WebSphere MQ for z/OS as the messaging backbone for SOA, Web 2.0 and file transfer

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both. For a complete list of IBM trademarks please visit www.ibm.com/legal/copytrade.shtml

CICS	**IBM Logo**	**S/390**
DB2	**IMS**	**Tivoli**
E-business logo	**iSeries**	**VM/ESA**
ESCON	**MVS**	**VSE/ESA**
eServer	**OS/390**	**WebSphere**
FICON	**pSeries**	**z/OS**
IBM	**Rational**	**zSeries**
Smart SOA	**RS/6000**	**System z**

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Microsoft trademark guidelines

Intel is a registeredtrademarksof Intel Corporation in the United States and other countries.

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

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.

Agenda

- **The role of WebSphere MQ in SOA**

- **WebSphere MQ for z/OS V6 highlights**

- **What's new in WebSphere for z/OS MQ V7**

- **WebSphere MQ and the Web**

- **WebSphere MQ and Files**

- **Summary**

WebSphere MQ – What and Why?

Transport for both service-oriented and non-service-oriented IT assets
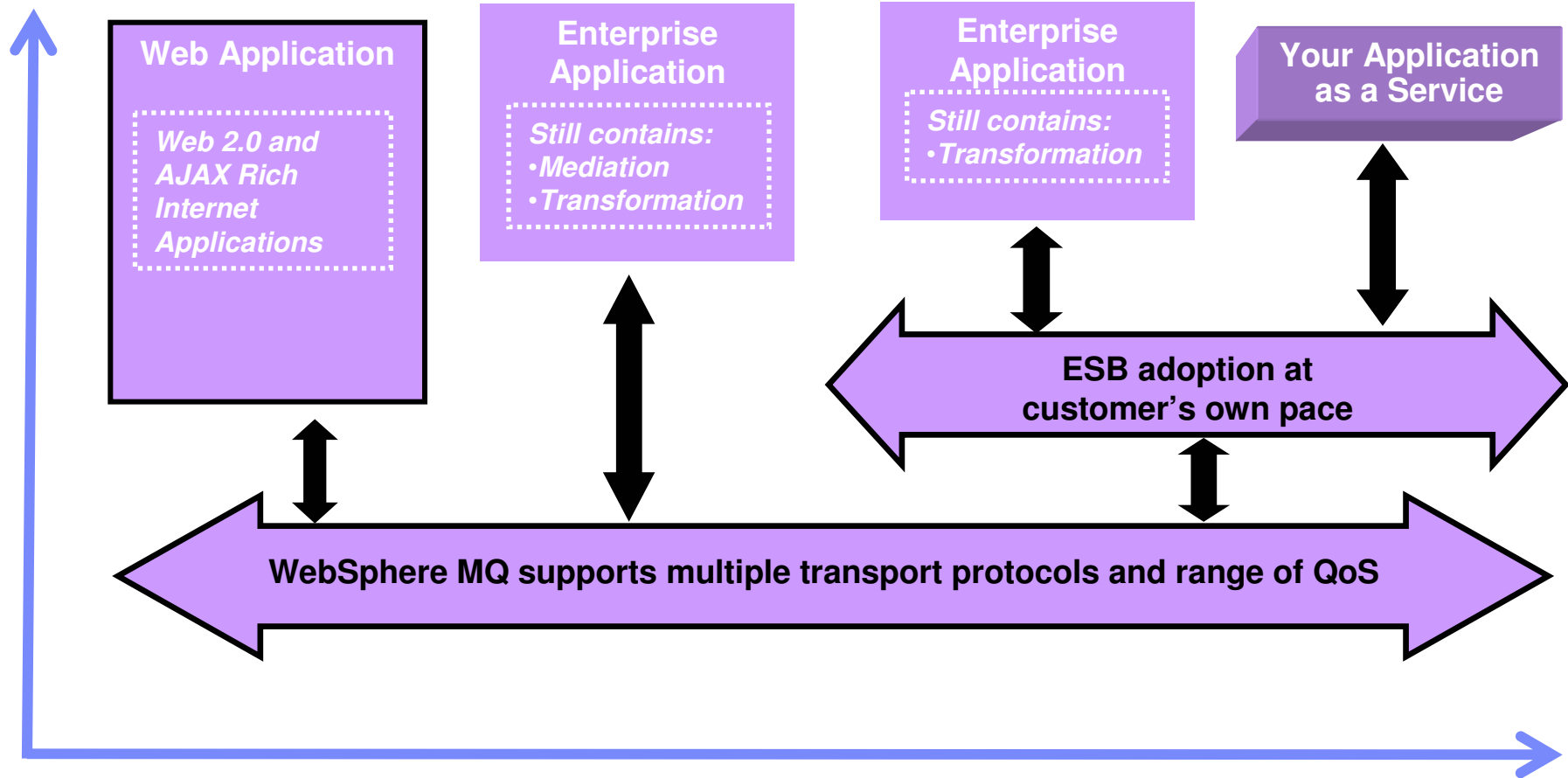
WebSphere MQ moves:
– *Data*
– *Messages*
– *Events*
– *Files*
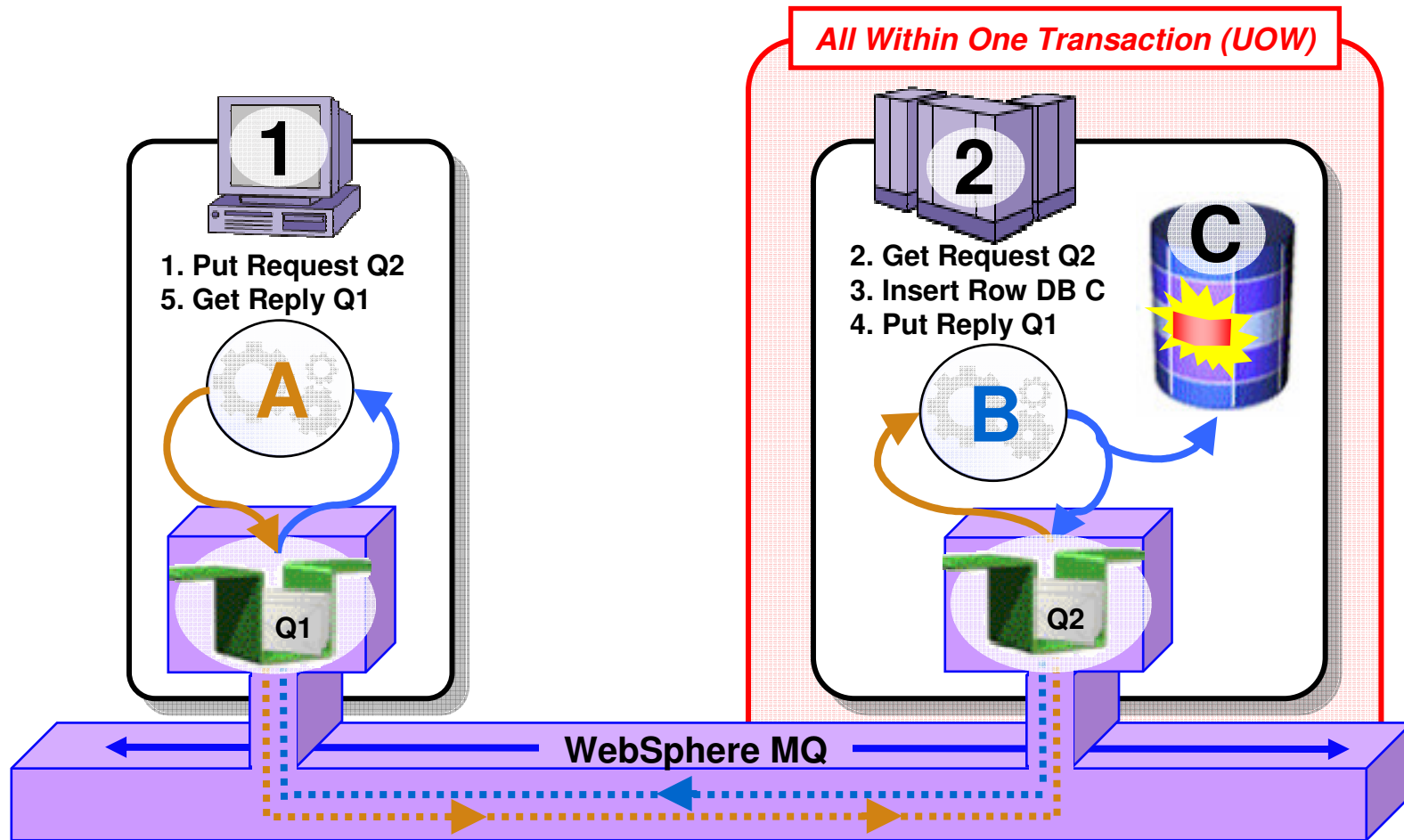– *Web service requests / responses*

