

# zWAS-IRL (WebSphere Application Server on z/OS - In Real Life)

Rod Feak  
MIB, Inc.  
rfeak@mib.com

David Follis  
IBM  
follis@us.ibm.com

August 13, 2013  
Session Number # 13641



# Trademarks

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

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

\* 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 Oracle.

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.

MIB is a trademark of MIB Group Inc.

\* 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.

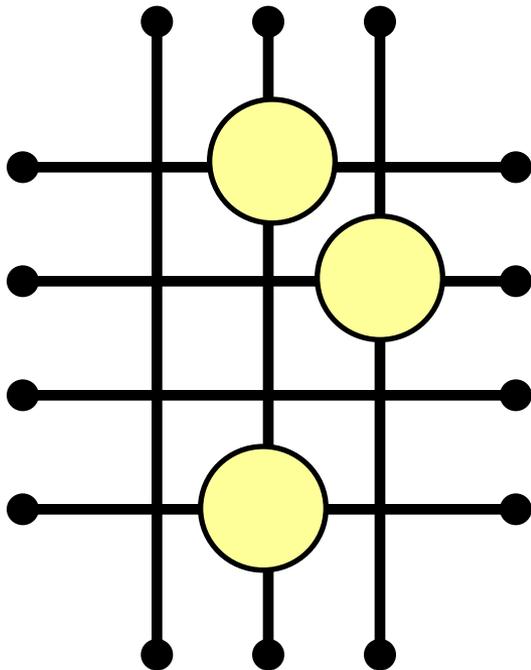
# WebSphere Application Server on z/OS Sessions



Day	Time	Room	#	Title	Speaker
Monday	9:30	203	13597	Getting Started with WebSphere Liberty Profile on z/OS	David Follis
Monday	4:30	203	13600	Managing Server Output from WAS on z/OS	Mike Loos
Tuesday	9:30	203	13644	Using WAS Optimized Local Adapters (WOLA) to migrate your COBOL to zAAP-able Java	Jim Mulvey
Tuesday	11:00	203	13640	Need A Support Assistant? Check Out IBM's! (ISA)	Mike Stephen
Tuesday	1:30	207	13953	What Would Life Be Like If You Ran Your Internet Applications On z/OS?	Ed McCarthy
Tuesday	3:00	203	13641	zWAS: In Real Life	Rod Feak
Wed.	1:30	202	13601	Lab: WebSphere Liberty Profile on z/OS	everybody
Thursday	11:00	203	13598	Getting Started With Compute Grid	John Hutchinson
Thursday	3:00	203	13645	Configuring Security for Liberty	Mike Loos

# Objective of Session

To offer some background on key functions of WAS z/OS  
and then discuss how customers use those functions

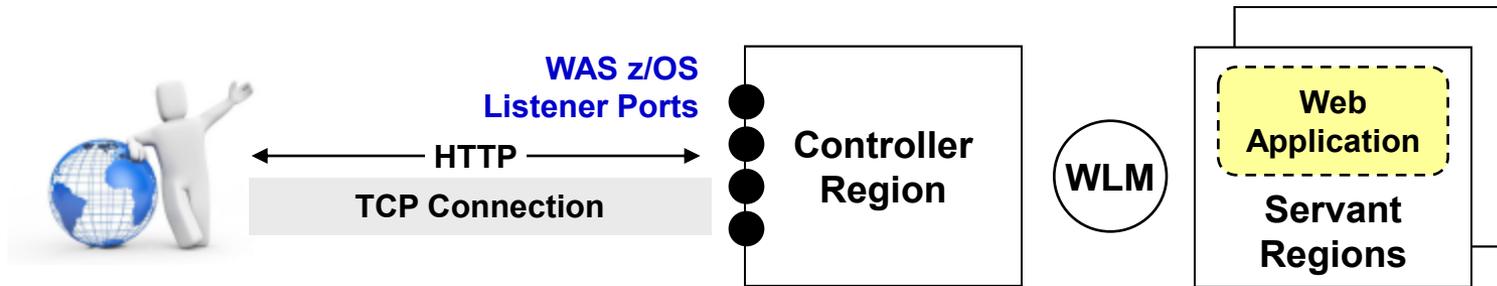


*Request and Application Designs*

*Data Access Approaches*

*Availability Designs*

# Request Access Approaches - HTTP



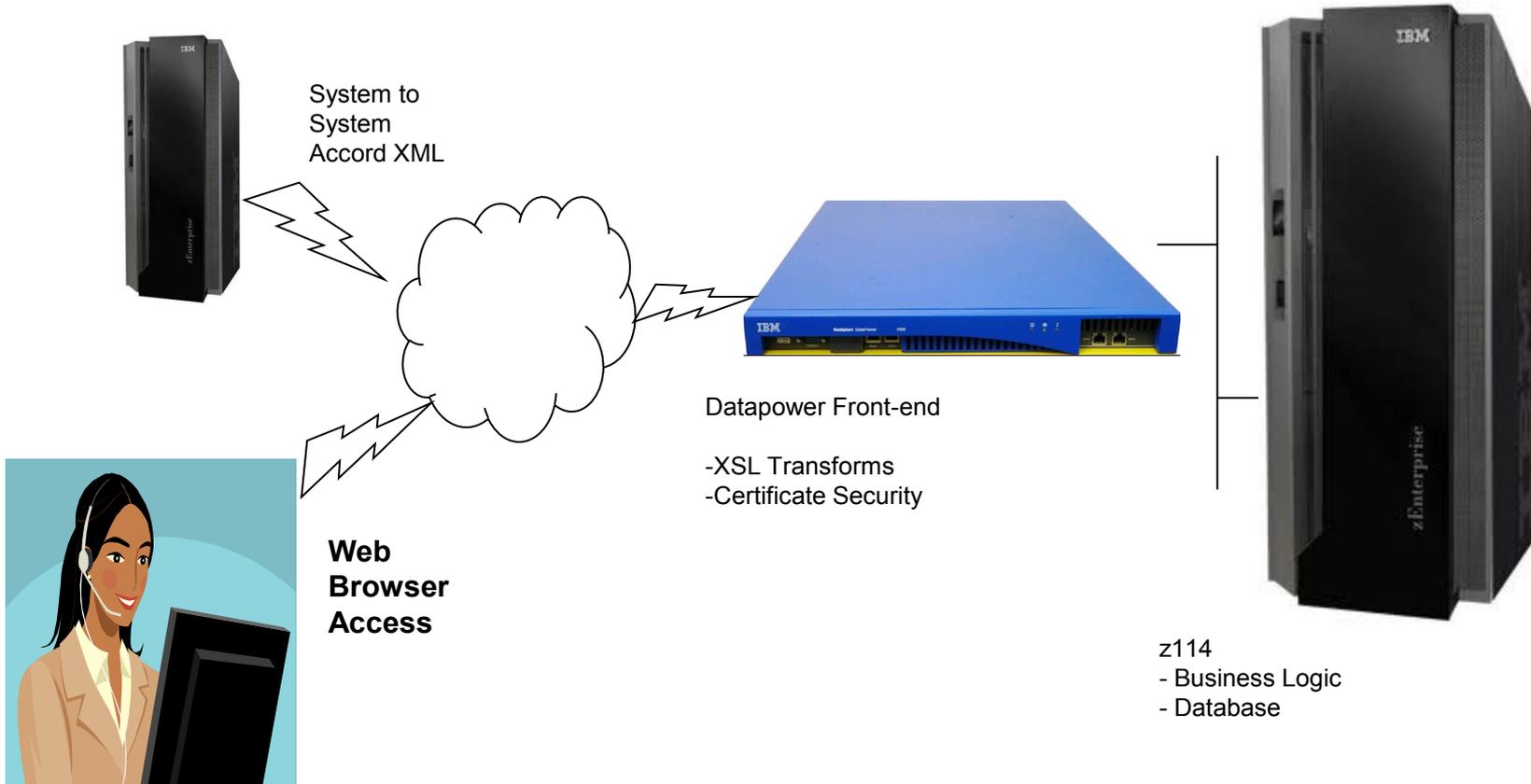
## Overview and Considerations:

- Direct to CR okay for internal, but not for external access
- Typically have some “front-end” device between users and CR  
*Examples: IBM HTTP Server, Apache webserver, OEM load balancing devices*
- Target is most often a JSP or servlet (essentially same thing to WAS)
- Applications often involve more beyond servlet/JSP (ex: EJBs)  
*“Tiered application” design – presentation, business logic, data*
- HTTP often implies HTTP sessions, which implies affinity routing  
*Whether sessions exist is a function of the application*  
*Affinity implies follow-up requests come back to same server*  
*Front-end devices provide routing affinity; WAS has ability to replicate sessions*

