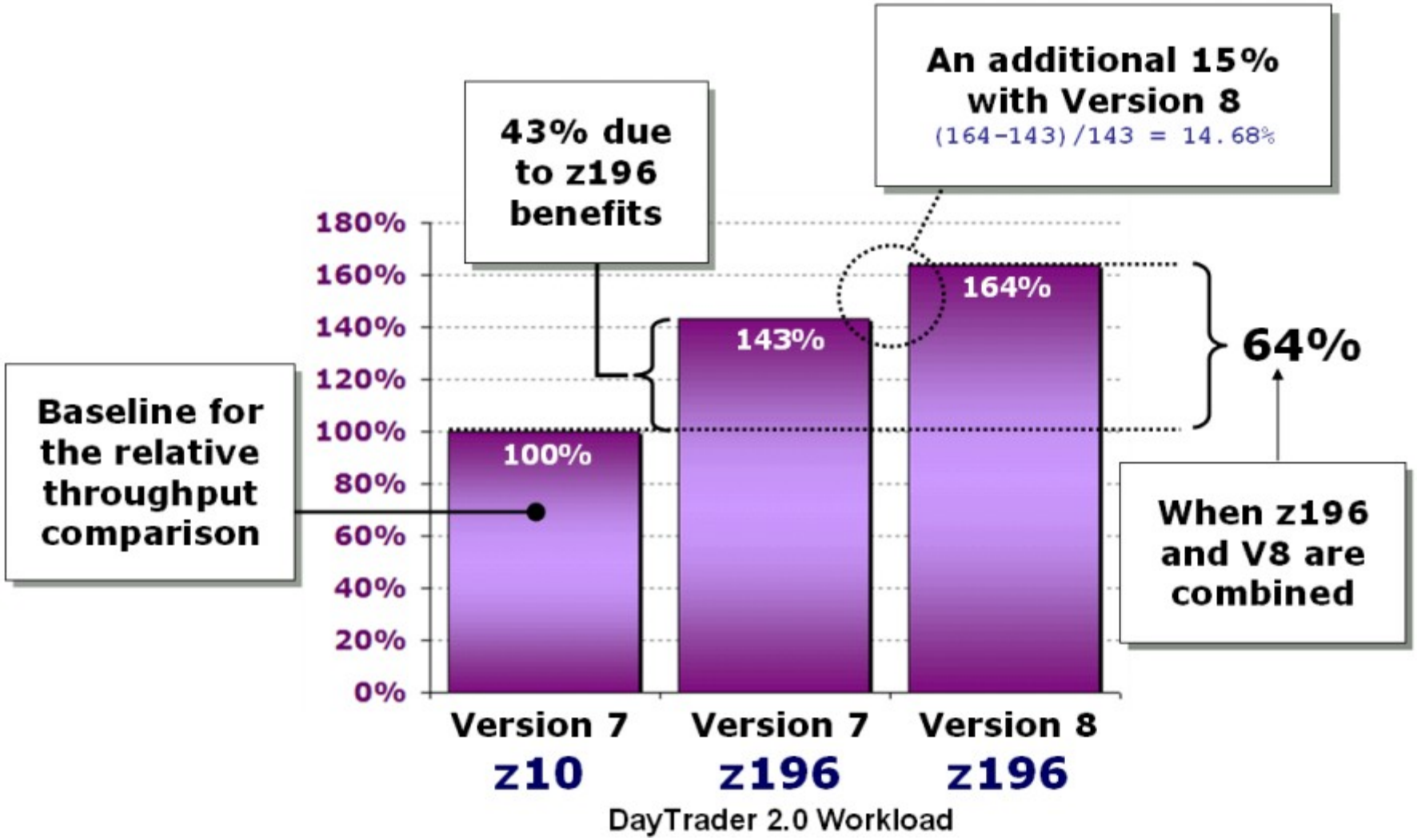
Native DLLs optimized for z196



/s390-common	/s390-31	/s390-64	/s390x9-64
Native modules required for any bitmode and hardware	Native modules required for 31 bit	Native modules required for 64 bit and any pre-z196 hardware	Native modules required for 64 bit optimized for z196

WAS V8 Performance at High Level

Three components to performance illustrated -- WAS V8 itself; improvements in JVM that comes with WAS V8; and benefits from the z196 processor:

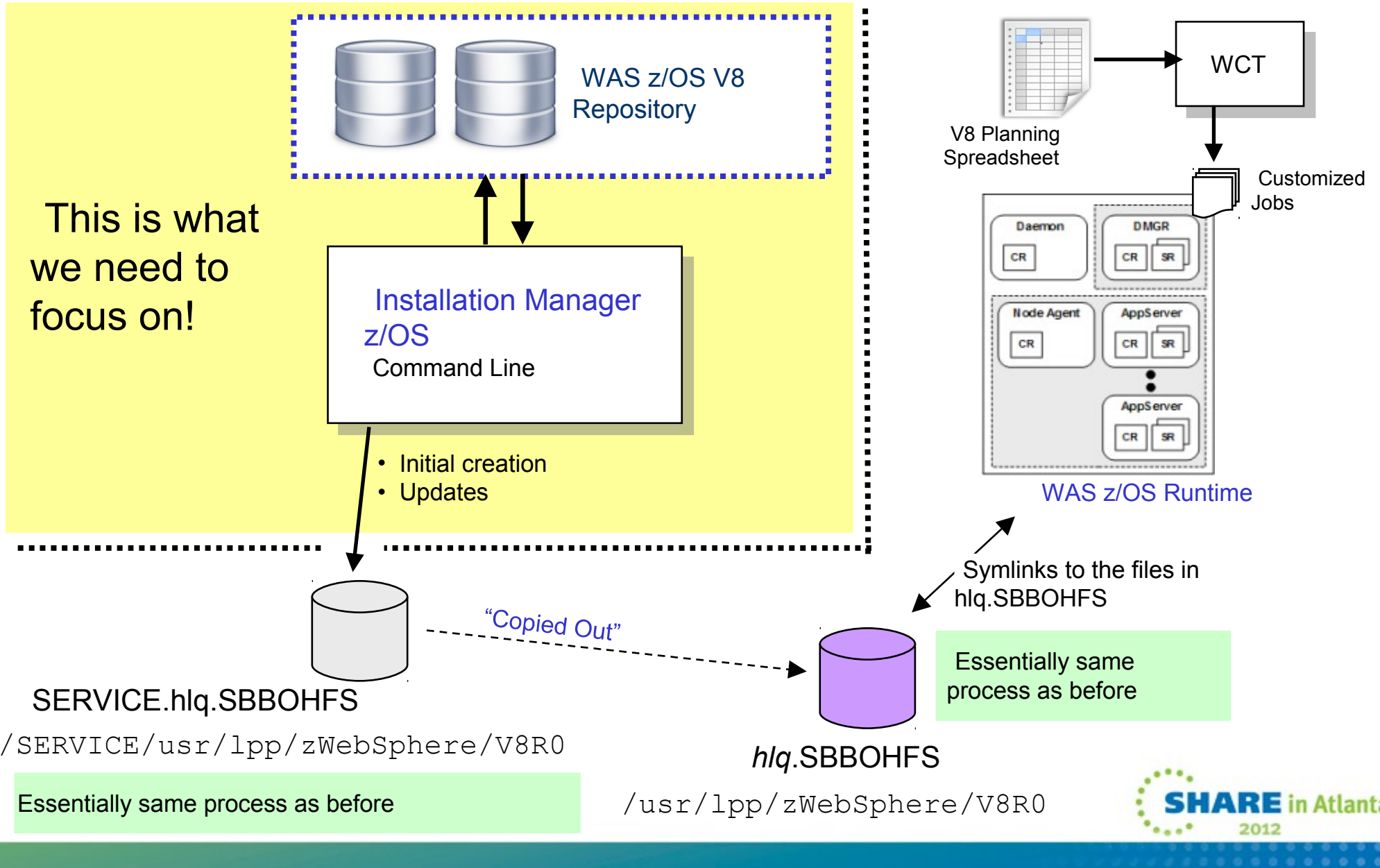


Performance results provided here are not a guarantee of performance.
Results vary depending on many complex factors.

Install

IM 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:



The WAS V8 Product Repository

We have a two-part story here: one for the initial product delivery, then the opportunity to use IBM's hosted repository for fixpacks/feature packs

Initial Acquisition of WAS z/OS V8 from IBM

SMP/E installable
Local Repository

SMP/E Process

CSI

Local Repository
(Mounted File System)

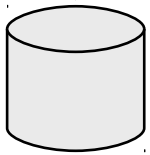
Fixpack and Feature Pack Updates

Internet

IBM-hosted Repository

/InstallationManager/V1R4

Installation Manager
z/OS
Command Line



SERVICE.hlq.SBBOHFS
/SERVICE/usr/lpp/zWebSphere/V8R0

#SHAREorg

SHARE
Technology · Connections · Results

High Performance Extensible Logging (HPEL)

Very High Level Overview of HPEL

This is an optional mechanism to format traces and logs into a WAS binary format. A utility is then used to offload to a viewable text file:

Configurable on a Server-by-Server Basis

WebSphere Application Server Version 8
All Supported Platforms

Java Trace

Java Logs

System.out

System.err

More efficient use of space, faster write operations

WAS-specific binary format log file

Write to memory buffer, then file

Controls to dictate size limits, what to do when limit reached, how to trim files, start new files, etc.