7 Keys to SOA Connectivity –

Reliable – proven, trusted, the standard
Secure – protects data end-to-end
Time-flexible & Resilient – connect whenever
Transactional – preserves integrity of data
Incremental – grows with your needs
Ubiquitous – connect virtually anything
Basis for ESB – underpins and extends`

Can you connect the pieces in your company reliably and flexibly?
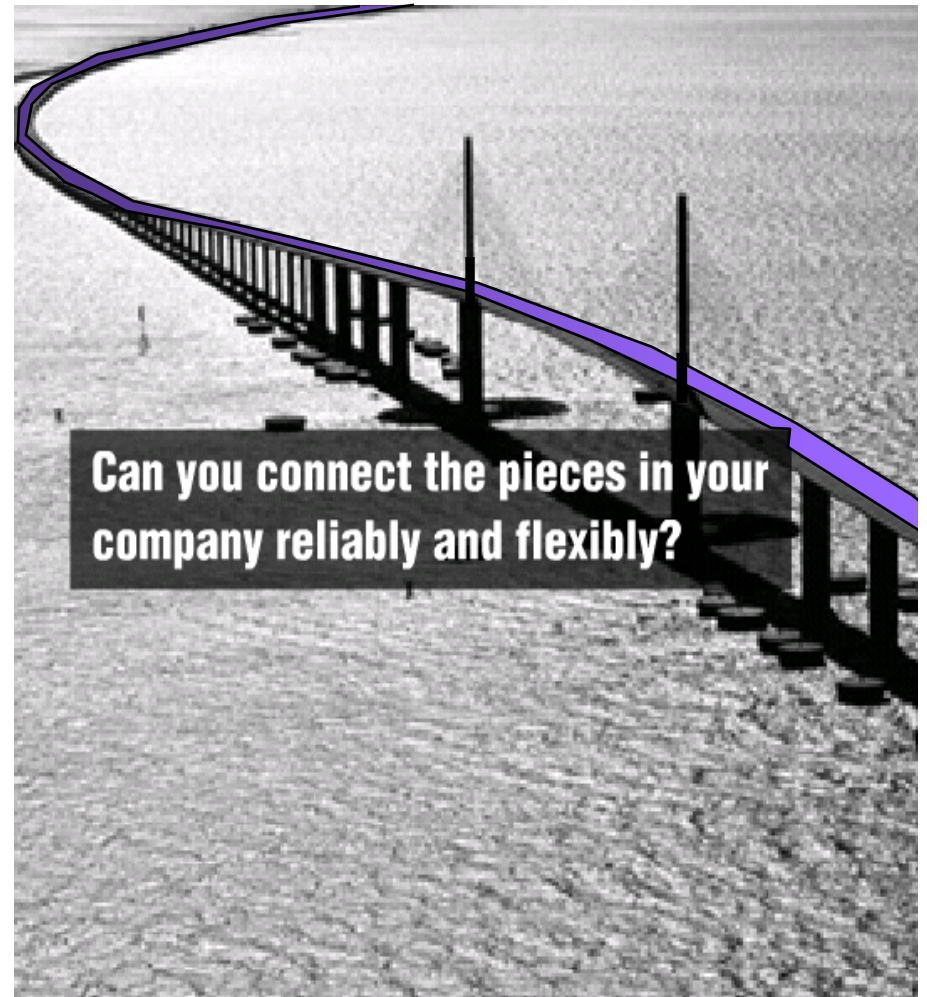
WebSphere MQ vision is to provide universal messaging backbone for the entire customer enterprise

Web Application

Web 2.0 and AJAX Rich Internet Applications

Enterprise Application

Still contains:
•Mediation
•Transformation

Enterprise Application

Still contains:
•Transformation

Your Application as a Service

ESB adoption at customer's own pace

WebSphere MQ supports multiple transport protocols and range of QoS

Full support for transactional environments

All Within One Transaction (UOW)

1
1. Put Request Q2
5. Get Reply Q1

A

Q1

2
2. Get Request Q2
3. Insert Row DB C
4. Put Reply Q1

C

B

Q2

WebSphere MQ

- Full support for CICS, IMS, WAS, DB2 SPs & Batch RSS
- Reliable Two-Phase commit involving other Resource Managers e.g. DB2

Shared queues on z/OS

- **Exploits Parallel Sysplex**
- **Automatic load balancing**
- **Scalable throughput**
- **Multiple processors can access the same queue**
- **Queue sharing groups**
- **VIPA support**

Highly Available with Failover for z/OS

- **Failure isolation**
  - Automatic peer recovery for failing Servers, Applications or Queue Managers
- **In-flight MQPUTs and MQGETs are rolled back**
  - Since messages are not hosted in any Queue Manager there are no marooned messages
- **24 x 7 availability**
- **Leverages ARM (Automatic Restart Manager)**