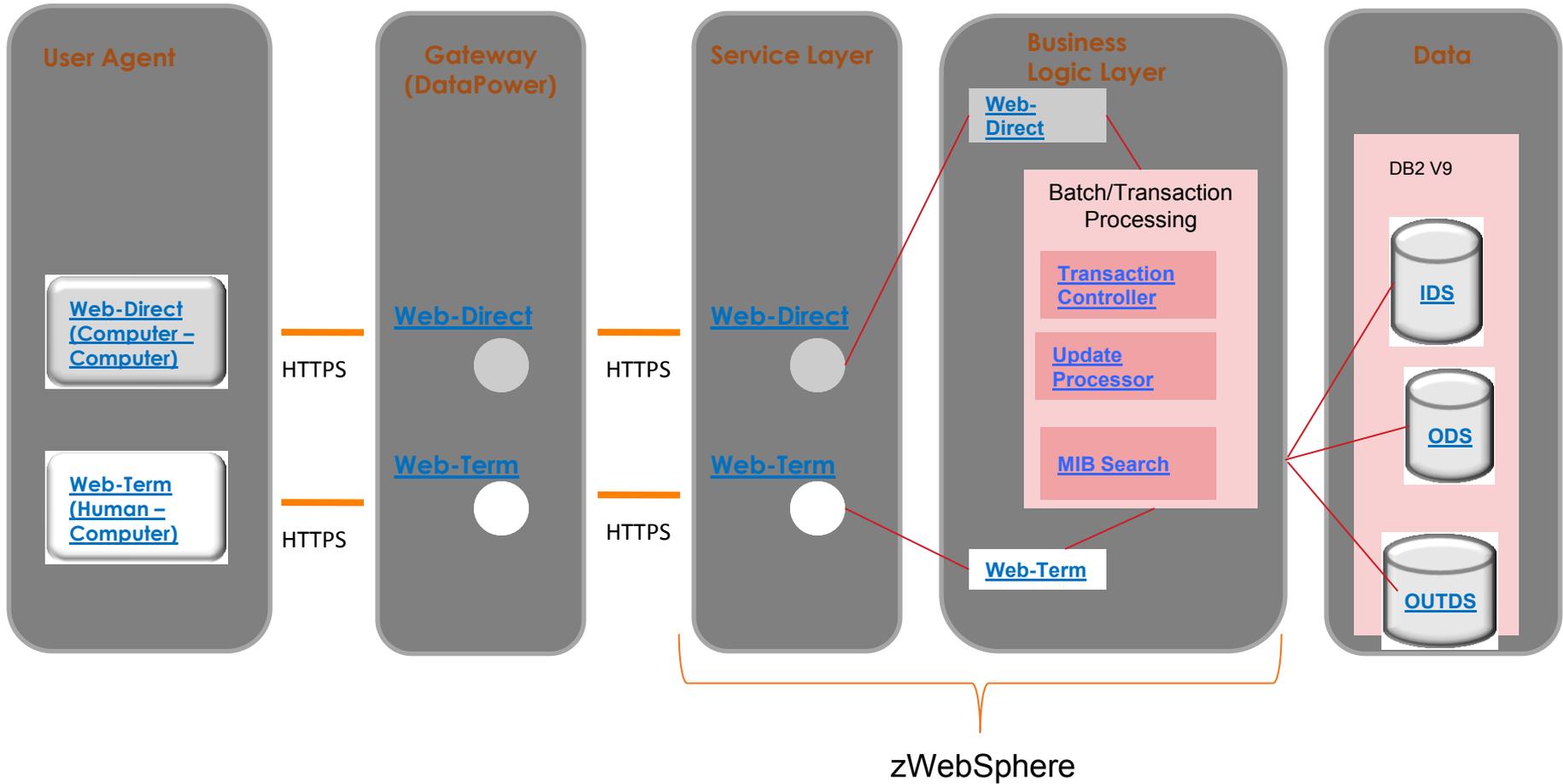
# Who is MIB?

- Formerly “Medical Information Bureau”, now “MIB Group, Inc.”
- A membership corporation owned by approximately 470 member insurance companies in the US and Canada.
- Organized in 1902 to provide core fraud protection services to all aspects of the insurance industry, including, life, health, critical illness, long-term care and disability-income insurance.
- Provide members with shared, searchable databases to support the underwriting process. (Think “insurance credit bureau”)
- Adhere to all privacy and disclosure regulations, including FCRA (15 U.S.C. § 1681 et seq.) HIPAA (Public Law 104-191, 1996) and HITECH Act (H.R.1/P.L.-111-5, 2009)

# Physical Environment



# Processing Overview

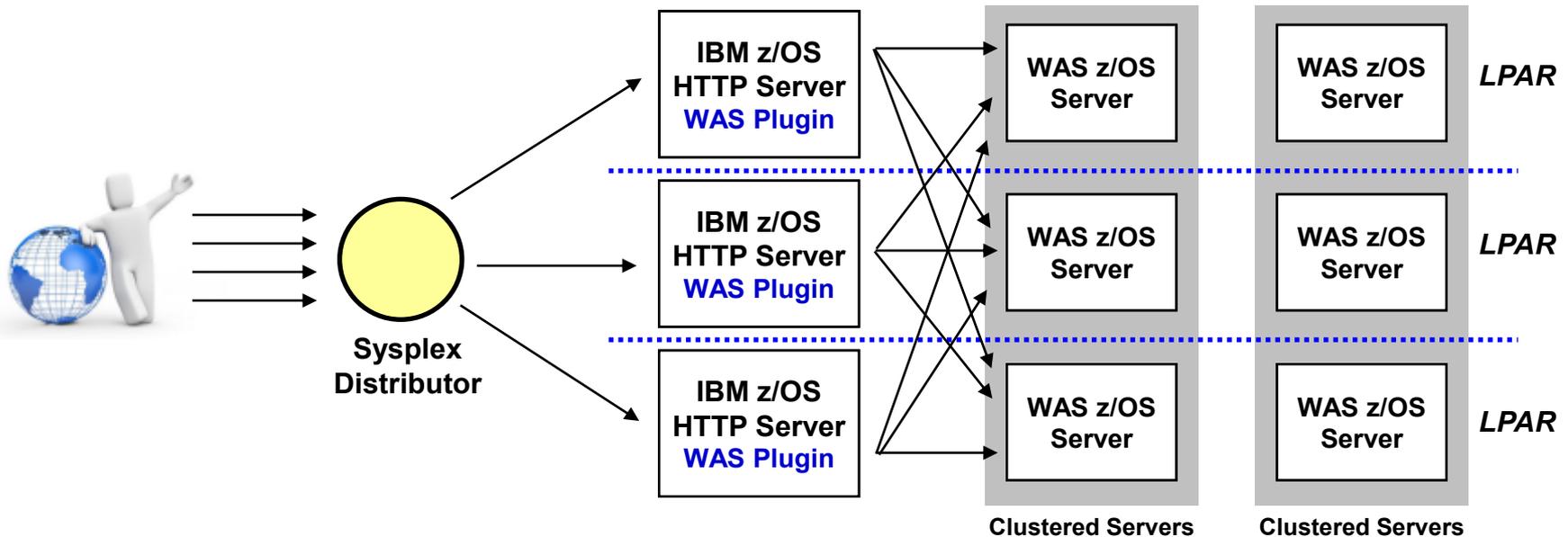


# Customer Usage Examples

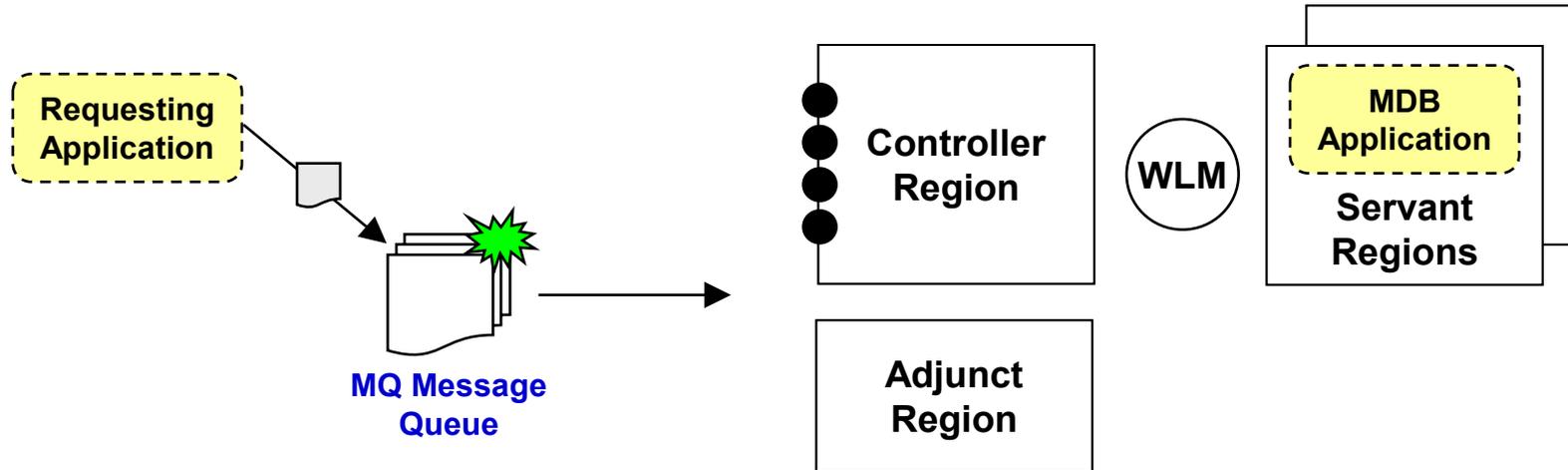
## Federal Government Agency

40+ application servers clustered across 3 z/OS LPARs.

IBM HTTP Server deployed on z/OS LPARs with WAS plugin providing affinity routing. Sysplex Distributor routing to HTTP Servers.