Log Viewer Utility

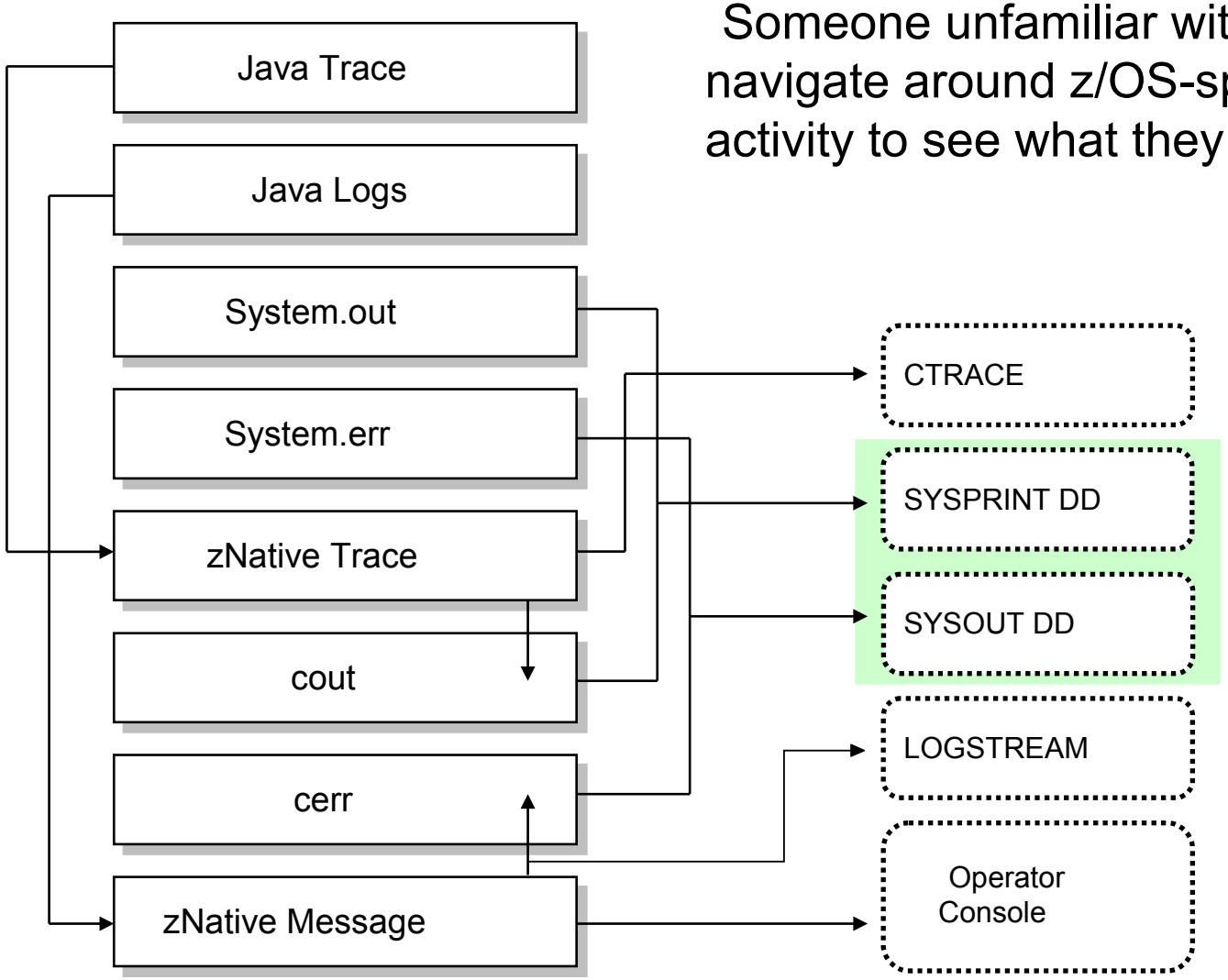
Common across all platforms

Use whatever view/edit tool you prefer

ASCII readable file

For z/OS: Output We Have Today

This is known as “Basic” in Version 8. On the next chart we’ll see where HPEL affects the picture

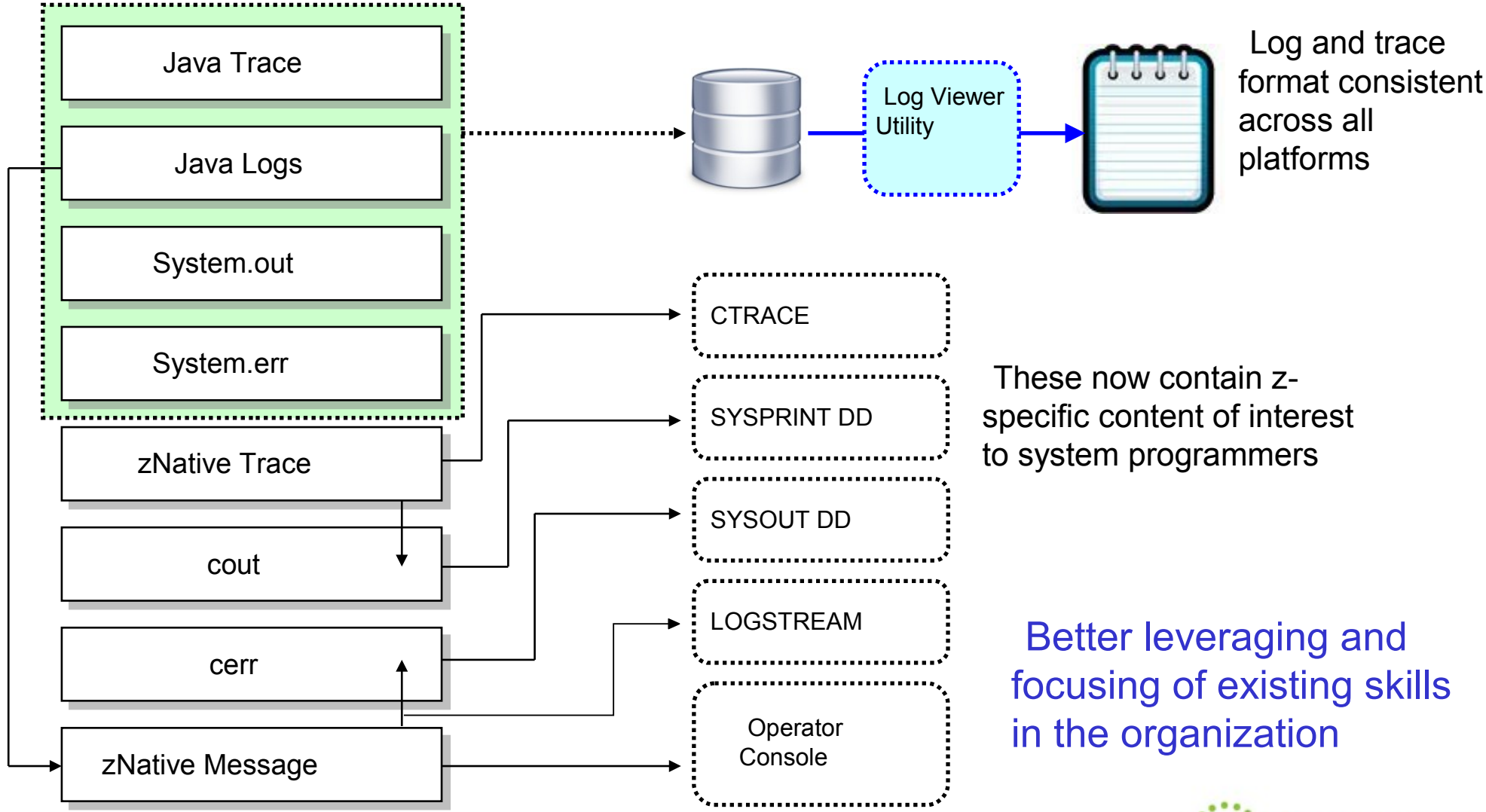


Someone unfamiliar with System z had to navigate around z/OS-specific log and trace activity to see what they needed to see

This portion could be routed to JES or USS/MVS output

For z/OS: With HPEL in Effect for Server

And here's what it looks like when you put HPEL into effect:



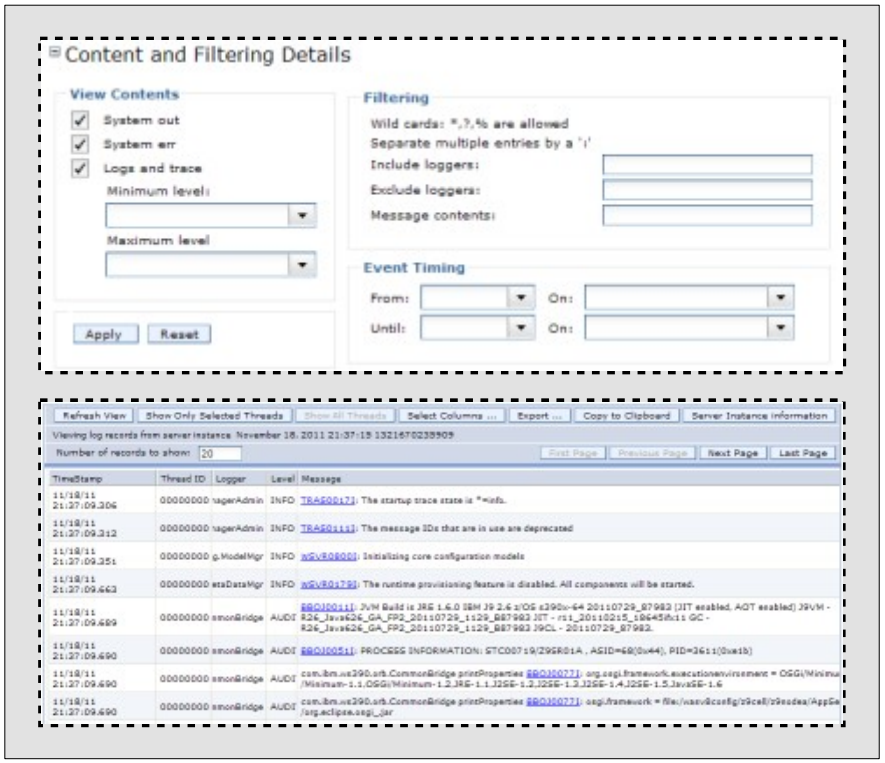
Log and trace format consistent across all platforms

These now contain z-specific content of interest to system programmers

Better leveraging and focusing of existing skills in the organization

High-Level of the Admin Console Log Viewer

This is a graphical log viewer supplied as part of the Admin Console:



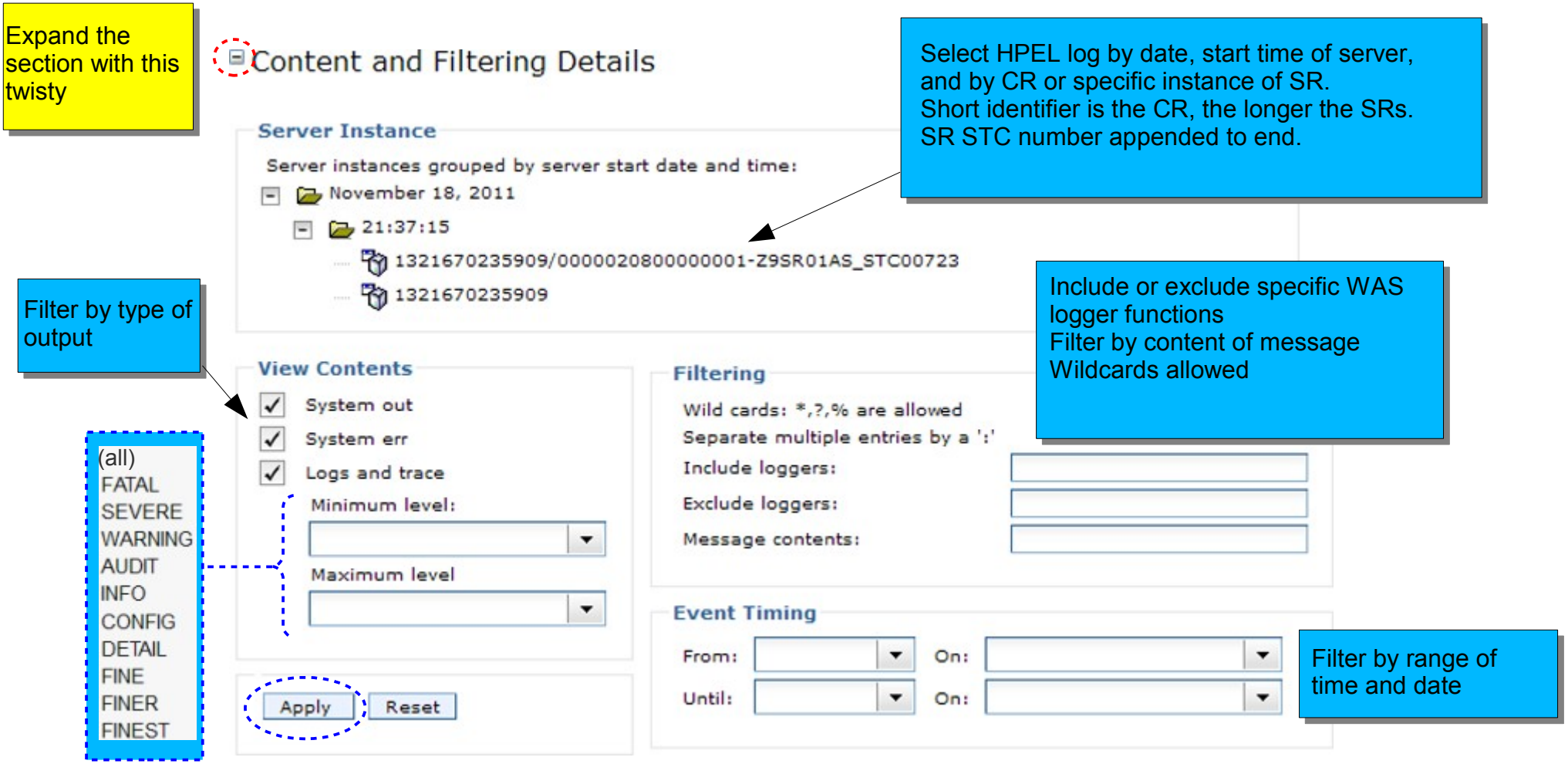
Log selection and content filtering section. This influences what appears below ...

... record by record display of HPEL content based on filtering down above.

Content and filtering details …

Content and Filtering Details

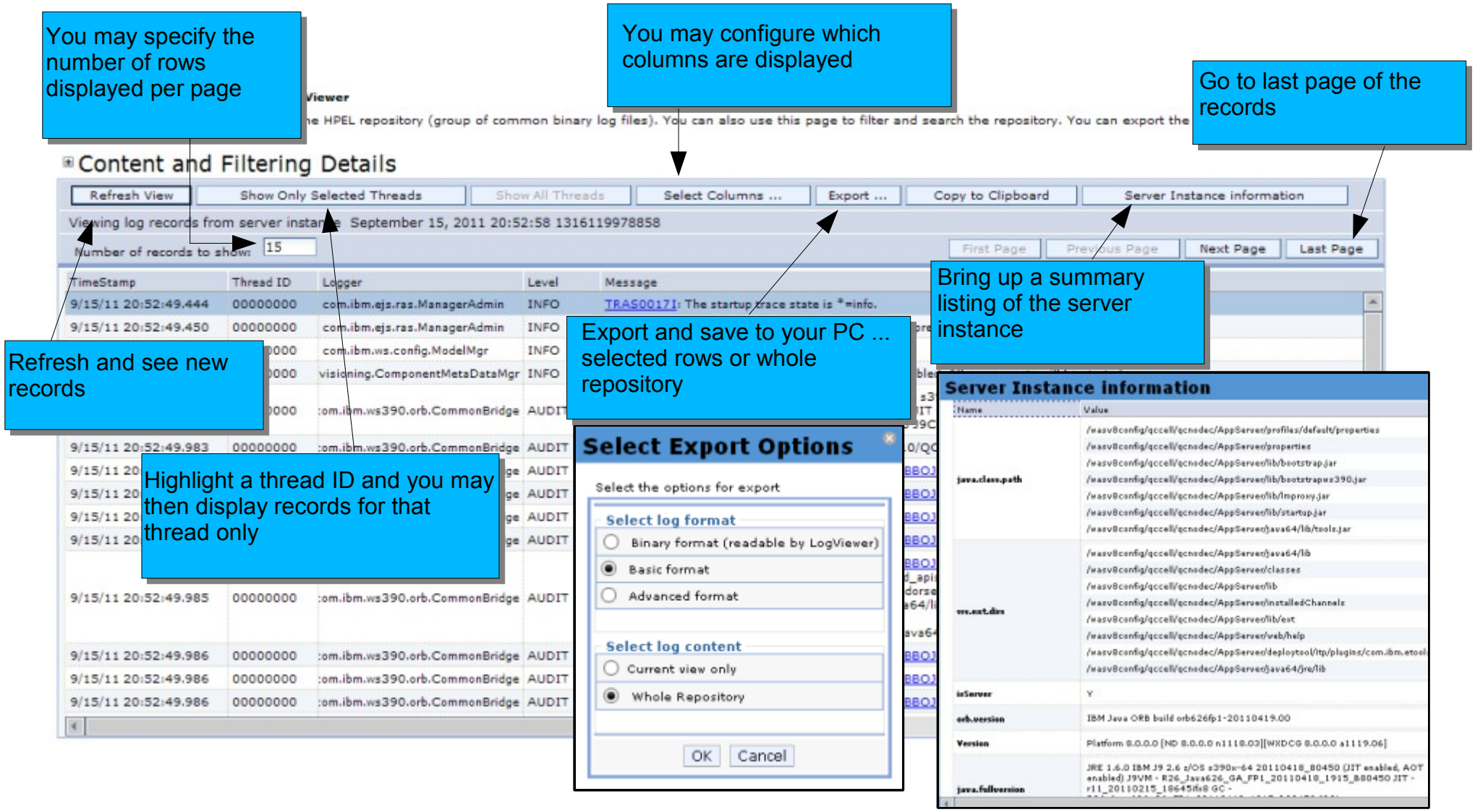
This allows you to determine, with a fair degree of granularity, what HPEL records will be displayed in the output result set:



"Apply" will put into effect your filtering selection

Record Display of Content Based on Filtering

This displays in your Admin Console for the selected server's HPEL log:

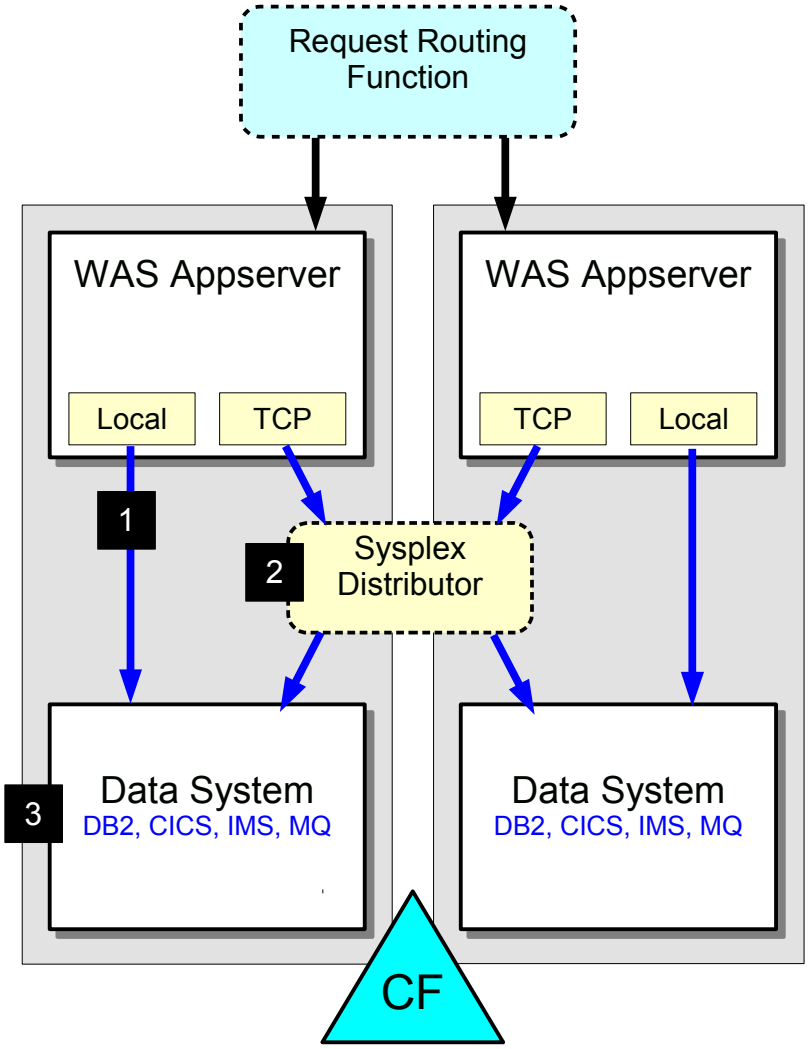


Example …

High Availability Connection Management

Data Access Approaches - The T2 vs. T4 Debate

You gain a degree of flexibility with an TCP-based connection but lose some of the advantages of a local cross-memory connection:



1. Local Connectors

Uses the cross-memory native interfaces. Available for DB2, CICS, IMS and MQ.