Parallel Sysplex

3

Queue Sharing Group

Coupling Facility

MQGET

Shared Queue

1

MQPUT
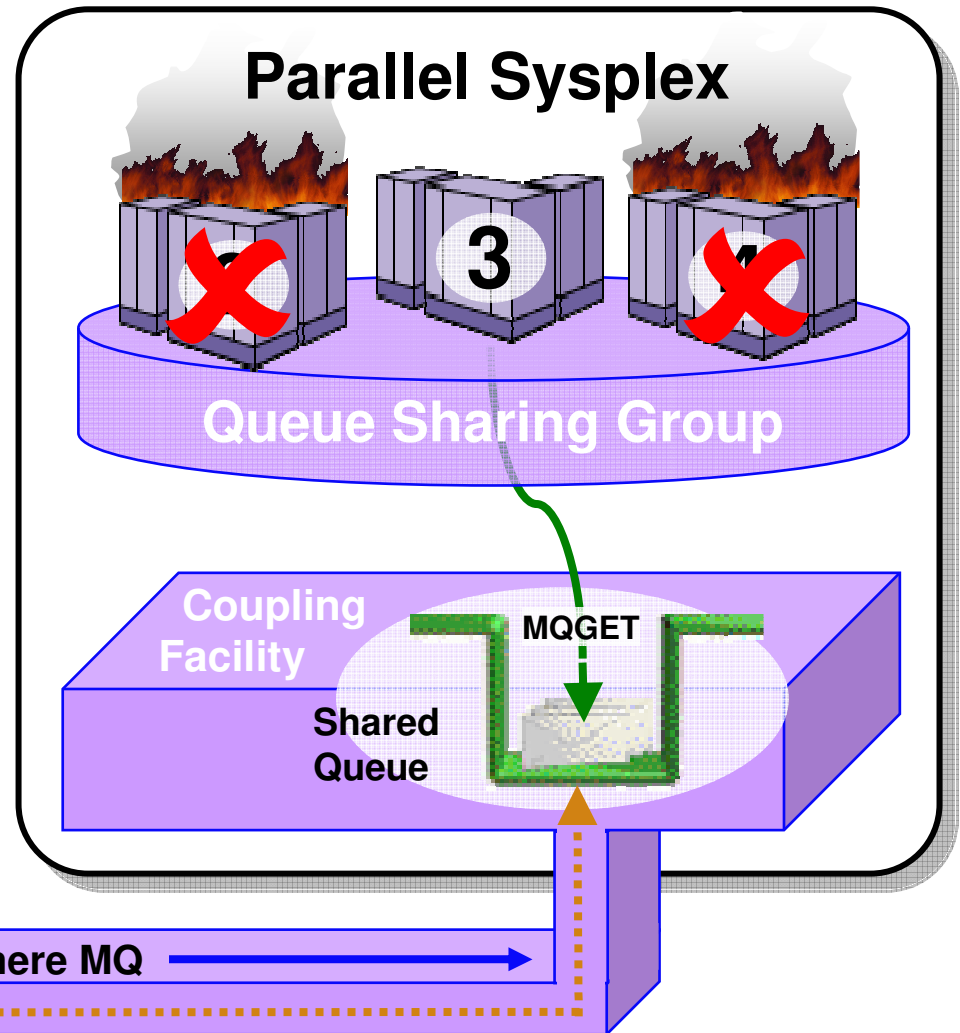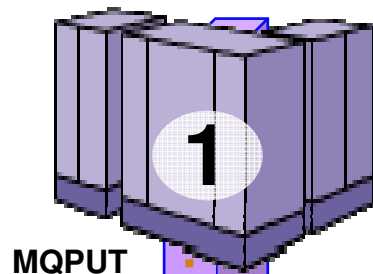
WebSphere MQ

Security

- **Security is #1 concern of CIOs**

- **Secure Sockets Layer (SSL)**
  – Provides channel-level security
  – WebSphere MQ V5.3 introduced SSL support
    • Uses the z/OS Cryptographic Services System SSL function

- **Security Server on z/OS (previously RACF)**
  – WebSphere MQ for z/OS has always provided close integration with RACF and 3rd-party alternatives
  – Security Server (RACF) can protect the resources that WebSphere MQ for z/OS owns and manages from access by unauthorized users

- **WebSphere MQ Extended Security Edition for z/OS**
  – Provides advanced security capabilities leveraging Tivoli security technology
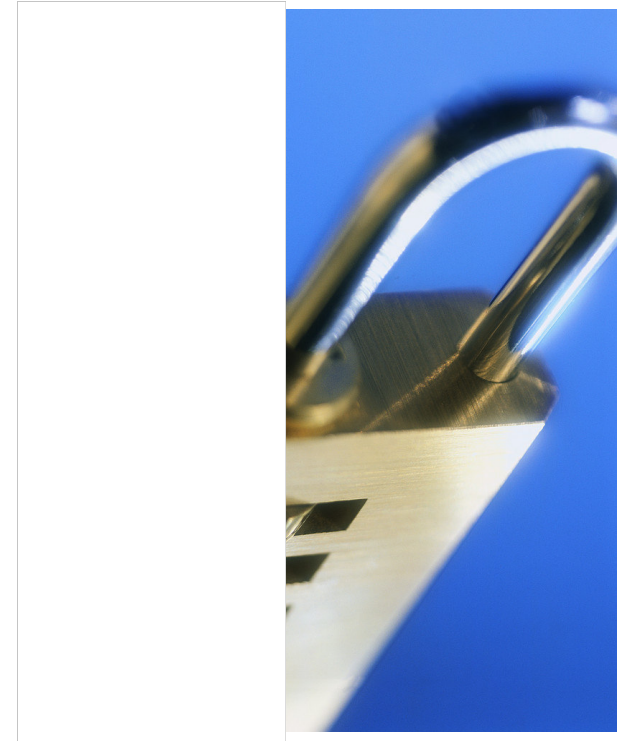
Standards - JMS 1.1 and J2EE

- **JMS 1.1 is latest level of standard, and is required for J2EE compliance**

- **Joins pub/sub and point-to-point methods into a single set of methods**
  - Backwards-compatible
  - JMS applications can still use the older 1.02 APIs

- **WMQ JMS is always supported inside WAS and now also for WebLogic**
  - It is not the "embedded" provider for WAS6, but still (of course) can be used

- **V6 also includes MA0C Publish/Subscribe component on distributed platforms only**
  - Making it easier to use full JMS function without downloading or buying extras

XMS

- **Common messaging API across IBM providers**
  - Transport neutral - WMQ, Real-Time, and WAS V6 Messaging
  - JMS-like API for other languages
  - Simplified Point-to-Point and Publish/Subscribe messaging
- **Decoupled administration**
  - XMS and JMS use the same administered Connection Factories and Destinations
- **XMS 1.2 for C/C++ available as SupportPac IA94**
  - Category 3 SupportPac
  - Initially for Windows, Linux, Solaris
- **XMS 1.2 for .NET available as SupportPac IA9H**
  - Now a Category 3 SupportPac
  - "Fully managed" code for WMQ, WAS Messaging and Real-Time clients
  - .NET support enables C#, VB .NET etc. languages
- **Inclusion with WMQ in future Fix Pack or Version**