# Message Driven Beans (MDBs)



## Overview and Considerations:

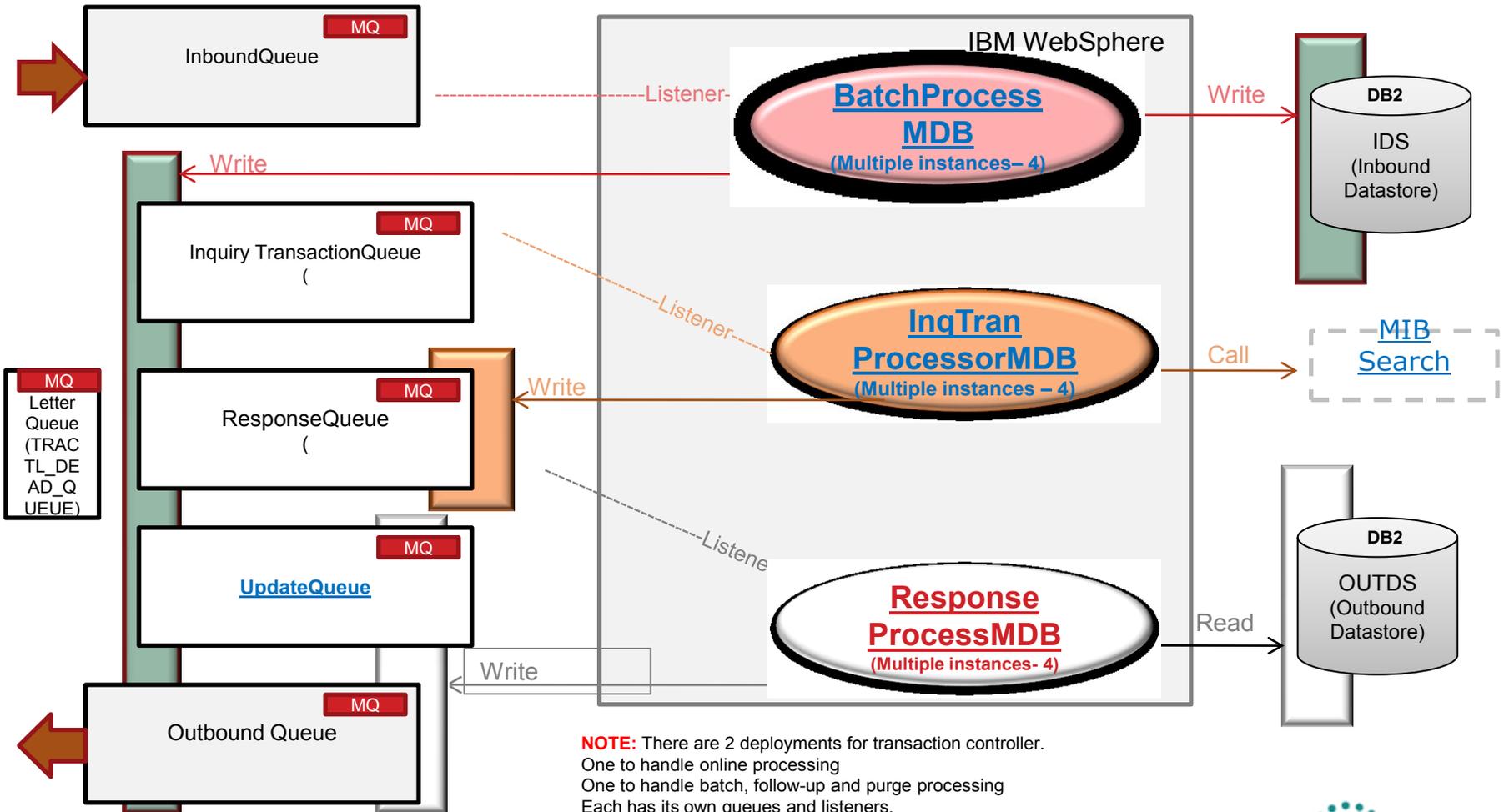
- Provides a very efficient *asynchronous* request interface
- MDB listens for arrival on queue, then reacts and processes
- Which region picks up message depends on how things configured:  
*Message Listener Ports if not Activation Spec*  
*If Activation Spec then Adjunct region*
- MDB applications are a form of Enterprise Java Bean (EJB)

# Processing Example

## Transaction Controller - Design/Deployment

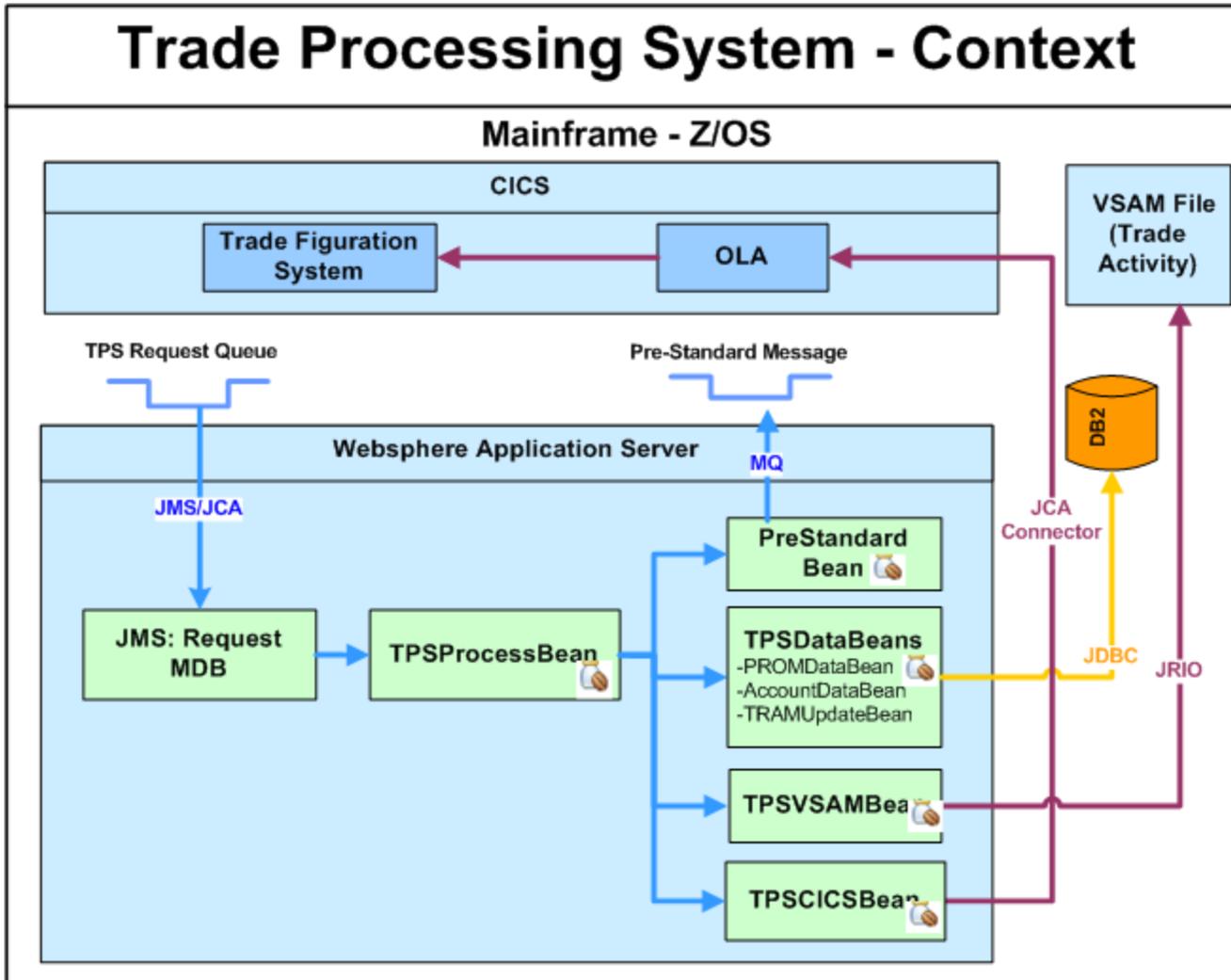
MIBXML

MIBXML

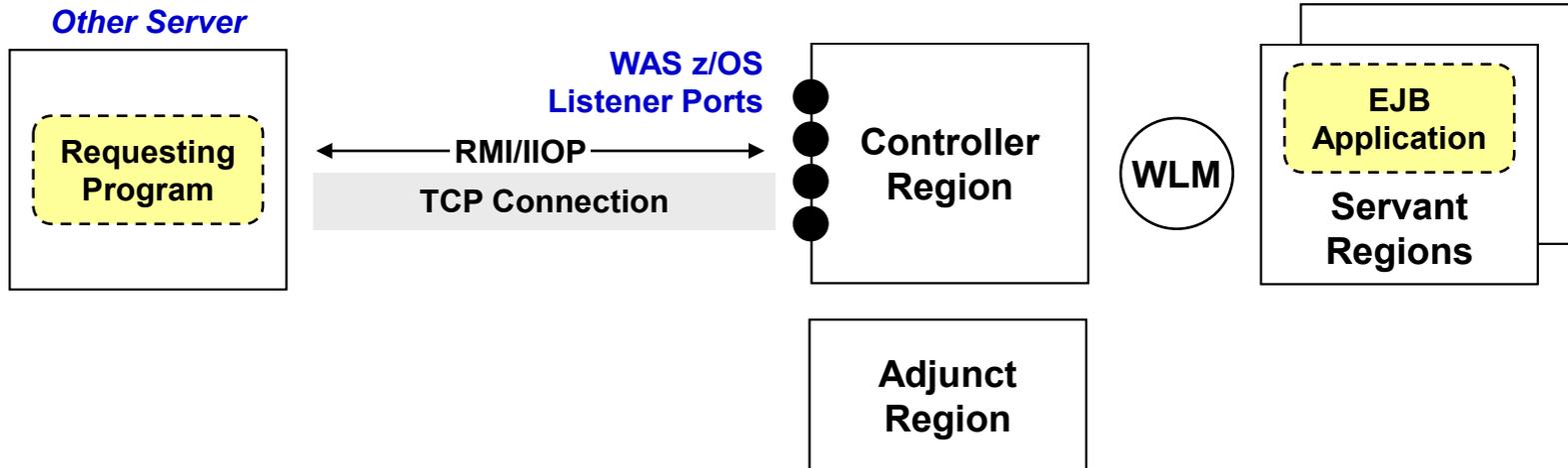


**NOTE:** There are 2 deployments for transaction controller.  
 One to handle online processing  
 One to handle batch, follow-up and purge processing  
 Each has its own queues and listeners.