Advantages: Speed, avoid serialization, assert identity, single thread of execution, propegate enclave for DB2

Disadvantages: Loss of data system means application has no access to data unless alternative connections are made available. Routing function may not know backend data system is gone.

2. TCP-based Connectors

Uses the TCP network to flow requests to target listener. Available for DB2, CICS, IMS and MQ.

Advantages: Loss of TCP connection typically signals retry; SD will connect to surviving member. DB2 T4 takes this even further.

Disadvantages: Potential loss in performance, generally implies alias ID and PW.

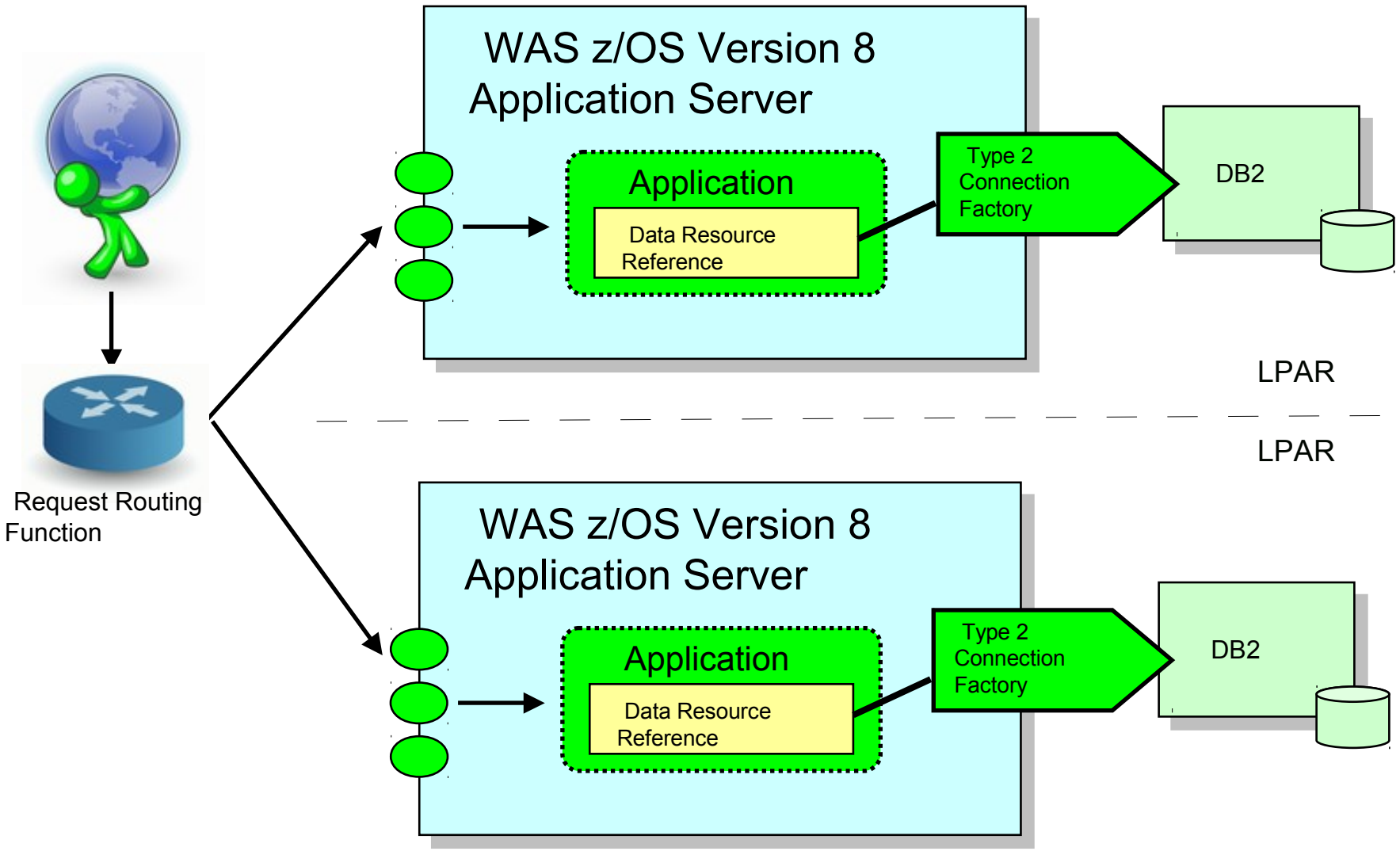
Data locks may exist ... other work may proceed but work related to held data can not until failure subsystem restarted so locks can be freed.

3. What is Being Protected Against?

If you are concerned about loss of data system while WAS server stays up, then the TCP based with intermediate routing is a consideration.

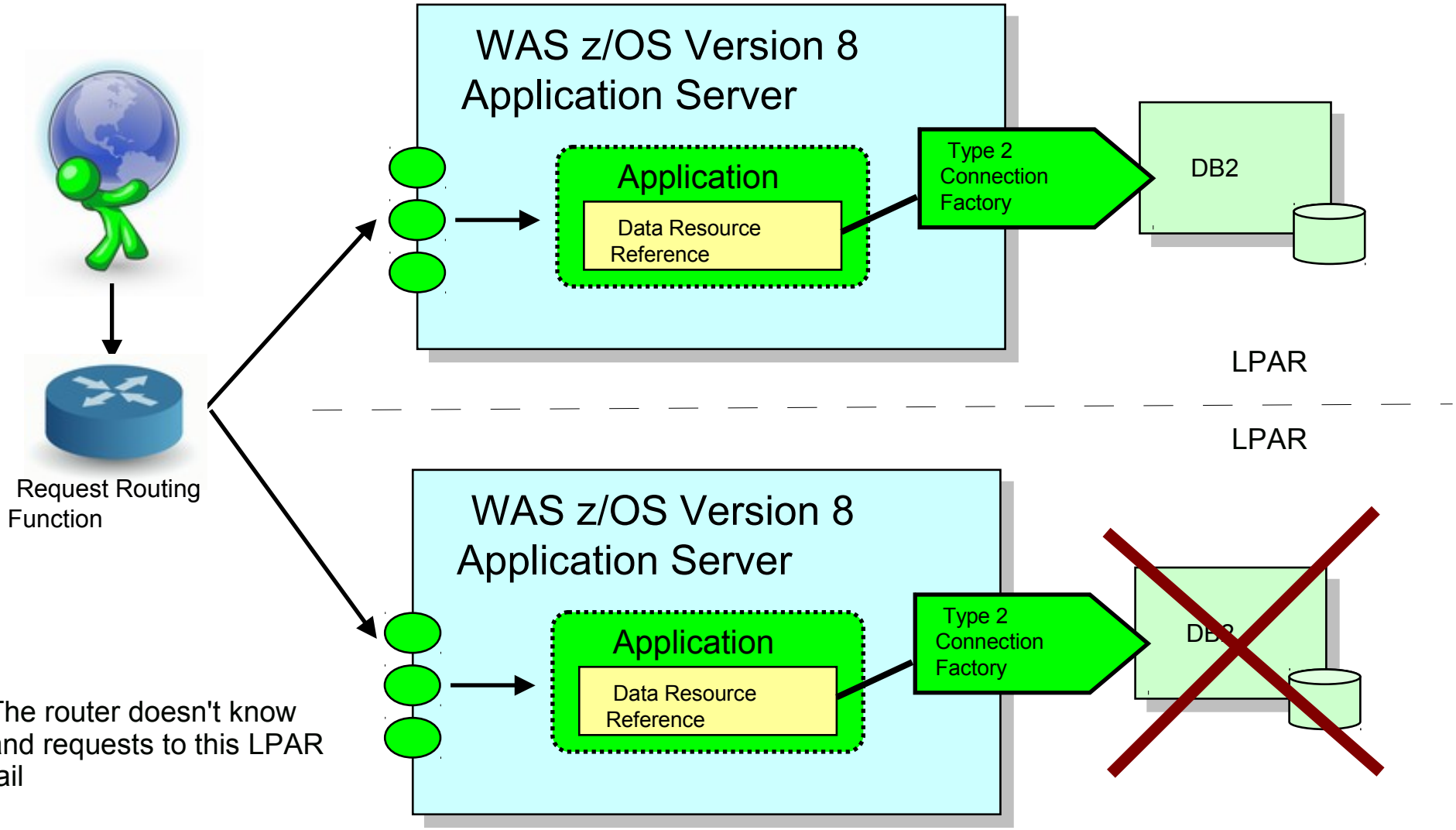
If your primary concern is loss of the LPAR "tower" then the use of TCP connectors becomes less important.