Improved capabilities for WMQ Clusters

- **New options make large clusters easier to handle**

  - Based on real customer scenarios

  - Providing greater flexibility in prioritising and assigning "weights" to the default round-robin algorithm

- **Can create backup networks that are automatically used when all prime servers are unavailable**

- **Can restrict the number of active outbound channels**

- **Can override the "always select local queue if it exists" rule**

- **Can ask for more work to be sent to more powerful servers**

- **No automatic feedback from servers; attributes could be adjusted by system monitors**

Improved z/OS Monitoring, Management and Availability

- **Improved monitoring and problem determination**
  - Who is using the Qmgr
  - Additional status information
- **PCF support on z/OS**
  - V6 explorer can extend its administration capability to z/OS V6 Queue Managers
- **Improved Availability for z/OS**
  - 63KB message size limit for Shared Queue messages now removed
  - Dynamic recovery from Shared Queue Administration Structure failure
  - Non-shared message pagesets can now be up to 64GB in size
  - Dynamic addition and deletion of pagesets and buffer pools
  - Dynamic addition and deletion of buffers in a buffer pool
  - Dynamic addition of active log datasets
  - Pre-emptive pageset expansion
  - Dynamic setting of Distributed Queueing (CHIN) parameters
  - Multiple TCP/IP stack support
  - Multiple browsers for the CICS Bridge
  - SUSPEND/RESUME IMS Bridge commands
- **Improved support in CICS TS V3.2**
  - CICS-MQ Adapter, CICS-MQ Trigger Monitor, CICS-MQ Bridge integrated in CICS TS 3.2
  - Exploitation of CICS OTE, improved diagnostics and statistics

IS02 Supportpac: Message Broker Admin Plug-in
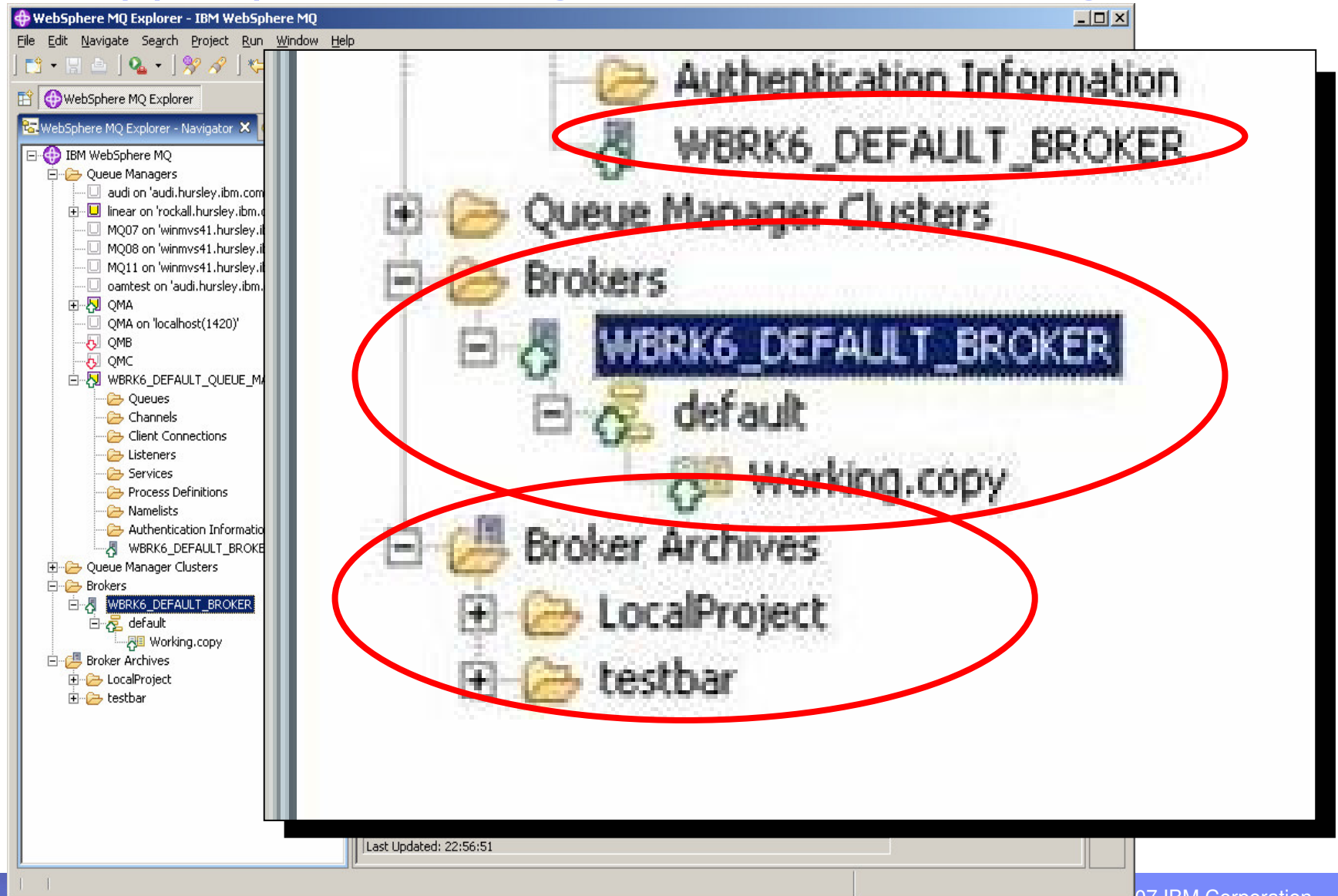
WebSphere MQ for z/OS V7.0
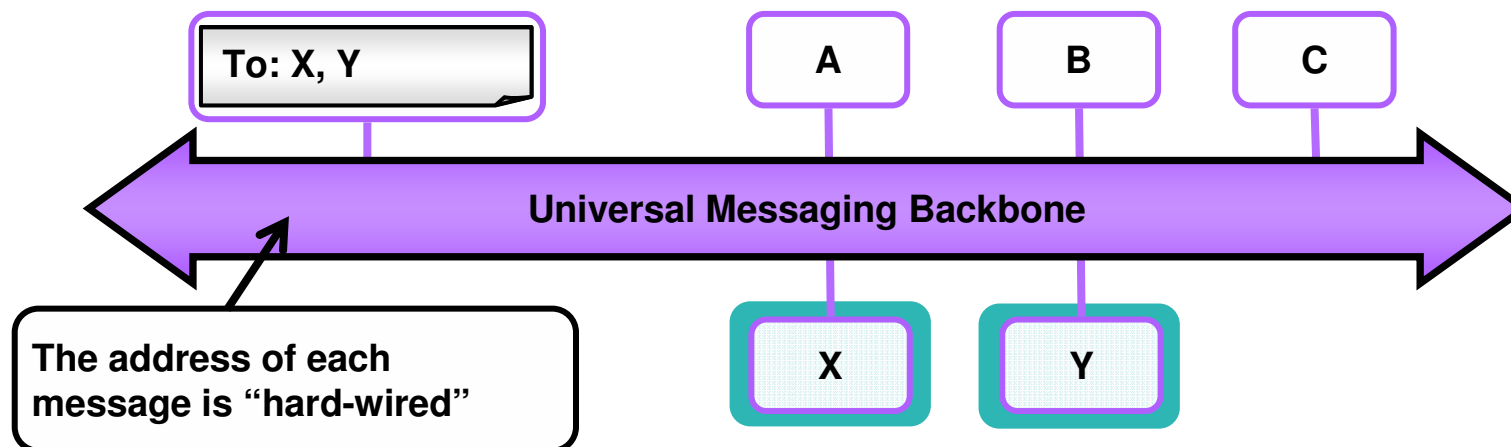
◄ UNIVERSAL MESSAGING BACKBONE ►