# A Large U.S. Bank



# IIOP Requests

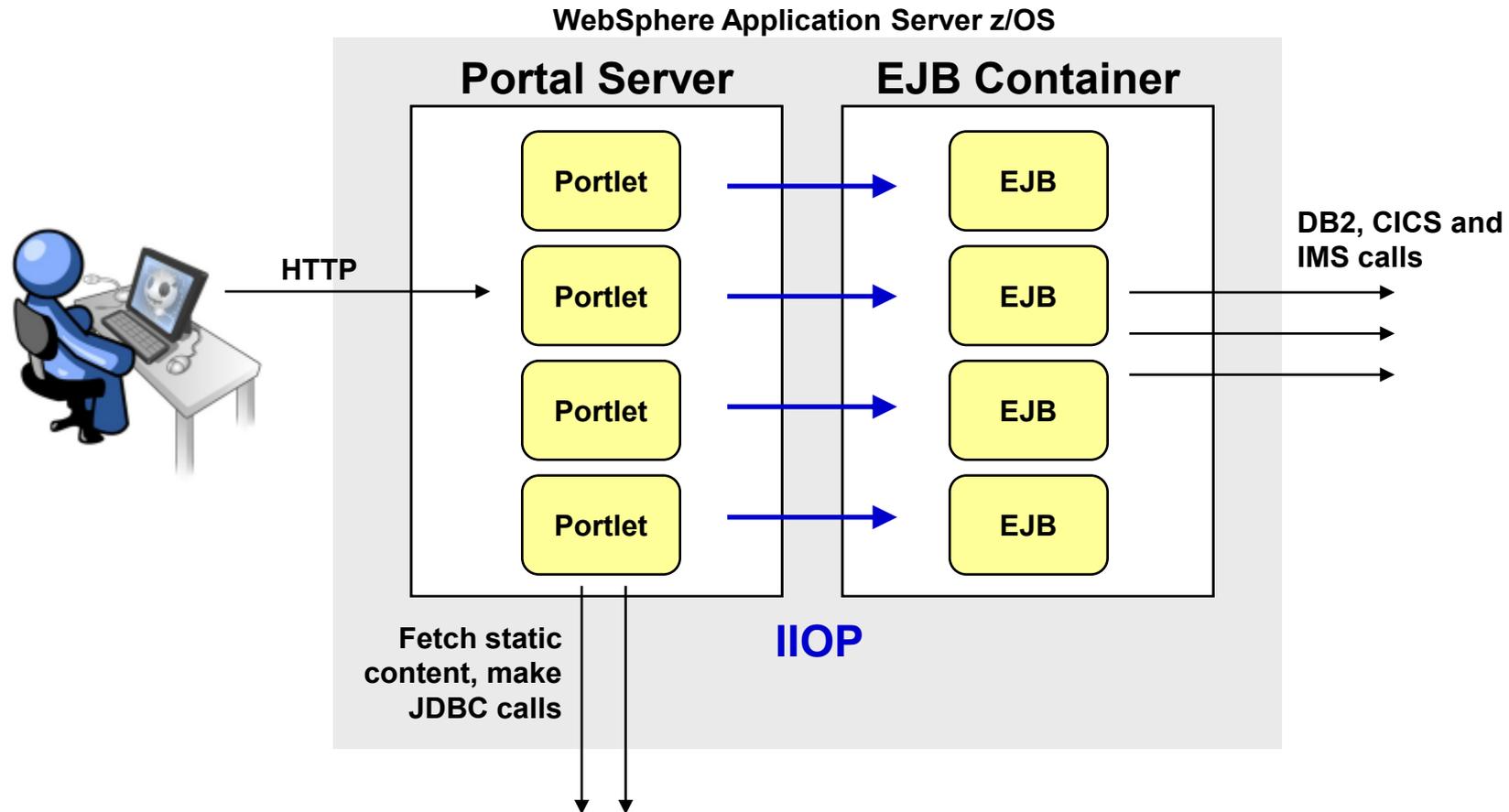


## Overview and Considerations:

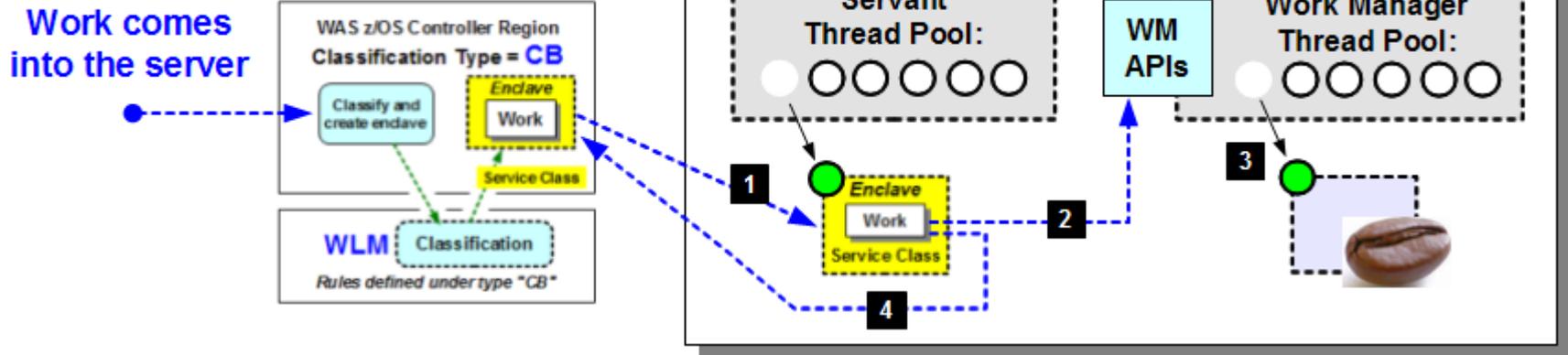
- Program-to-program communications
- Target must be an EJB; requester may be EJB or Servlet
- Requester in same server as target very common  
*Tiered application with servlet calling EJB in same server*
- Requester in other servers, or outside WAS z/OS cell, does exist  
*Where separation of application tiers done for capacity and availability purposes*

# Customer Usage Examples

## Canadian Bank and use of IBM Portal Server:



# Asynchronous Beans

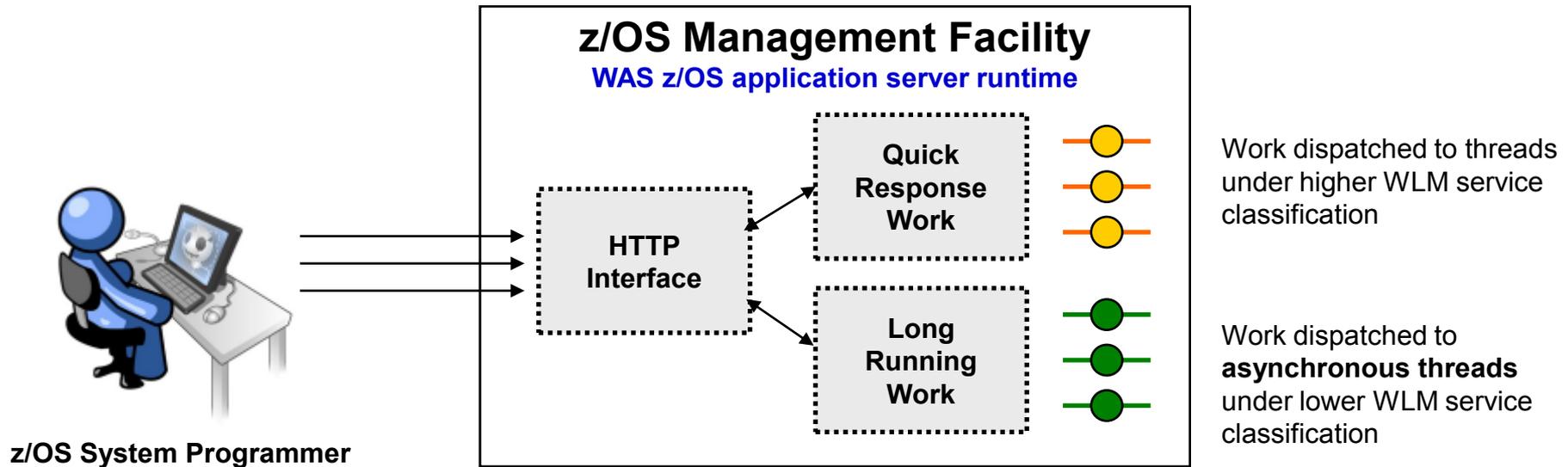


## Overview and Considerations:

- Designed for long-running tasks not subject to normal OLTP timing
- Involves use of “Work Manager” function in application server  
*Requesting application requests thread and invocation from work manager*