A Typical Clustered Environment with Type-2 Connectors



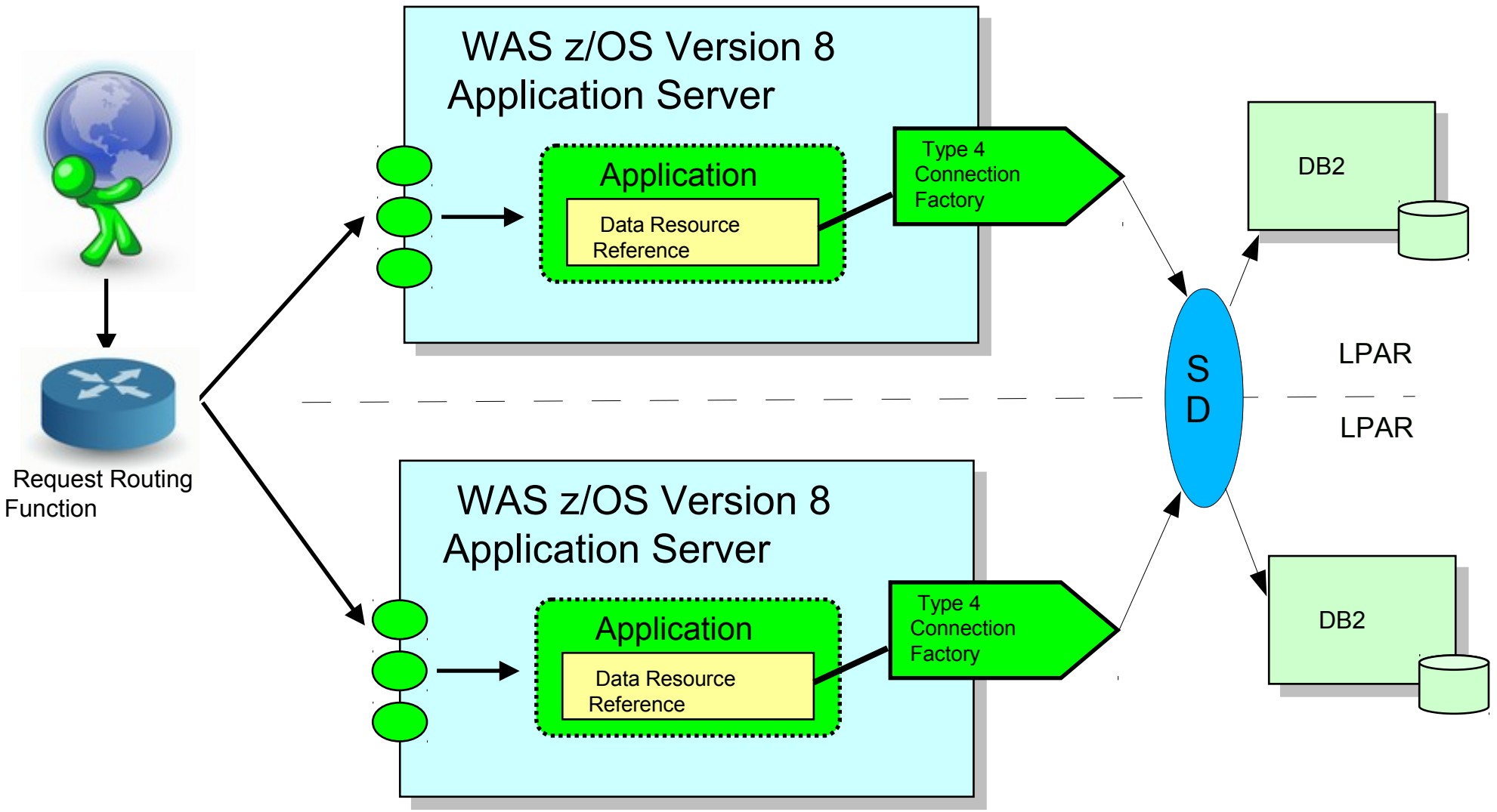
Just using DB2 as an example..

When something bad happens....



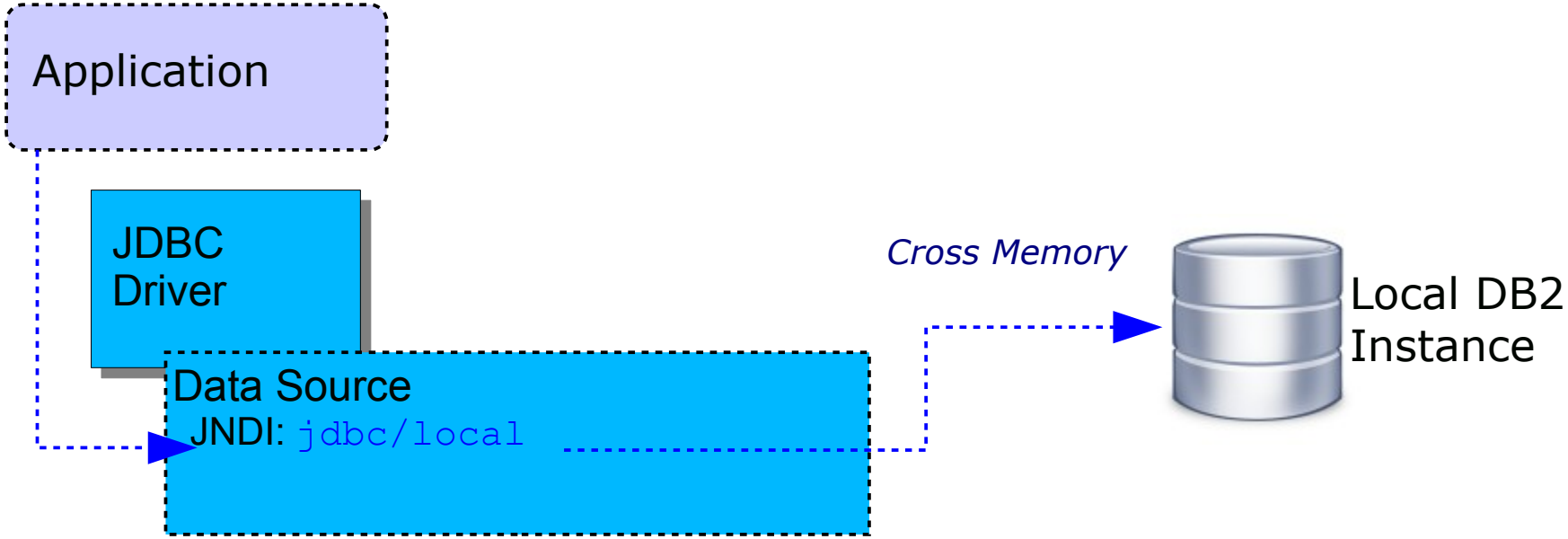
The router doesn't know and requests to this LPAR fail

A Common Solution...



Use Type-4 Connectors and Sysplex Distributor to eliminate close coupling between WAS and DB2... But this surrenders the value of co-location!

Accessing JDBC Resources



Connection Management

The component of WAS that keeps track of connections

Connection Pool

A group of connections to a particular resource manager

Resource Adapter

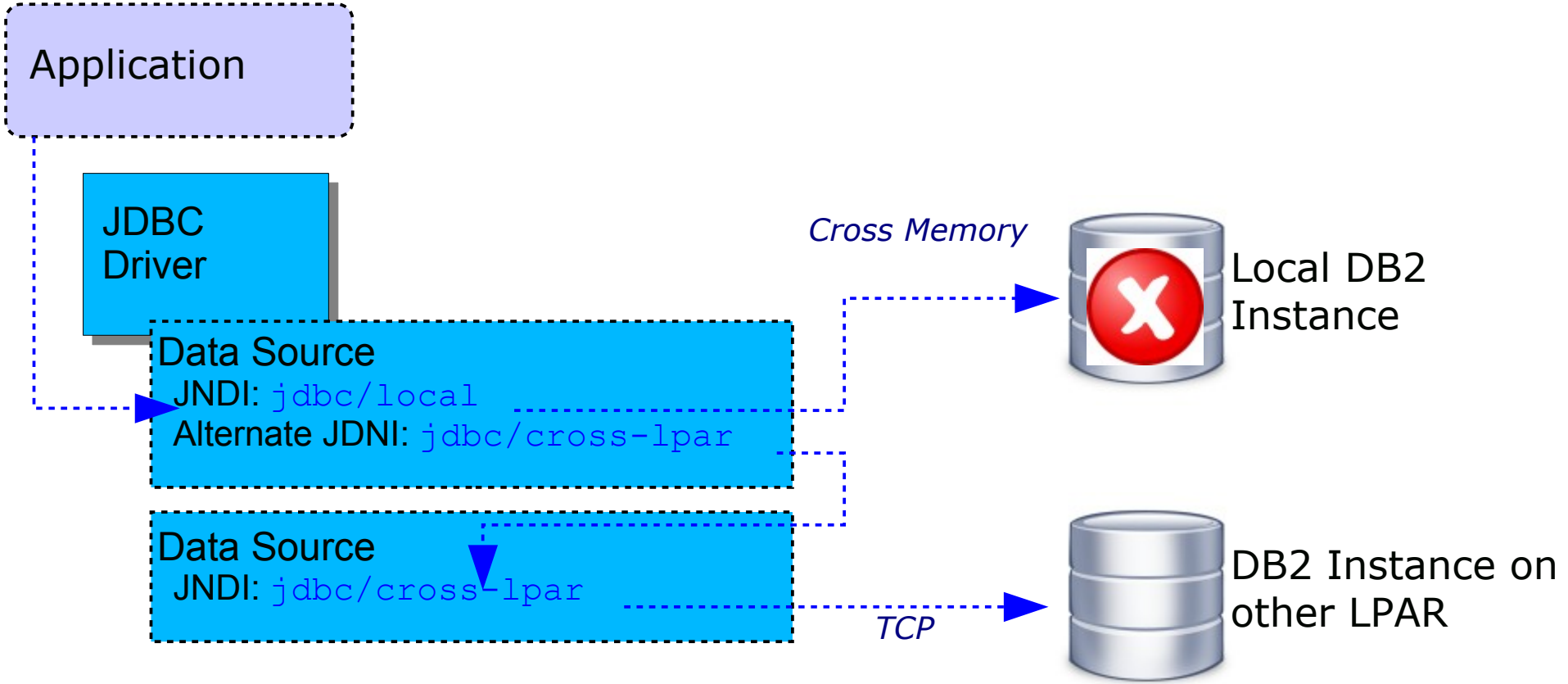
The code supplied by the resource manager that is used to access it. Supplied in a .rar file

Connection Factory

Defined for a particular resource adapter. Applications look up a connection factory to get connections to a particular resource manager. Connection information can either be configured with the factory or provided by the application.