- **Latest Evolution of IBM's Universal Messaging Backbone for z/OS**
  - Enabling reliable connectivity from z/OS to virtually any other commercial IT system
  - Providing the messaging transport layer to underpin SOA, Web 2.0 and your ESB

- **New Publish and Subscribe support for z/OS**
  - Provides flexible, dynamic routing of messages based on topics or keywords helping reduce time needed for solution changes
  - Supports native MQ Interface and JMS API

- **Enhanced Ease of Use**
  - Remote, graphical configuration of JMS and Publish-and-Subscribe
  - Via Eclipse-based MQ Explorer now enabled for up to 5 connections without need for Client Attach Feature license

- **Enhanced JMS performance**
  - Increasing JMS listener throughput by up to 220%*

- **New MQI verbs providing greater flexibility for:**
  - Selecting messages for processing via SQL queries
  - Adding custom properties to messages
  - Automatically notifying apps when messages arrive

- **Client Enhancements**
  - Heart-beat monitoring of client connections
  - Pre-emptive delivery for increased throughput
  - New Quality-of-Service that avoids waiting for confirmation of delivery – enables "receipts" to be received later

- **WebSphere MQ goes Web 2.0!**
  - Helps enrich Web 2.0 applications with real business data from z/OS applications
  - Web 2.0 developer needs no MQ skills to use
    - Uses Ajax and simple RESTful interface to access data by URIs
  - Helps simplify deployment and maintenance of large scale distributed applications
    - Enables simple access to MQ without need to install MQ clients

* Preliminary results observed on pre-release level code.
For the latest performance information please click on *Performance Reports* at www.ibm.com/webspheremq/support
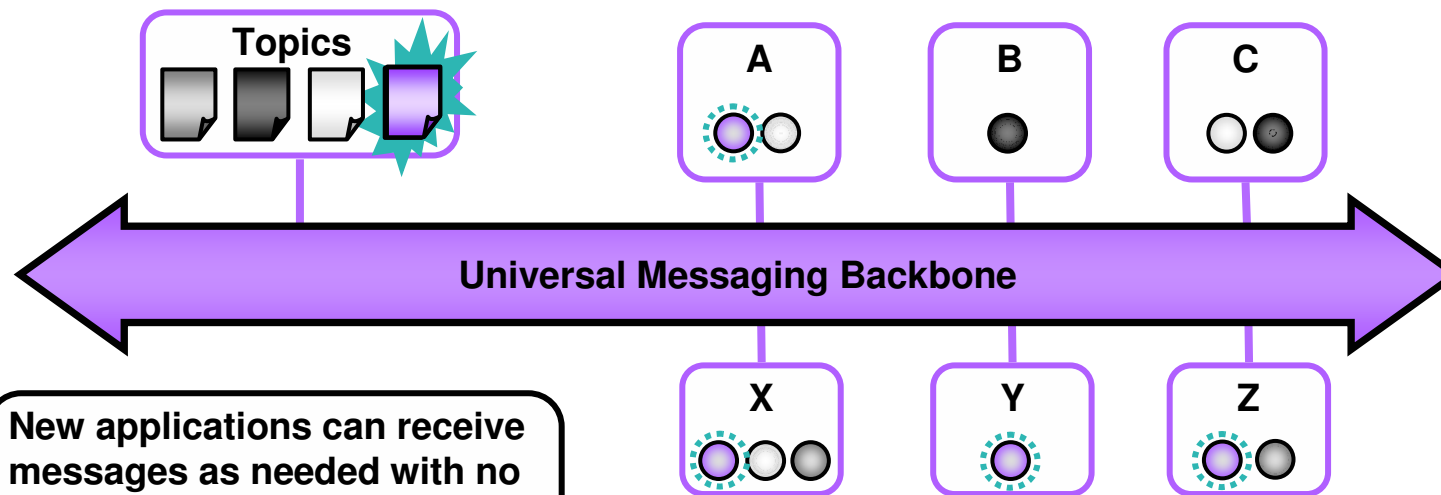
Publish/Subscribe is available for the first time on z/OS

- **A natural part of the JMS API**
  - Combines both Publish/Subscribe and Point-to-Point patterns
  - Now also a natural part of the native MQI
- **Point-to-point asynchronous messaging decouples applications**
  - But still implies a one-one relationship between sender and receiver
- **Publish/subscribe is a further stage of decoupling**
  - Sender has no direct knowledge of how many (if any) apps will see a message
  - Link between applications is a Topic, not a Queue
- **WMQ V6 (Distributed) included a Publish/Subscribe broker (formerly MA0C)**
  - Compatibility mode available in V7
- **Implementation substantially improved with V7**

To: X, Y

A   B   C

Universal Messaging Backbone

X   Y

The address of each message is "hard-wired"

Publish/Subscribe is available for the first time on z/OS

- **A natural part of the JMS API**
  - Combines both Publish/Subscribe and Point-to-Point patterns
  - Now also a natural part of the native MQI
- **Point-to-point asynchronous messaging decouples applications**
  - But still implies a one-one relationship between sender and receiver
- **Publish/subscribe is a further stage of decoupling**
  - Sender has no direct knowledge of how many (if any) apps will see a message
  - Link between applications is a Topic, not a Queue
- **WMQ V6 (Distributed) included a Publish/Subscribe broker (formerly MA0C)**
  - Compatibility mode available in V7
- **Implementation substantially improved with V7**

Topics

A   B   C

Universal Messaging Backbone

X   Y   Z

New applications can receive messages as needed with no modifications required to the other applications

Publish/Subscribe in the WMQ Explorer

Publish/Subscribe Topologies

- **WMQ V6 publish/subscribe networks based on hierarchies**
  - All brokers linked in parent/child tree
- **WMB publish/subscribe networks based on hierarchies of collectives**
  - All systems in a collective are connected to each other (mesh)
  - Also has "clones"
- **New product built on hierarchies and clusters**
  - With interoperability to other pub/sub systems through hierarchies
- **Clusters are built on existing WMQ cluster technology**
  - Cluster can be defined independently of any existing cluster used for queuing
  - Behaves in a similar way to WMB collective
- **Design gives**
  - Scalability
  - Availability
  - Ease of administration