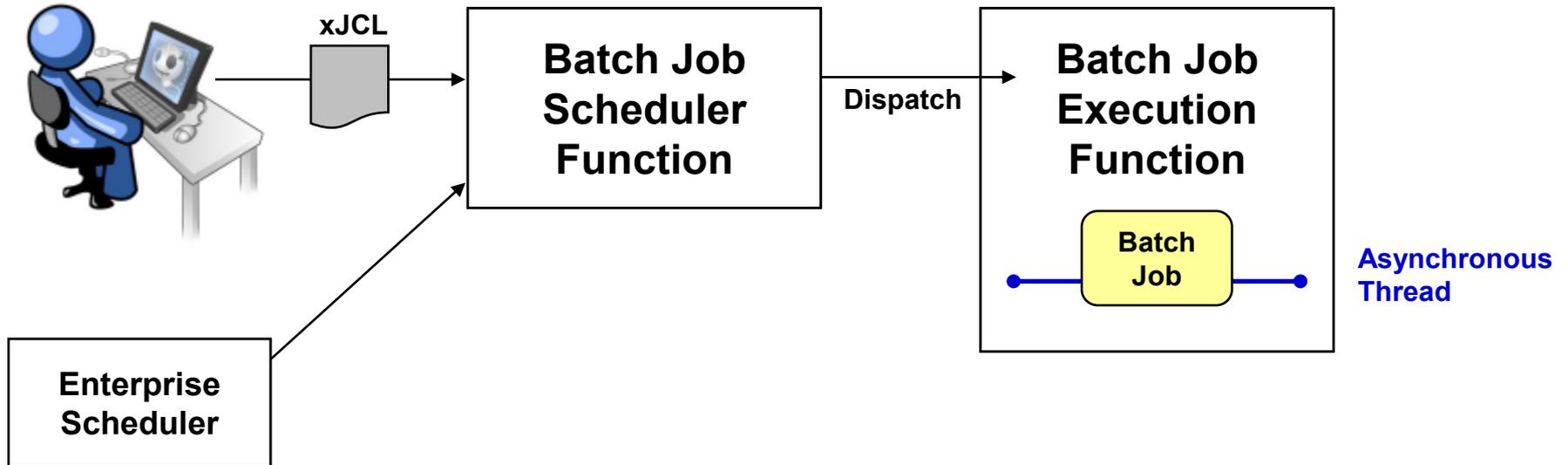
# IBM Use – z/OS MF

IBM's z/OS Management Facility has an HTTP interface but dispatches long running management tasks to asynchronous threads:



# Customer Usage Examples

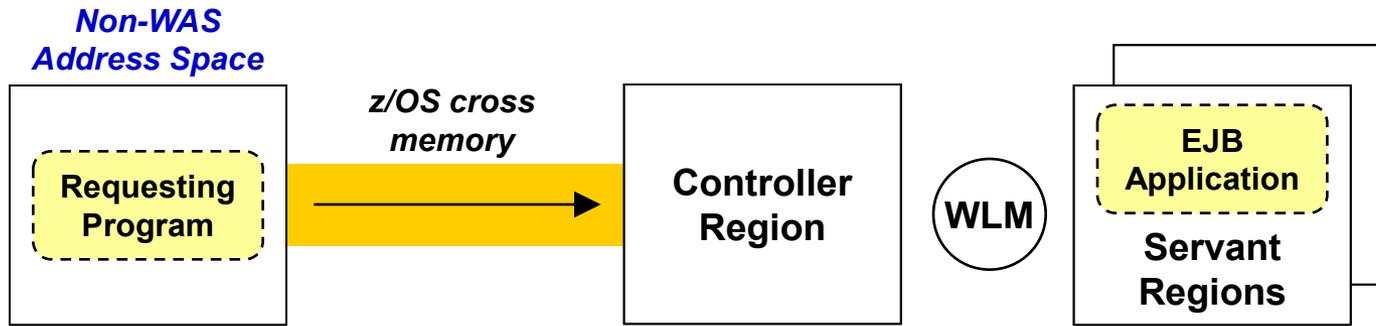
## IBM WebSphere Compute Grid (function now included in WAS V8.5)



By the way, integration with enterprise schedulers is accomplished with a utility that places messages on a queue, and an MDB interface in scheduler processes jobs

**Compute Grid is used by several large customers**

# WebSphere Optimized Local Adapters

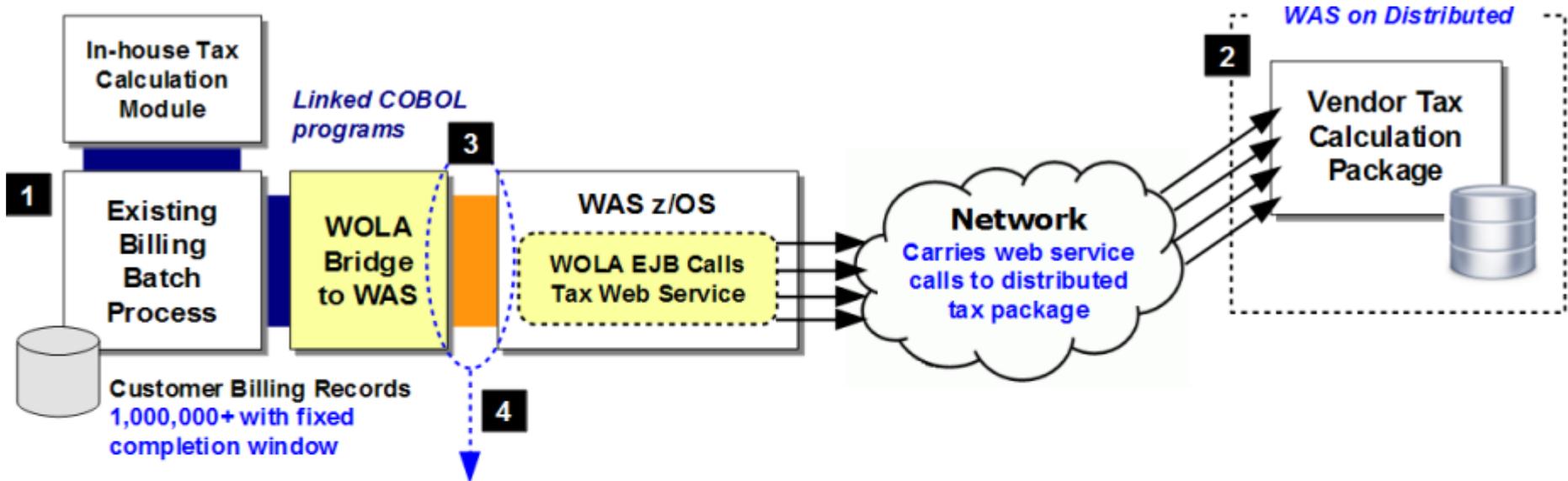


## Overview and Considerations:

- **WAS z/OS exclusive ... exploits z/OS cross-memory services**  
*Very fast with a minimum of latency and overhead*
- **Non-WAS address space: CICS, IMS, Batch Program**  
*Must be on same z/OS LPAR as target WAS z/OS application server*
- **Target application in WAS is a stateless EJB**  
*That implements interfaces using supplied WOLA class libraries*
- **Requester program uses supplied APIs to invoke target and get response**

# Customer Usage Examples

## U.S. Telecommunications Company



### Advanced inbound APIs with asynchronous control

Asynchronous because COBOL is single-threaded and web service call to external tax package is the slowest link. Asynchronous APIs allows COBOL to get program control immediately.

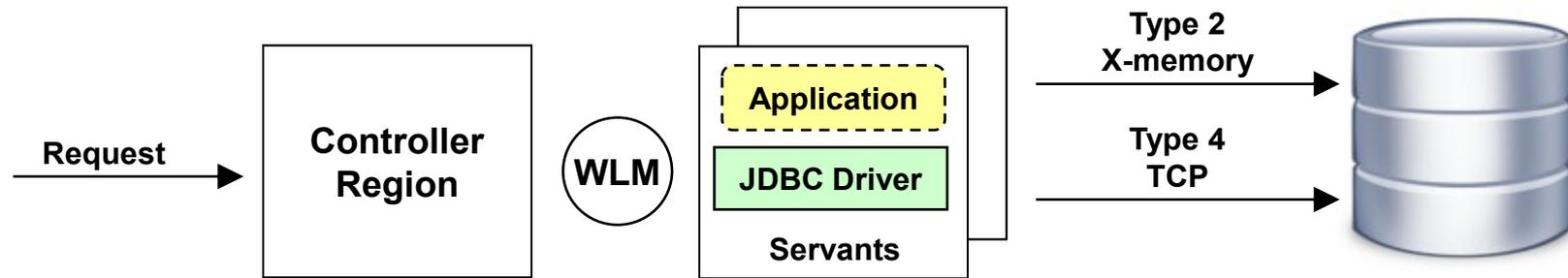
### 150 connections kept loaded with work and busy

Maximum connections over WOLA to EJB. All 150 loaded up with work requests. COBOL then loops through array to see if response received. If so, then process back results and load that connection with another request. Connections kept fully busy in this manner.