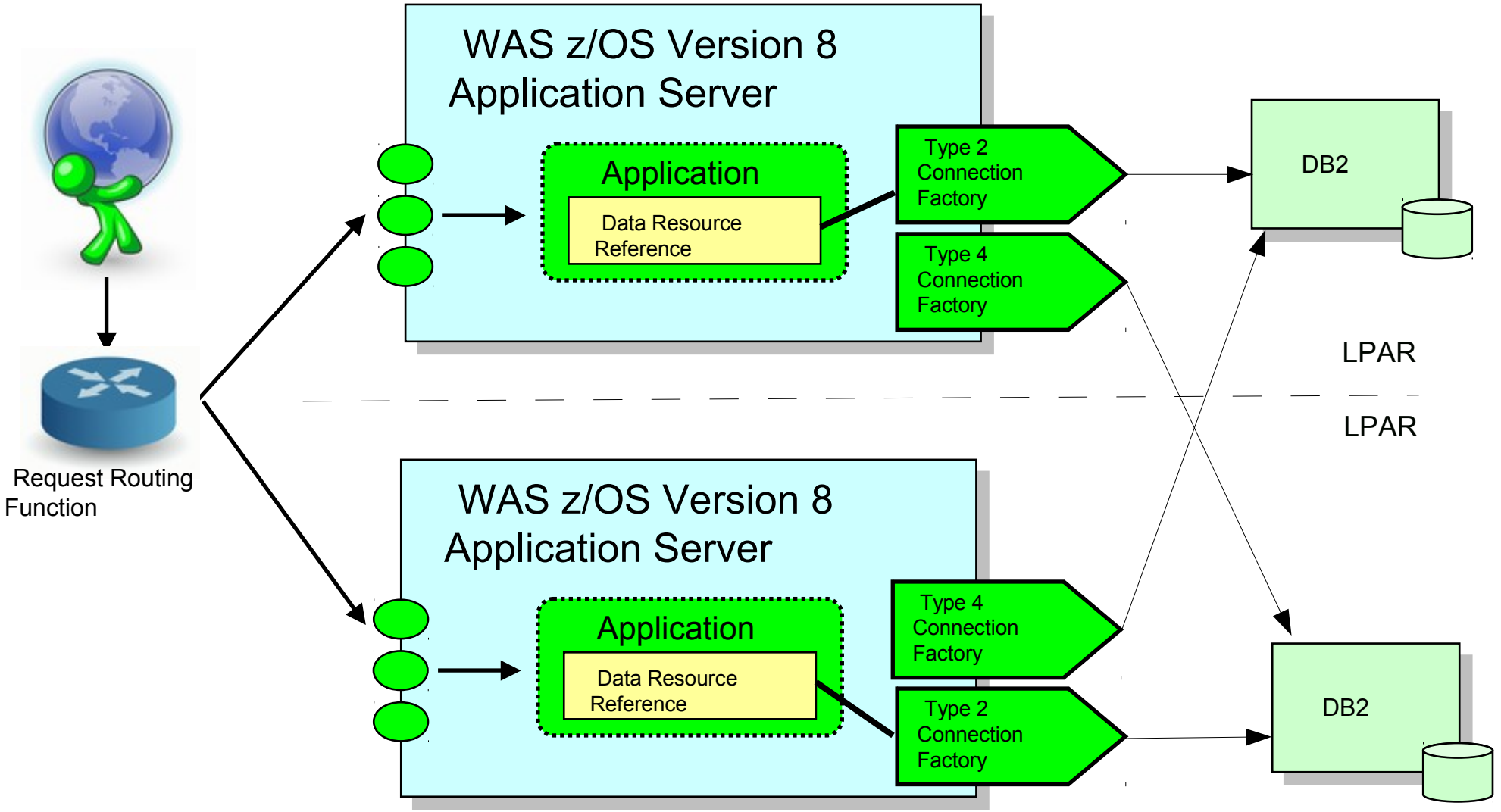
Availability: Resource Routing for Data Sources

Provides a mechanism to fail over to a defined "alternate JNDI" for JDBC data sources so data availability is increased:



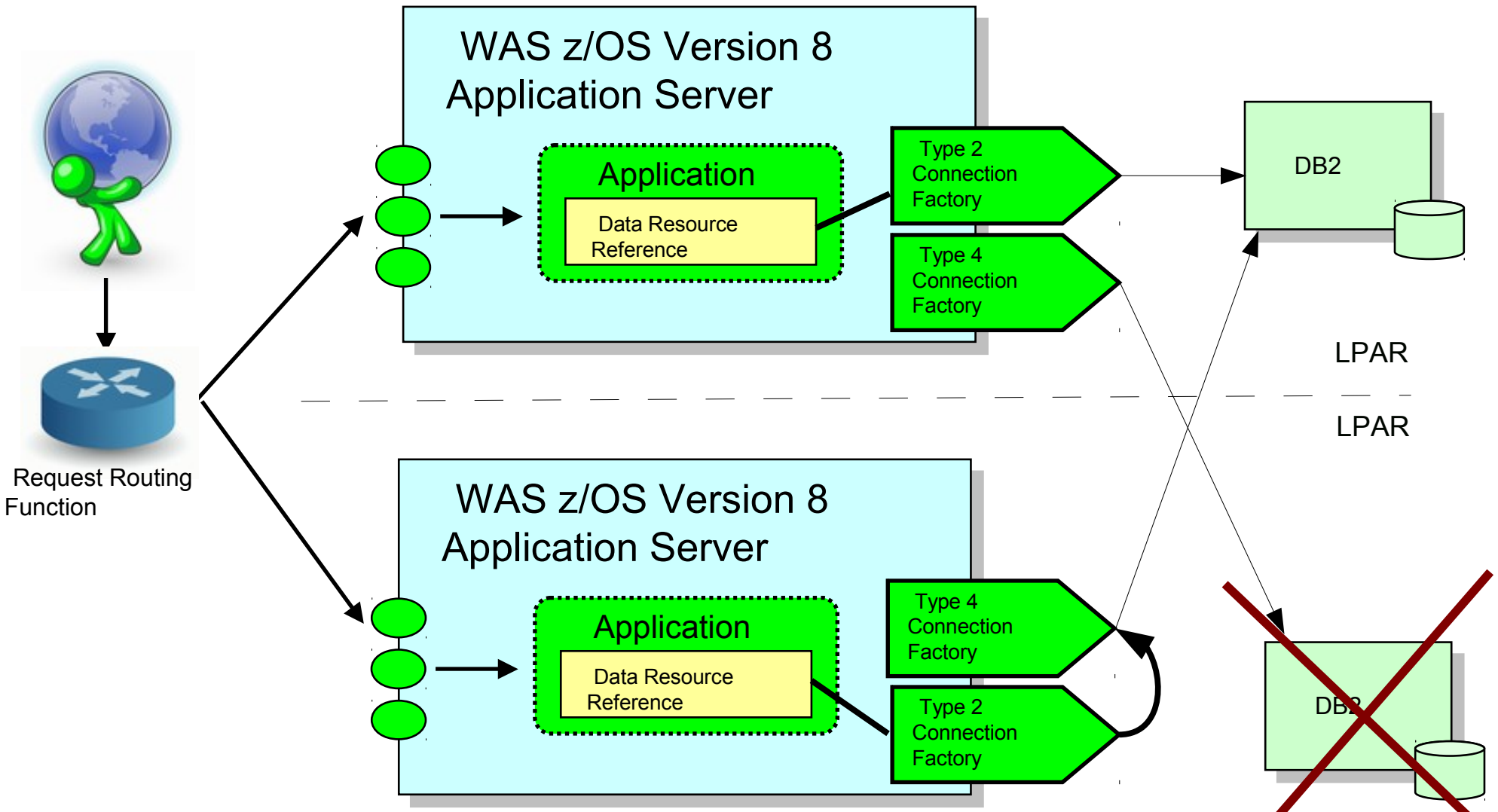
When primary data resource seen as down, new connections then routed to alternate JNDI, which may be defined as remote. When local data resource comes back, new connections resume local.

Suppose we configure both connectors...



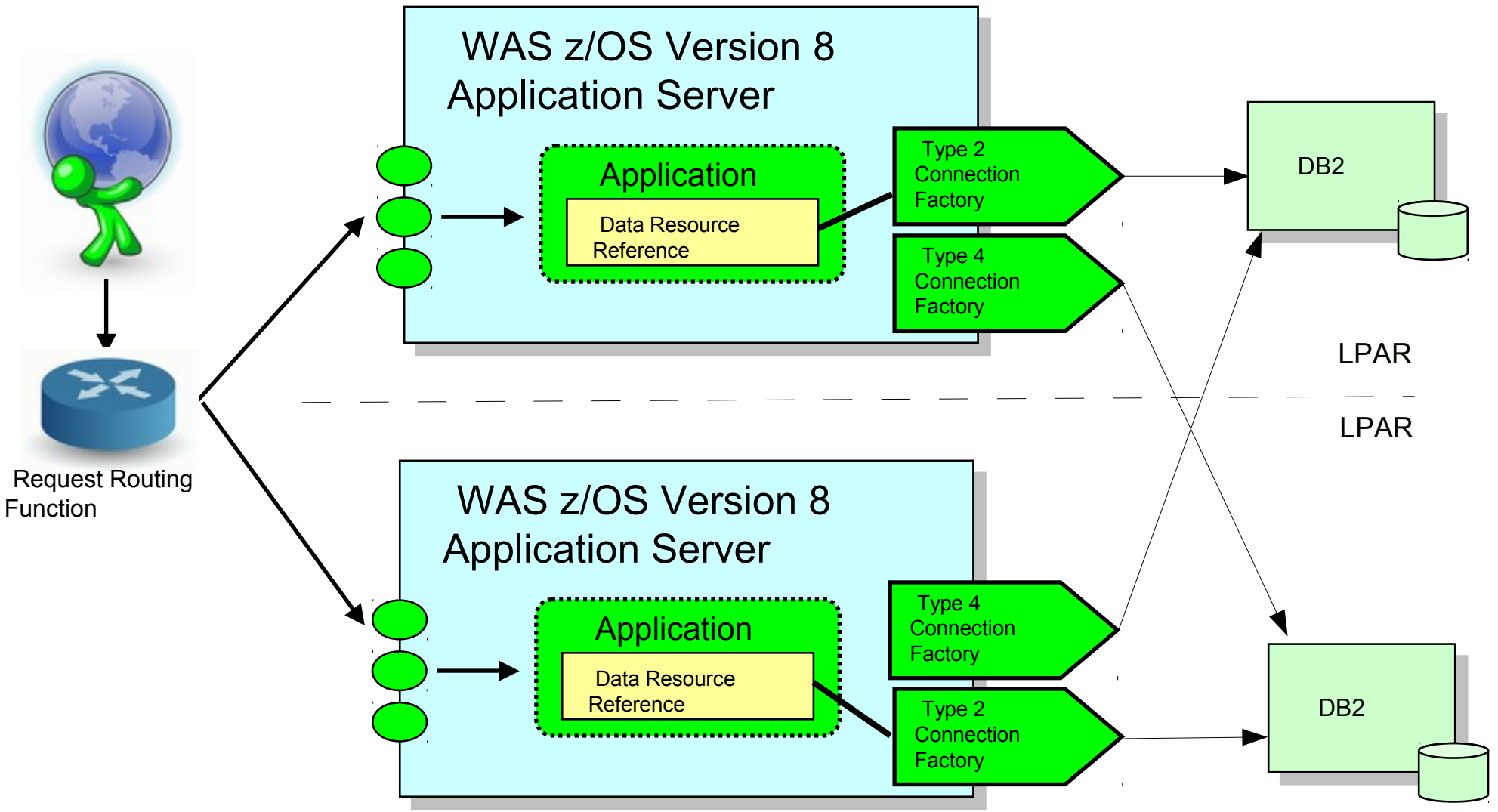
And the Type-4 connector is the 'alternate' for the Type-2

Then something bad happens....