Publish/Subscribe Application Programming

- **Cannot significantly change the JMS API**
  - But we want some of its facilities more easily available in the MQI
  - To improve MQI programming and improve (make thinner) the JMS layer
  - JMS implementation exploits new MQI functions
- **New verb for subscribing**
  - So you do not need to build RFH or RFH2 headers in the application
  - **MQSUB** registers a subscription
    - Includes information about where messages will be read from
    - Do not need to specify a queue – can be automatically assigned
  - Retained publications delivered immediately after subscribing
- **New options on existing verbs**
  - MQOPEN to get access to a topic
  - MQCLOSE deregisters a subscription
  - MQPUT, MQGET to publish and to receive publications
- **Sample programs included to demonstrate use**

MQI Enhancements - Message Properties

- **Arbitrary values associated with the message but not part of the body**
  - Like a user-extendable MQMD
  - Already part of JMS
- **New verbs including** MQSETMP **and** MQINQMP
  - Properties can be integers, strings, boolean, etc.
- **Easier to use than RFH2 folders**
  - Receiving apps do not see them unless they want
  - No need to parse and skip over message headers
- **Configuration options for compatibility**
  - Queue and channel attributes define behaviour
  - Defaults will create RFH2 folders
- **Permits explicit statement of relationships between messages**
  - eg Message X is a **REPLY** to Message Y
  - Messages referred to by handles

```
MQMD

BODY        PROPS
            XX=YY
```
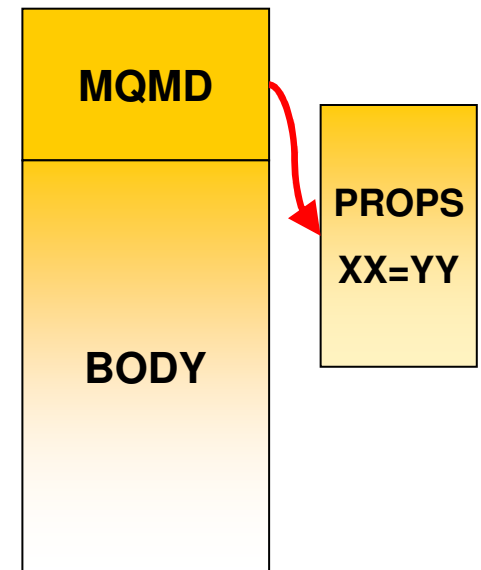
Other MQI Enhancements

- **Asynchronous Message Reception**
  - New verb **MQCB** defines a callback function
  - Automatically Invoked when a message arrives
  - No need for MQGET(WAIT) or MQGET(SIGNAL)
  - A thread can receive messages from multiple queues
  - New verb **MQCTL** to start and stop message delivery to callback

- **Selectors**
  - Use a SQL92 clause to select messages by properties including MQMD fields
  - Can be specified on MQOPEN, MQCB for filtering messages
  - Selection is done inside queue manager
  - Not looking inside message body
    - Message Broker still required for content filtering

- **Cooperative Browsing and Message Tokens**
  - Efficient interface for applications reading from the same queue
  - Example: "master" program browses a queue telling "slaves" which message to work with, based on elements within the message
  - No races – messages locked but available to any cooperating process

Client Enhancements

- **Client Performance**
  - Traditional WMQ non-persistent messages more reliable than some need
  - "Read Ahead" for Receiving Messages/Publications
  - "Asynchronous Put" for Sending/Publishing Messages
- **Client Connection Management**
  - Shared Client Conversations
  - Implementation also gives us more heartbeat opportunities
    - Faster failure notification for clients
  - Client Connections
    - Automatic workload distribution via CCDT
    - Control number of connected clients at a queue manager
  - Free connections to z/OS for administration programs like WMQ Explorer
    - Limited number of clients permitted by V7 license without CAF

WMQ Explorer Enhancements

- **Sets**
  - Queue Managers can be partitioned into sets within the Navigator
  - For example "Test", "Production"

- **Security Configuration**
  - Easy to define channel exits, userid/password configurations
  - Configured for each queue manager or for all queue managers in a set
  - Password manager included
  - Still recommend security exit or service for authentication at the server

- **Tighter JMS integration**
  - Creating an queue/topic can define a JMS destination at the same time

- **Message browser configuration**
  - Number, size of messages

- **Plug-in Migration**
  - Explorer now based on Eclipse 3.3 – compatibility not guaranteed
  - Major change is availability of supported PCF classes

Queue Manager Sets

MQ and the Web
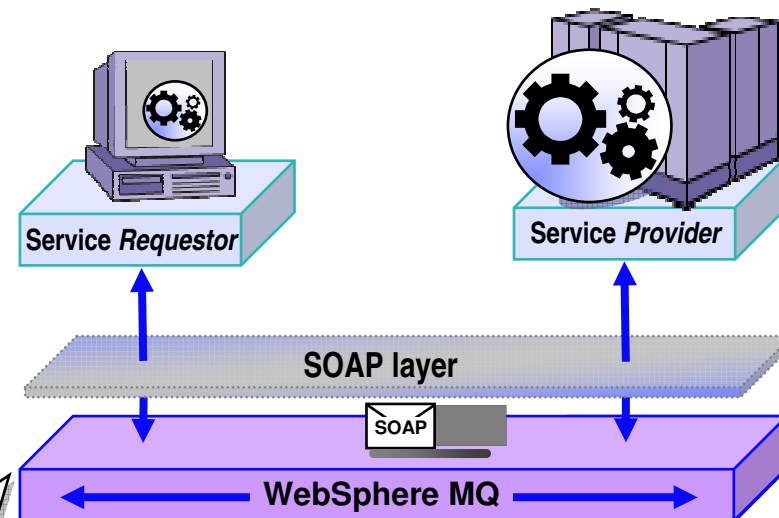
WebSphere MQ, Web services and the Web

- **SOA Backbone**

- **WebSphere MQ can be used as a Web services transport**
  - Using SOAP over JMS, or by sending SOAP messages in raw MQ messages
  - As an integrated Custom Channel for Windows Communication Foundation (WCF aka. Indigo)
    - See Alphaworks: http://www.alphaworks.ibm.com/tech/mqwcf/

- **Re-use**

- **WebSphere MQ assets can be described and re-used in Service Oriented Architectures**
  - WebSphere MQ applications described in a standard way like other Web service assets using WSDL and IRI's
  - Catalogued, stored in Service Registry, and available for ESB or any other access

- **Web 2.0**

- **WebSphere MQ can be used as a Web 2.0 enabler with the HTTP Bridge**
  - Extending the reach of Enterprise applications to the Web

WebSphere MQ makes Reliable Web services possible

- **Many developers today connect Web services using HTTP**
  - May seem "good-enough" to start with but doesn't address reliability and auditability and locks-up the services whilst they are communicating