### Multi-threaded Java then parallelized web service calls

WAS z/OS and WAS distributed are multi-threaded. Given sufficient processing capacity, the work requests from COBOL may then be handled in a parallel execution fashion.

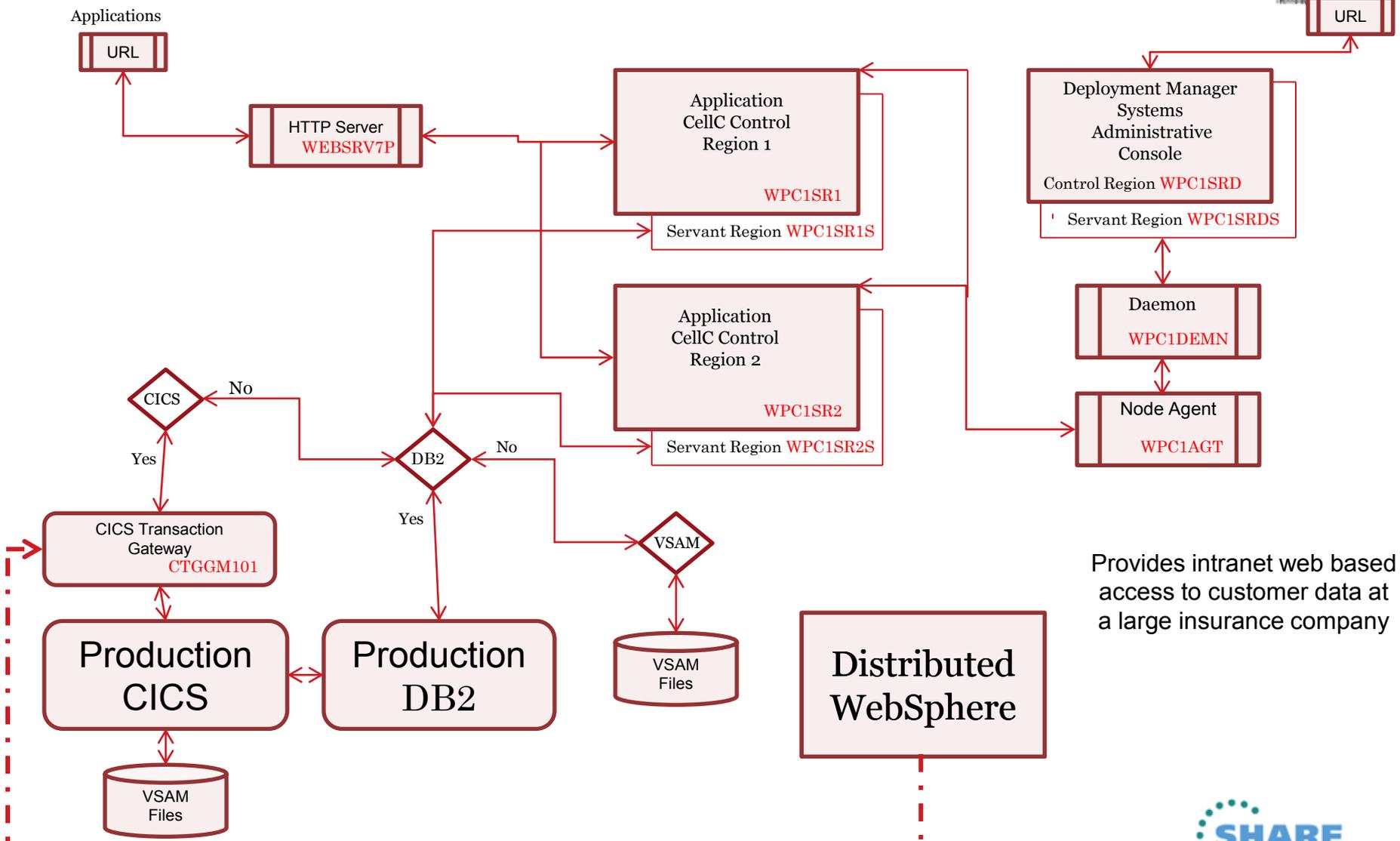
# Data Access -- JDBC



## Overview and Considerations:

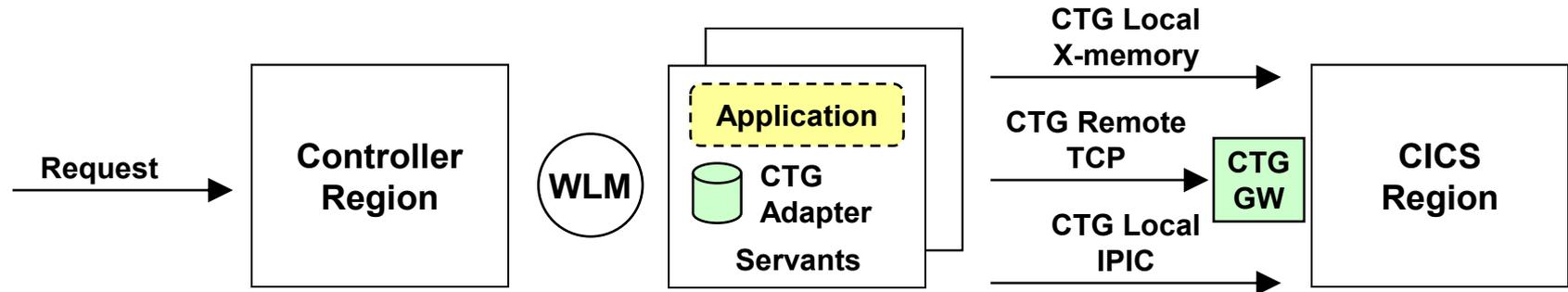
- **JDBC Driver code is provided by vendor of relational database**  
*Provider and data source information configured into WAS by administrator*
- **Transaction: RRS if Type 2; XA if Type 4**
- **T2/T4 zIIP/zAAP offload comparison: WP101476 on Techdocs**  
<http://www-03.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP101476>  
*Type 2 overall CPU slightly lower; GP equivalent (based on specific test run)*
- **Very common data access pattern with WAS z/OS**  
*Nearly every WAS z/OS customer has at least some JDBC in their architecture*

# zWebSphere Processing Flow – System Diagram



Distributed WebSphere Server Connection using TCP/IP Network and Type 4 JDBC Driver

# Data Access – JCA CTG to CICS



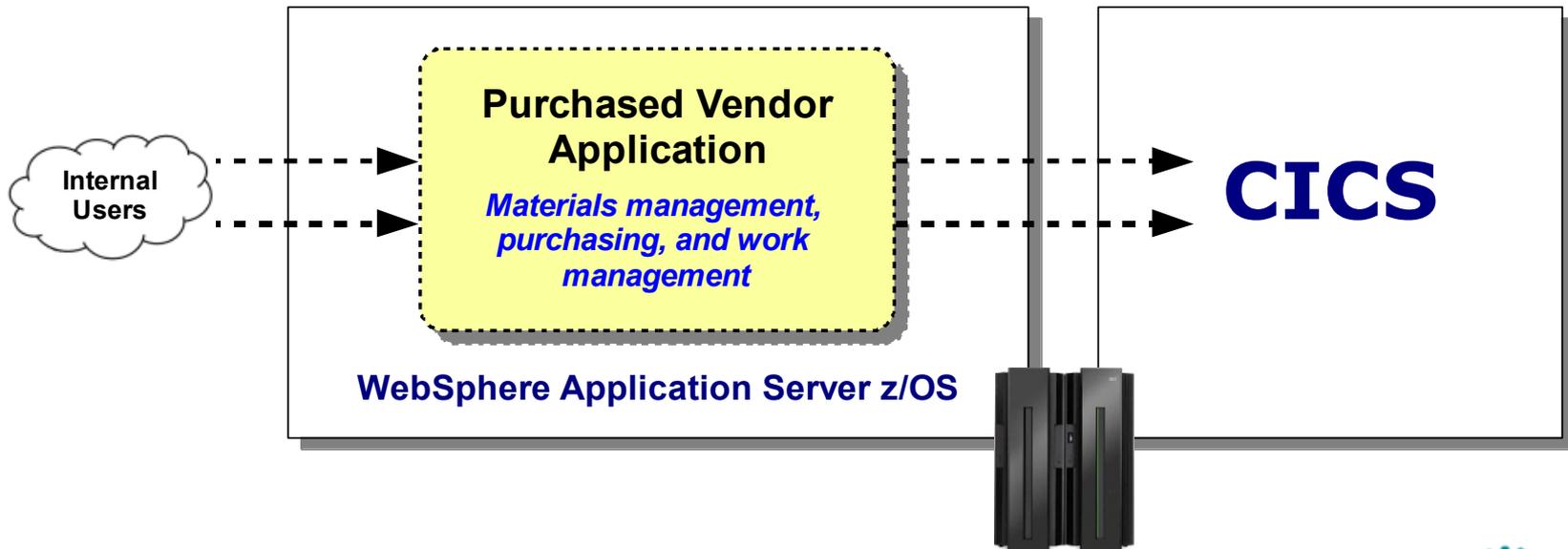
## Overview and Considerations:

- **CTG = “CICS Transaction Gateway”**  
*IBM product that provides the JCA resource adapter and connectivity code*
- **CTG JCA Resource Adapter installed into WAS z/OS**  
*Connection Factory (CF) configuration then provides information about region to connect to*
- **Accessing CICS also very common for WAS z/OS customers**  
*A great deal of business logic has been developed / deployed on CICS over the years  
JCA CTG to CICS provides relatively easy access to those assets from Java front-end  
Also used to access VSAM data managed by CICS applications*