New application connection requests use the Type-4 – AUTOMATICALLY!
WAS will also 'watch' for DB2 to come back....

When the bottom DB2 is back...



Use of the Type-4 quiesces and we're back to normal

#SHAREorg

SHARE
Technology · Connections · Results

WebSphere Application Server BETA Liberty Profile

Preface

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

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

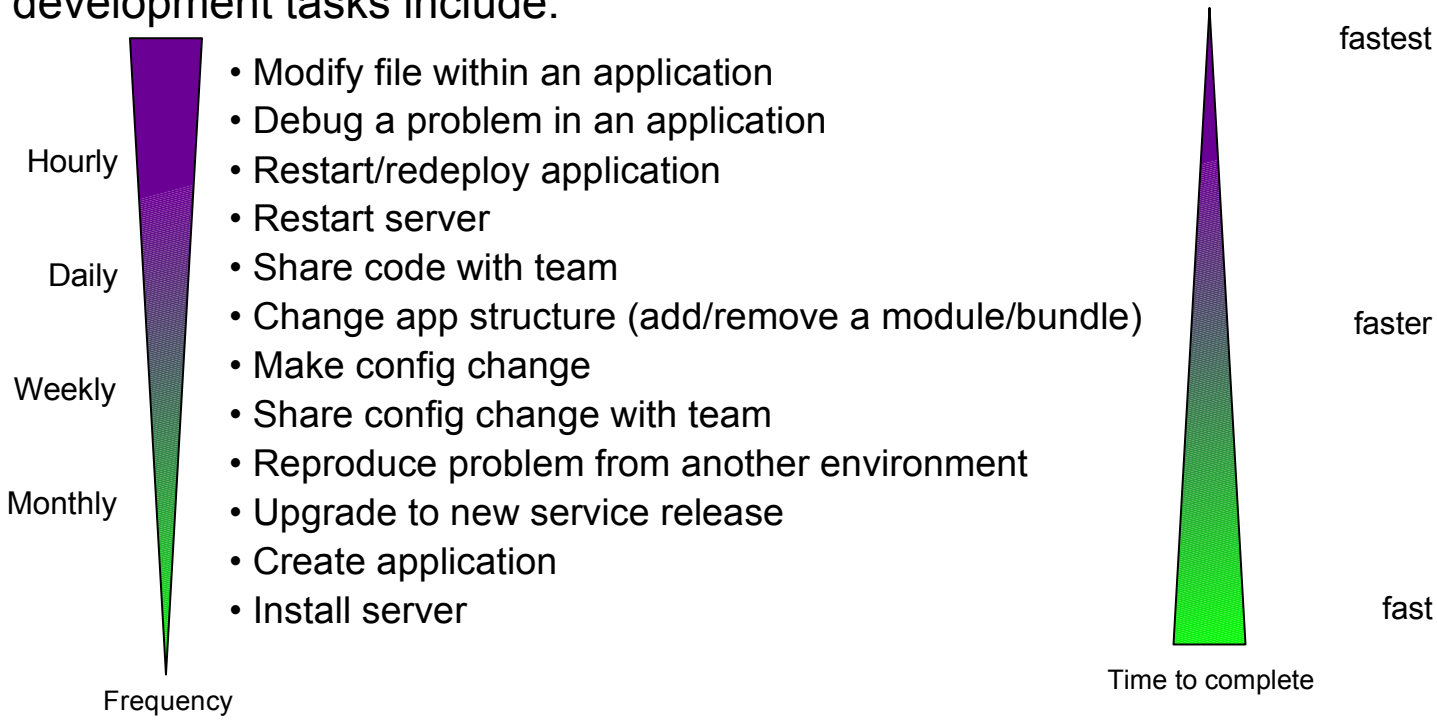
- CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
- ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

Developer Feedback

- Developers are looking for more (or less) from the test server runtime in the tools.
  - Greater test simplicity
    - Config model (hard to edit, backup, share)
    - Admin console is for operational mgmt rather than development
    - Problem determination (including within customer apps)
  - Responsiveness
    - Primarily incremental publishing, app install, server startup time
  - Footprint
    - WAS server types defined to RAD have a WAS production server footprint and memory use.
      - For all types of application

Frequency of Development Tasks

- Common development tasks include:



- All tasks should be as painless as possible, with special emphasis on the more frequent ones. If the time taken to accomplish these tasks is an impediment to the development, the cost of the fidelity of the test server runtime is challenged.

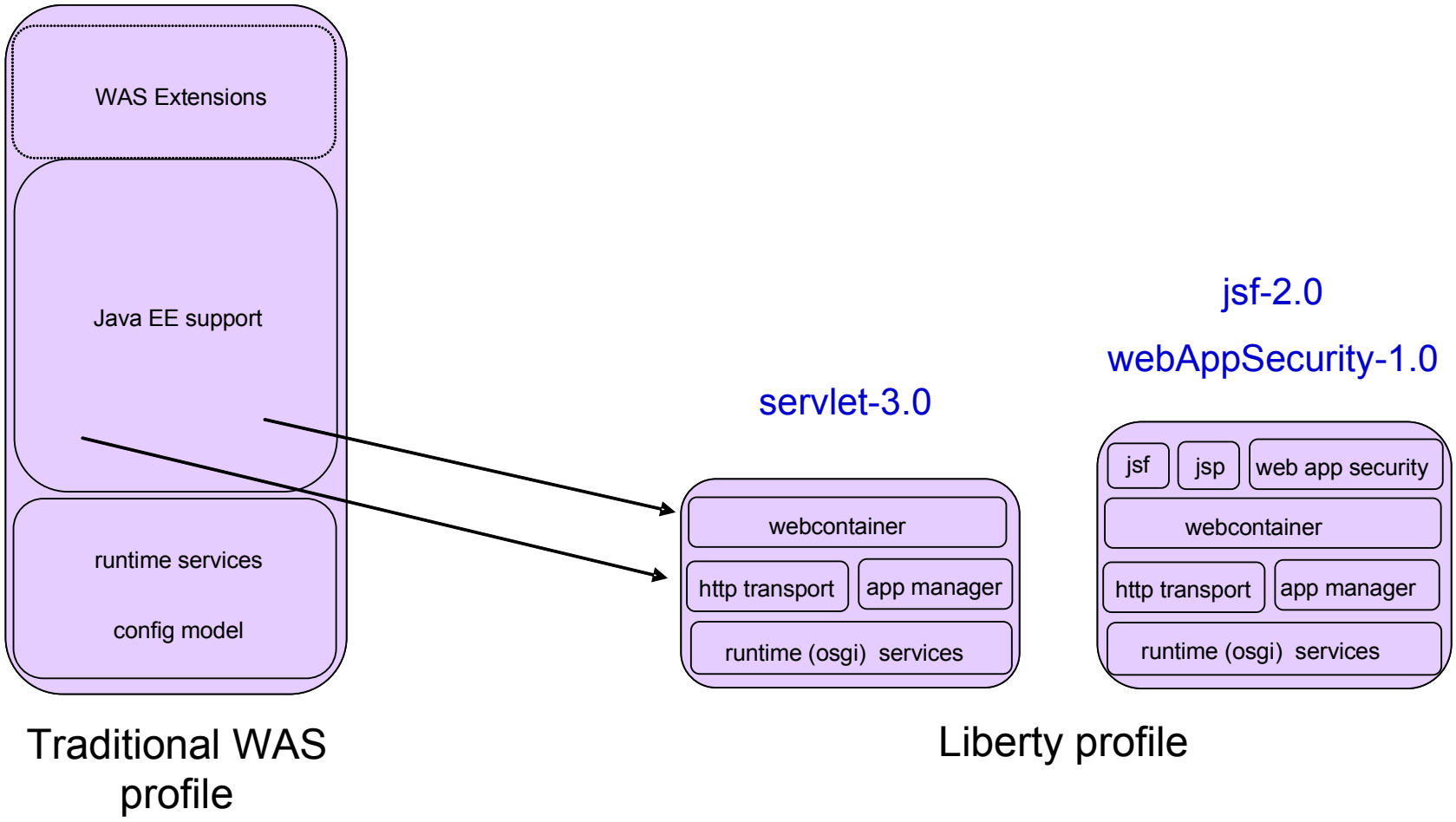
Administrator Feedback

- Administrators want more options for simpler deployments
  - appropriate resource use
    - many applications need only a subset of app server functions
    - increased need to host multiple server runtimes in shared systems (eg clouds)
    - simple ways to share product binaries, configuration and application binaries across servers
  - highly scalable central management
    - thousands → tens of thousands of servers
  - easier and faster version → version migration

WAS V8.5 Beta: Introducing WAS "Liberty" Profile

- Not a single static profile: rather a dynamic, flexible profile of the runtime to load only what the application needs
  - Memory footprint (web feature): < 50 MB
  - Profile is dynamic - switch parts of the server on & off w/out restart
- Extremely lightweight
  - Incredibly fast (re)start times: <5 seconds
- Simplified configuration for quick time to productivity; one single config file or modular config (as desired)
  - Easy to share / diff / manage in version control
  - Easy to componentize config across larger development teams
- Easy access
  - smaller download, unzip install, eclipse plugin, Mac OS support, jdk flexibility
- Tools available as Eclipse features as well as in RAD…

Highly composable runtime based on ‘features’



Simplified Configuration

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

   <logging trace.specification=”webcontainer=all=enabled:*=info=enabled” />
   <application type="war" id="tradelite“
        location="${shared.app.dir}/webcontainer/tradelite.war" />

  <datasource jndiName="jdbc/TradeDataSource" jdbcDriverRef="DerbyEmbedded"/>
  <jdbcDriver id="DerbyEmbedded" libraryRef="DerbyLib"/>
</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