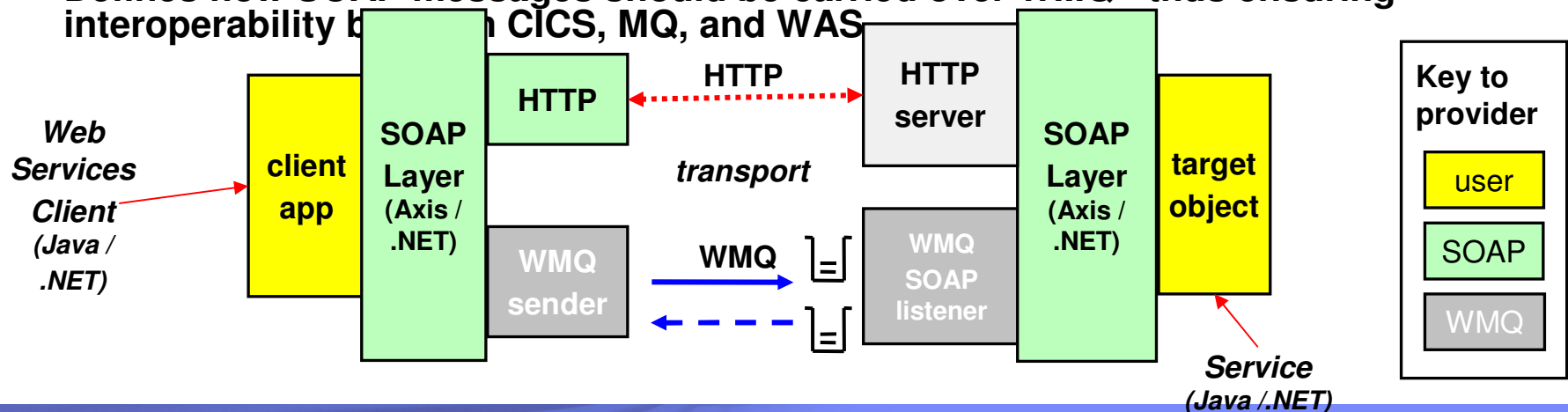**WebSphere MQ provides a <u>layer</u> of messaging services to help make SOAP exchanges:**

- ✔ *More <u>reliable</u> than those sent over HTTP*

- ✔ *<u>Transactional</u> – allowing service resources to be updated atomically so they <u>retain integrity</u> and failures can be <u>recovered</u> and retried*

- ✔ *Arrive in the <u>same order</u> as they were sent*

- ✔ *<u>Decoupled in time</u> – so that they can handle other requests whilst waiting for responses*

- ✔ *<u>Buffered</u> so that SOAP requests can be throttled for batch-style Web services*

- ✔ *Clustered so that SOAP requests can be <u>processed in parallel</u> by multiple service providers*

- ✔ *Easily <u>integrated</u> with applications that are not enabled for SOA*

- ✔ *More <u>auditable</u>*

Service *Requestor*

Service *Provider*

SOAP layer

SOAP

WebSphere MQ

- **Web services that interoperate with HTTP will also with WebSphere MQ**

- **Sender and listener support for**

  - Axis host Web services environment

  - .NET host Web services environment

Standards - WS-* and SOAP / WMQ-JMS

- **WMQ acts as a SOAP transport for WS-* Web services and this offers a number of advantages compared to HTTP transport -**

  - Improved reliability – invaluable for one-way message exchange patterns – since with HTTP there is no way for the Web service requestor to know if the request has been successful. SOAP/JMS is really capable of fire-and-forget

  - Manageability

    - WMQ / JMS messages are more tangible and traceable than HTTP requests

  - Asynchronous invocation

    - Limited (short term) asynchrony has existed in .NET for some time, and it has been added to Java JAX-WS. This does not offer any guidance on how to persist and correlate requests over a long time. True queued asynchrony adds resilience – (both parties don't need to be running at the same time), and requests can be longer running than typical HTTP request.

- **SupportPac MA0R was incorporated into WMQ v6**

- **Defines how SOAP messages should be carried over WMQ - thus ensuring interoperability between CICS, MQ, and WAS**

Web Services Client (Java / .NET) → **client app** | **SOAP Layer (Axis / .NET)** | **HTTP** ←·····HTTP·····→ **HTTP server** | **SOAP Layer (Axis / .NET)** | **target object**

transport

WMQ sender →WMQ→ **WMQ SOAP listener**

Service (Java /.NET)

Key to provider

| user |
| SOAP |
| WMQ |

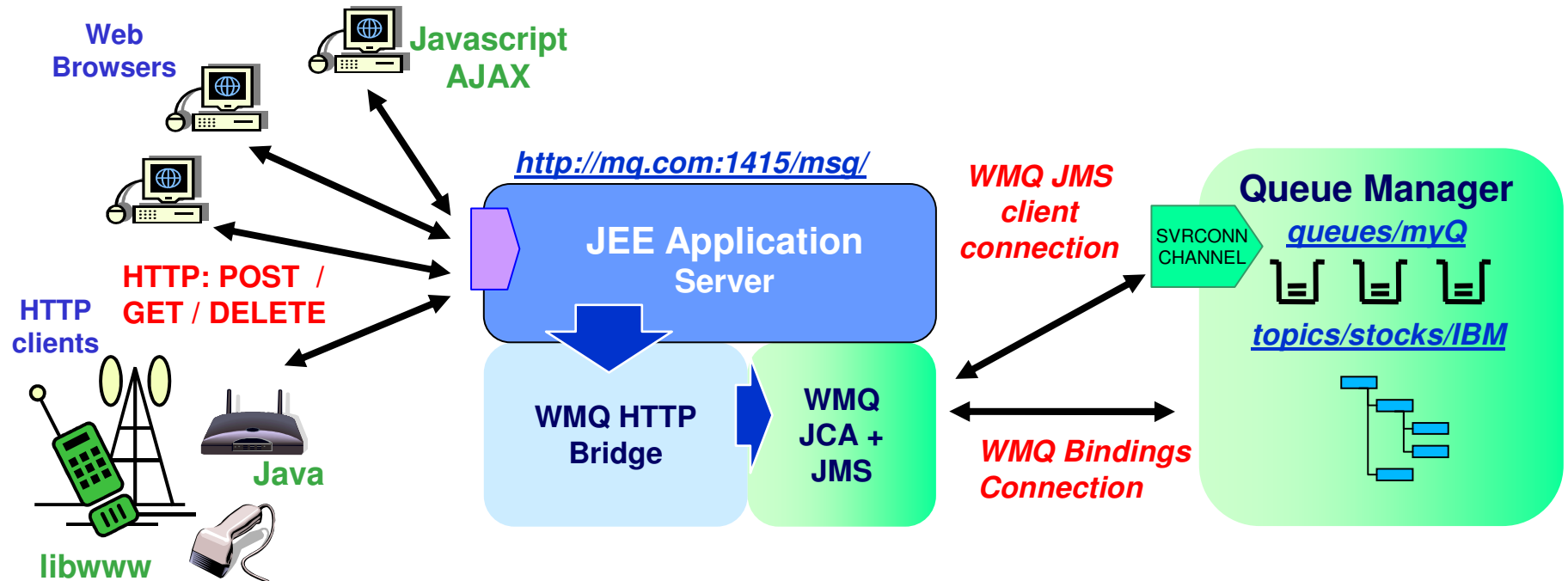# WMQ as an SOA Asset - Service Descriptions

- **A standard way to describe all WMQ apps as SOA assets (services)**

  – To be inventoried, and catalogued in Service Registry

  – To be re-used as services in composite SOA applications

  – To be managed and traced with SOA tools

- **IBM has created the WMQ Service Definition and SOAP binding**

  – IRI for WMQ addresses ("wmq:")

    • Message destinations - Queues or Topics

    • Other resources - Qmgrs, channels, channel status etc.