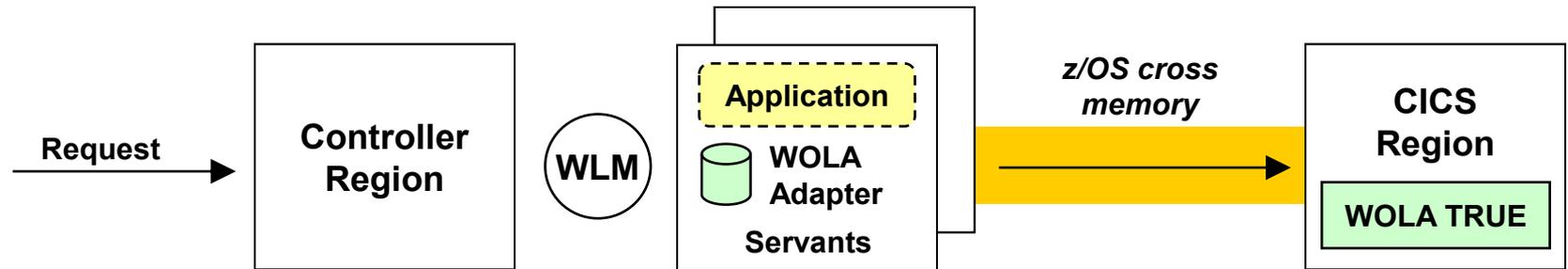
# A Large Power Company



- **1800** unique users per day
- **400** concurrent peak on average day
- **2200** concurrent peak every fortnight



# Data Access – JCA using WOLA

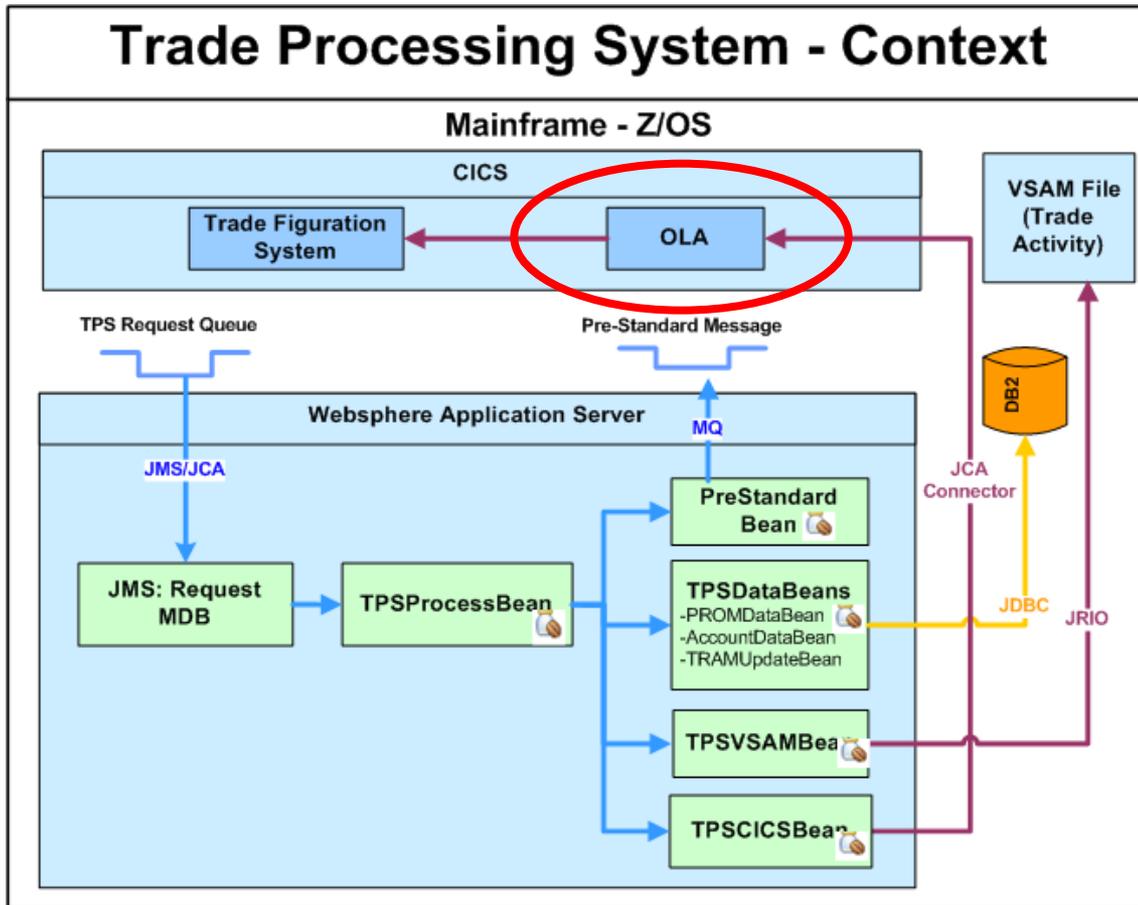


## Overview and Considerations:

- **WOLA Adapter is JCA with Common Client Interface (CCI) implemented**  
*Methods the same as CTG JCA adapter; values supplied to methods based on WOLA, not CTG*
- **Task Related User Exit (TRUE) in CICS implements WOLA code**  
*Provides the low-level cross-memory connectivity function*
- **WOLA Link Server in CICS (not shown) invokes the named CICS program**  
*WOLA link server task handles WOLA call, then turns and does DPL to program  
CICS program unchanged ... unaware of WOLA*
- **Supports 2PC and WAS user thread identity assertion into CICS**

# Customer Usage Examples

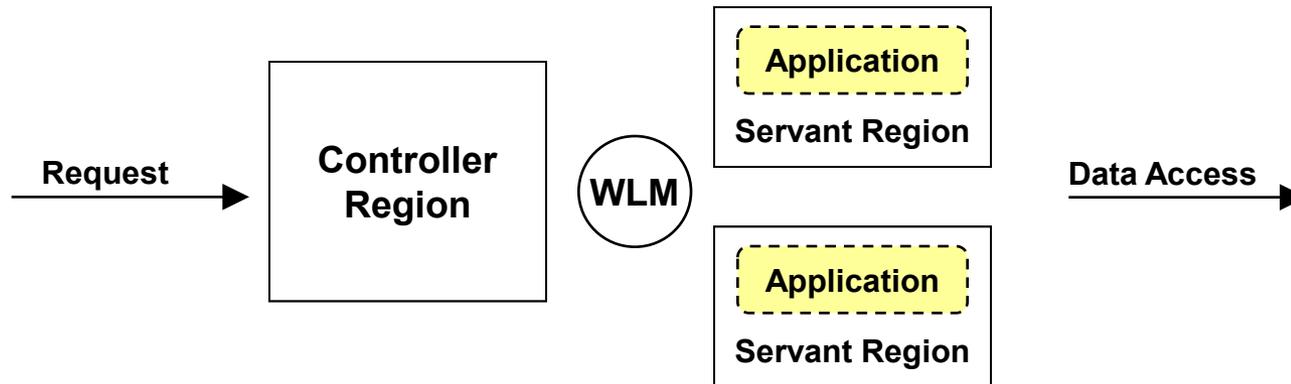
## A Large U.S. Bank



TPSCICSBean uses WOLA JCA adapter to access local CICS region

Another financial company using WOLA to access multiple CICS gateway regions using round-robin function introduced in 8.0.0.1