‘references' can be embedded in multiple elements

Flexible Configuration

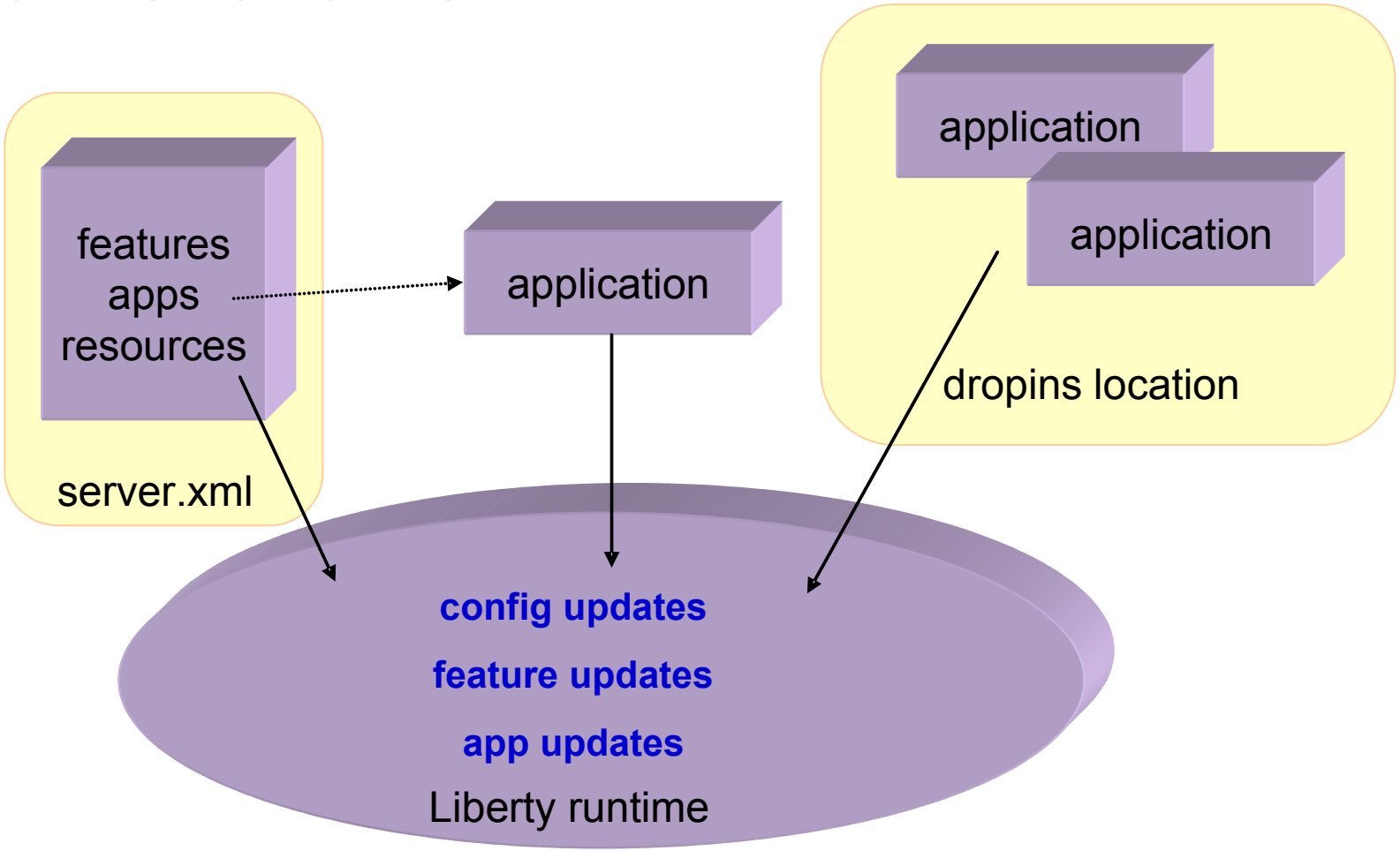
- Shareable config snippets

```
<server>
    ...
    <include url="http://hurgsa.ibm.com/~irobins/
                  delight/config/blog.sample.xml" />
    <include file="${shared.config.dir}/blabber.sample.xml" />
    <include file="same.dir.xml" />
    ...
<server>
```

server.xml

- Config can be componentized at any level of granularity, from 1 file to many.
- Visualization through Eclipse Liberty server adapter as a single logical view.

Dynamic runtime



What This Means For Developers

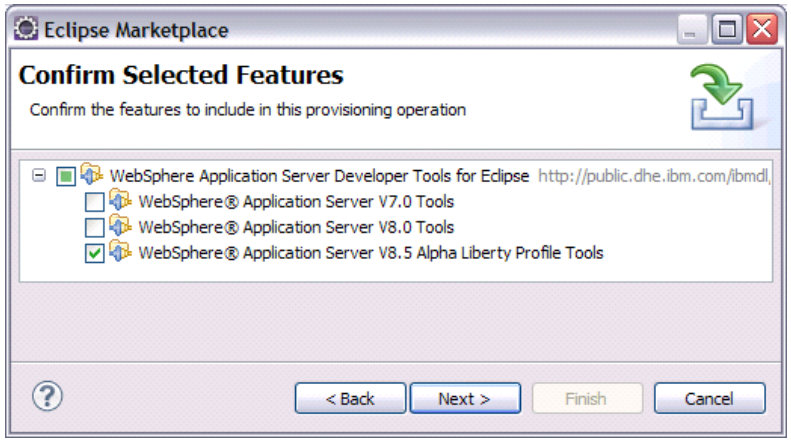
- A new WAS server type for Eclipse which is focused on the development experience
  - Fast to download, unzip and set up
  - Easy to share, version and compare configs
  - No more waiting for application updates or server restarts
- Good fidelity retained through evolution of the WAS server type
  - Simple promotion of application through development, test and deployment systems

What This Means for Production

- An “Embedded Server” profile is a production instance of the configured Liberty server type.
  - Think of zipping up the application, configuration and server type you just tested on.
  - Application centric – the server is pre-configured for a specific application(s).
- Deployment options:
  - Unmanaged unzip install
  - Managed ND deployment of standalone instances
    - “Light-touch” ND management: start and stop server
  - Server configuration remains via the same simplified XML config created in the development environment.

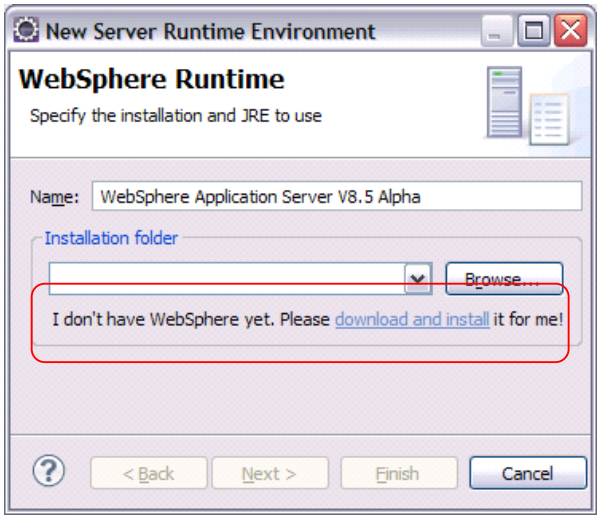
Get the WebSphere Dev Tools Easily

- Free & frictionless download of **developer tools**
- Extends Eclipse JEE Helios (3.6) and Indigo (3.7)
- No time limit. Available from:
  - New WASdev community site: http://wasdev.net
  - Within Eclipse (Help→MarketPlace)
  - Eclipse Marketplace web site
    - http://marketplace.eclipse.org/
  - Eclipse update repository



Get the WAS Runtime Easily

- Free & frictionless download of the **WAS runtime** for development & test
- No time limit.
- Full profile “WAS for Developers” has been available since WAS V7.
- New lightweight “WAS Liberty profile” is now available through the WAS V8.5 Open Alpha program.
- A zip download for web application development: **25 MB**. Available from:
  - New WASdev community site: http://wasdev.net
  - Within the WebSphere Development Eclipse
- Develop, debug and test web applications using lightweight tools and runtime – deploy with complete fidelity to full-profile WAS server.



What is the WAS for z/OS Liberty profile?

- The WAS for z/OS Liberty profile is the Liberty profile with *optional* extensions for z/OS
  - z/OS specific functions are modeled as independently enabled *feature sets*
  - Simple configurations that require no active z/OS exploitation can run in a completely unauthorized environment
  - Advanced configurations allow for active exploitation of z/OS services and tight platform integration
  - Enables fine grained access control to authorized services
- Liberty is a single process, 64-bit only application server
  - Traditional split-process runtime remains a part of the product
  - Administrators choose which profile to deploy
  - Liberty offers a subset of the traditional programming model
- An *Angel* process enables the use of authorized code in the server process

Why Liberty on z/OS?

- Simplified configuration
  - Single XML document for server configuration; no need for a profile management tool
  - Optional exploitation reduces required z/OS system configuration dramatically
    - Only required when exploitation of features is enabled
- Significantly improved performance
  - Server startup time for simple applications < 5 seconds
  - Projected throughput for simple web-apps like Trade +20%
- Significantly reduced real storage and disk requirements
  - ~20-25MB disk footprint
  - ~50-75MB real storage per server instance
- Stack portability
  - Provides an environment that is nearly identical to distributed platforms while still enabling z/OS differentiation

Liberty for z/OS – Use cases

- Test
  - Simple environment for testing applications on z/OS
  - Shared binaries; independent configuration
  - Fine grained controls allow administrators to control the services development has access to
    - Example: Completely disable use of authorized services
    - Example: Enable RRS transaction management but not identity assertion
- Lightweight production
  - Stateless web applications that can be deployed across multiple servers for availability
  - Integration with the z/OS security product for authentication and authorization
  - Command and control via the MVS operator console
- Embedded application server
  - Enables a “roll-your-own” server that includes the Liberty core, product specific extensions, and the required configuration documents

Liberty for z/OS – Process structure

- *Angel* – the authorized agent
  - Extremely light-weight started task
  - Required to bootstrap servers that use authorized system services
  - Single instance per system image regardless of WAS topology
  - No configuration to manage
  - No code level dependency between angel and server
- Liberty Profile Application Server
  - Managed as a started task with a native launcher
  - Uses Angel owned PCs to setup environment necessary to run authorized system services
  - Infrastructure is based on distributed code
  - All authorized z/OS extensions optional (Pay as you go)
- Traditional WAS
  - Split process, needs daemon infrastructure