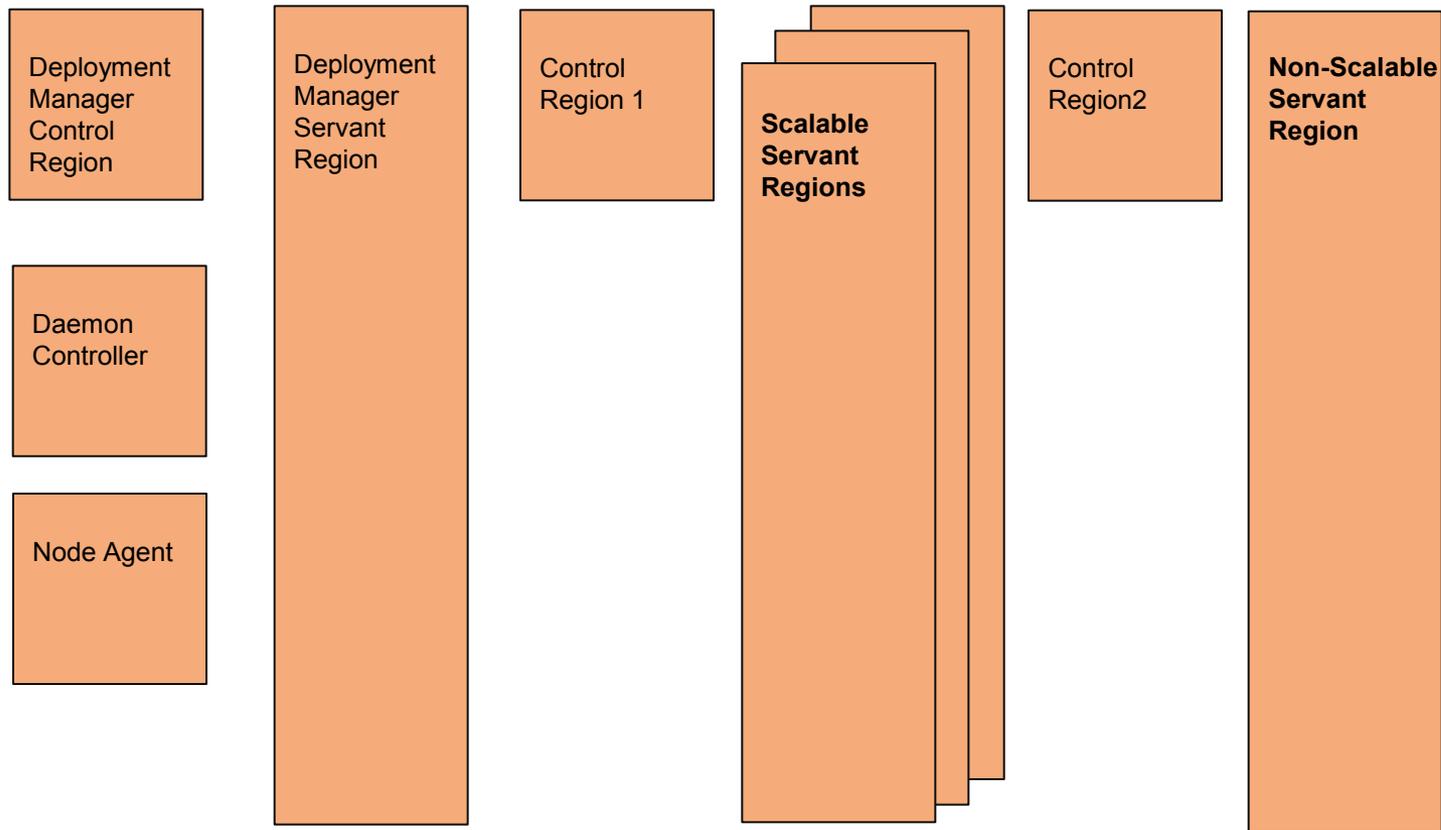
# Clustering – Multiple Servant Regions



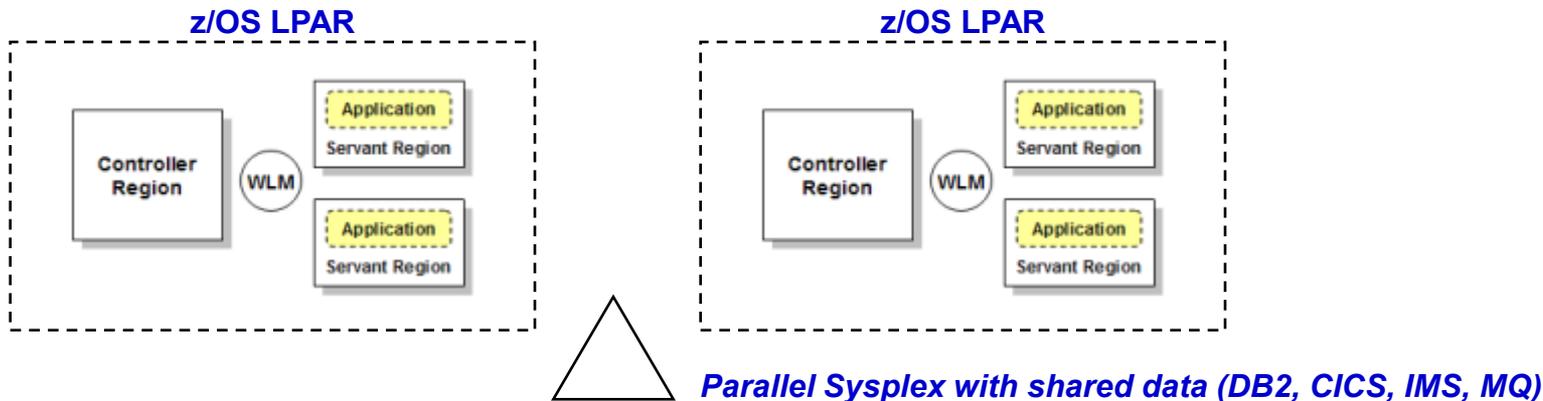
## Overview and Considerations:

- **Provides multiple application JVM instances behind CR**  
*Vertical scalability as well as protection against outage of any given application JVM instance*  
*WLM will automatically restart failed servant region*  
*Several work distribution methodologies available (favor first, stateful balance, stateless dist.)*  
*Affinities understood and maintained by WLM*
- **HTTP Session replication between servant regions supported**  
*Configure a “replication domain” or “session persistence” just like other WAS cluster*
- **Loss of CR implies loss of all SRs (so this is not complete HA picture)**  
*Use of other WAS cluster provides multiple CR/SR structures across LPARs*

# zWebSphere Configuration



# Clustering – Multiple Application Servers

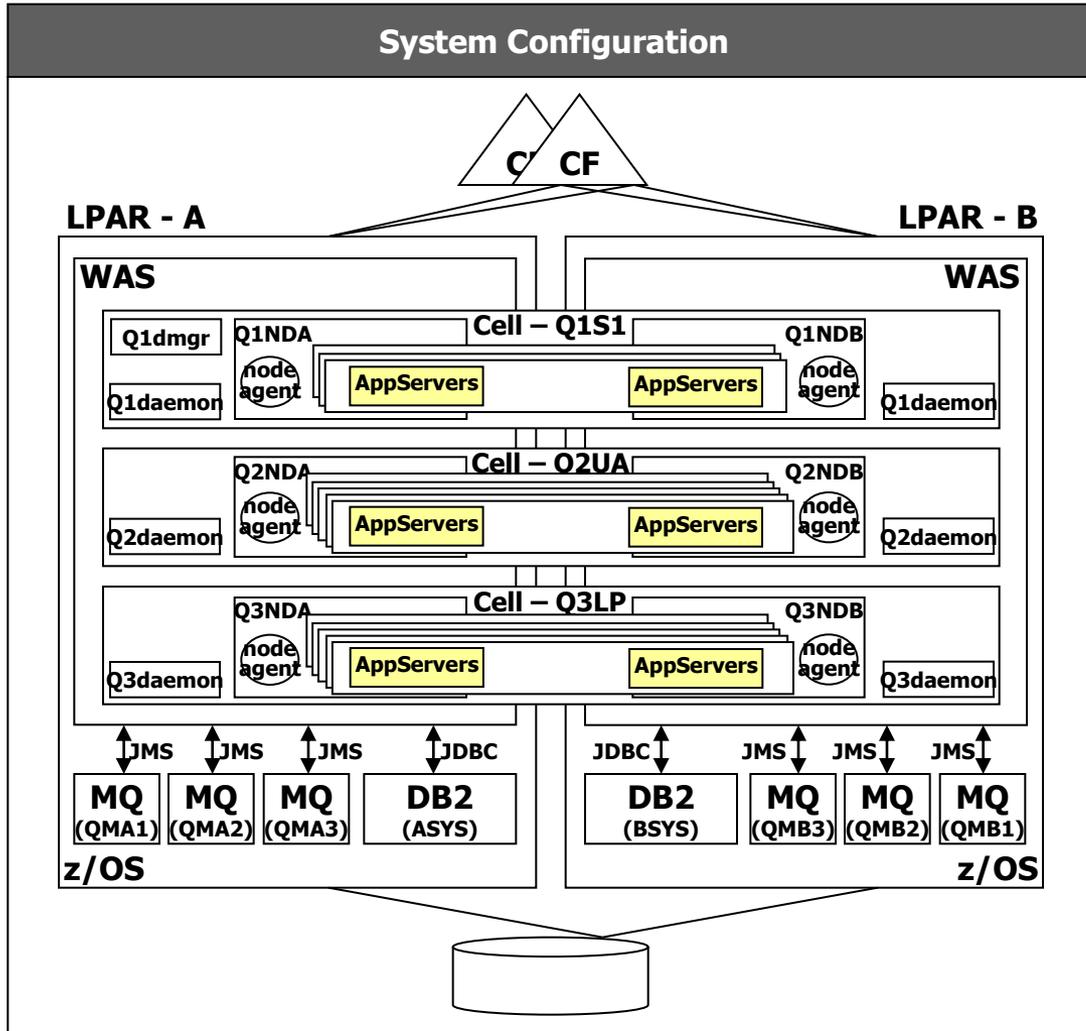


## Overview and Considerations:

- **WAS “horizontal” cluster – multiple CR/SR structures**  
*Provides protection against outage of CR or LPAR*  
*Picture shows multiple SR each ... not necessary, but is another level of HA*
- **HTTP Session replication between application servers supported**  
*Configure a “replication domain” or “session persistence”*
- **Parallel Sysplex provides data sharing across LPARs**  
*Key IBM data facilities Sysplex-aware and data-sharing enabled (DB2, CICS, IMS and MQ)*
- **Inbound request distribution must be considered (HTTP, MDB)**
- **Transaction failure and recovery another consideration for HA**

# A Large U.S. Health Insurance Company

The design for maximum availability of System z, z/OS, Software, Storage and Network



## About...

- A customer-owned health benefits company in US
- Serving more than 12 million members

## Why WAS on z/OS was selected?

- The design for maximum availability of System z, z/OS, Software, Storage and Network
- The capability of WAS on z/OS that supports high transactional volume for mission-critical applications
- High QoS (Quality of Services)
- Minimized physical tiers by using local connection from WAS to DB2
- Removes of any N/W latency and overhead by using single tier of WAS and DB2

# Questions?