  – WSDL bindings to define application properties

    • Also defines the Message Exchange Pattern; Request queue; Response queue; Correlation style; Message format; Message persistence, priority etc.

- **Published as SupportPac MA93**

WebSphere MQ Bridge for HTTP Bridge Goals

- **Simplify access to MQ Apps from Rich Internet Applications**
  - Gives AJAX and Web 2.0 access to the Enterprise
    - Submit data direct to queues from Browser
    - Low Latency Web Pub/Sub
      - Stock price update
      - Sports scoreboard
      - Airport / Rail Departures / Arrivals Notification

- **Enable MQ Application Connectivity from *any* Platform or Language with HTTP capabilities**
  - Significantly increase range of supported platforms
    - e.g. – Linux distros, POS terminal running Windows Services for Unix environment, RFID reader, Mobile devices
  - Reduce cost of MQ Client deployment
    - No client library installation required
  - Lightweight (low QoS) messaging

WebSphere MQ Bridge for HTTP - Architectural Overview

Web Browsers

Javascript AJAX

http://mq.com:1415/msg/

WMQ JMS client connection

Queue Manager

queues/myQ

HTTP: POST / GET / DELETE

JEE Application Server

SVRCONN CHANNEL

topics/stocks/IBM

HTTP clients

WMQ HTTP Bridge

WMQ JCA + JMS

WMQ Bindings Connection

Java

libwww

- Key features of the WebSphere MQ Bridge for HTTP -
  - Maps URIs to queues and topics
  - Enables MQPUT and MQGET from
    - Web Browser
    - Lightweight client
- Alternative non-servlet implementation available as MA94

HTTP-MQI Verb / Resource Mapping

- **Define URI to identify queue (or topic)**

- **Modelled on REST principles**
  - Simple translation of HTTP to MQI

- **Message Format:**
  - Header fields (MQMD) conveyed in HTTP headers
  - Body is passed in HTTP entity body
  - Message type is conveyed in HTTP Content-Type
    - "text/plain" or "text/html" equate to WMQ string messages (MQFMT_STRING)
    - All other media types map to WMQ binary messages (MQFMT_NONE)

Resource	Sample URIs	HTTP verb mapping			
		GET	POST	PUT	DELETE
Messages	http://host/msg/queue/*qname/* http://host/msg/topic/*topic_path/*	MQGET w. browse	MQPUT	-	MQGET

MQ and Files

Introducing WebSphere MQ File Transfer Edition for z/OS V7.0

- **The newest member of the WebSphere MQ family**
  - Builds upon WebSphere MQ's proven transport backbone
  - Licensed as a trade-up from WebSphere MQ on distributed platforms

- **Will deliver robust solution for Managed File Transfer**
  - Enable control of all aspects of file movement between IT systems
  - Provide file delivery reliability
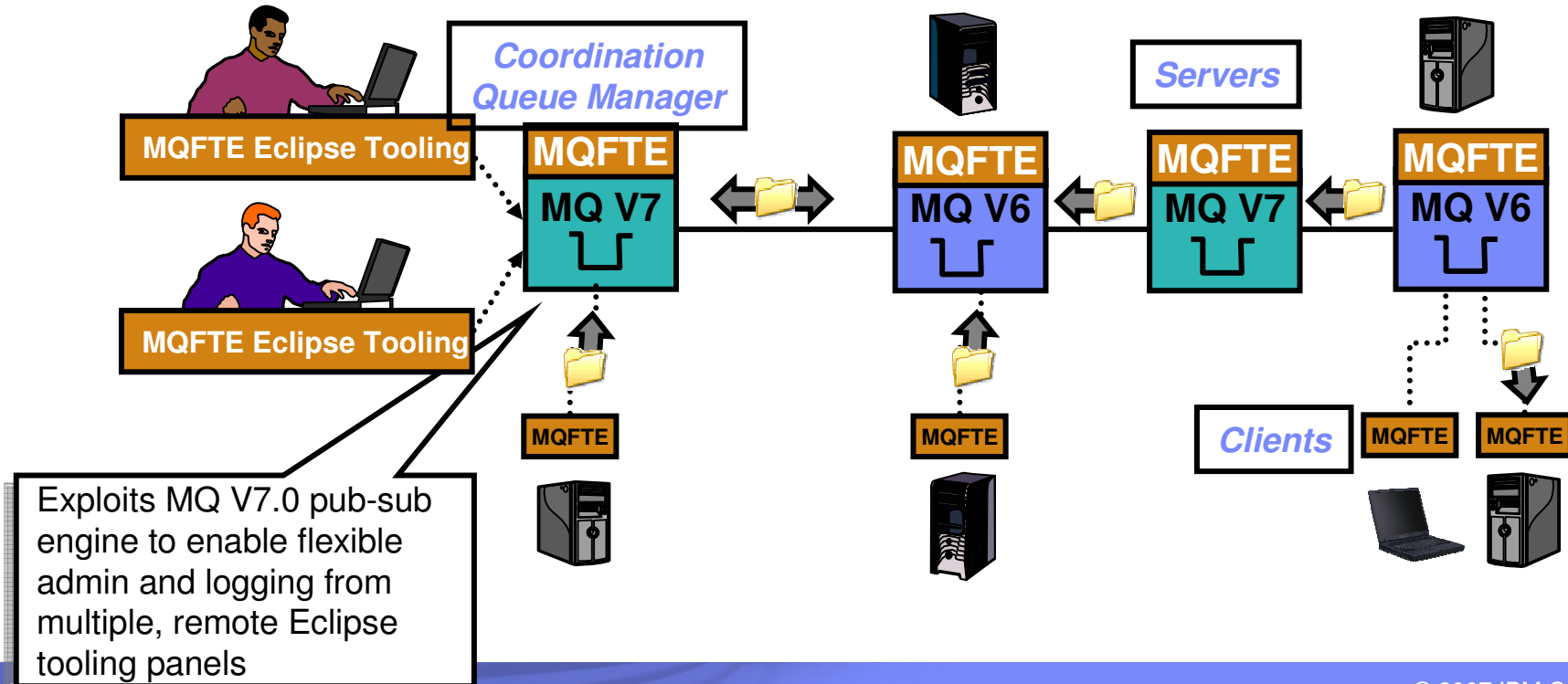 - Optimized for both small and massive files
  - Provides audit trail of transfers

- **Designed to integrate with IBM's SOA portfolio**
  - Enables files to be delivered to WebSphere Message Broker for File Processing
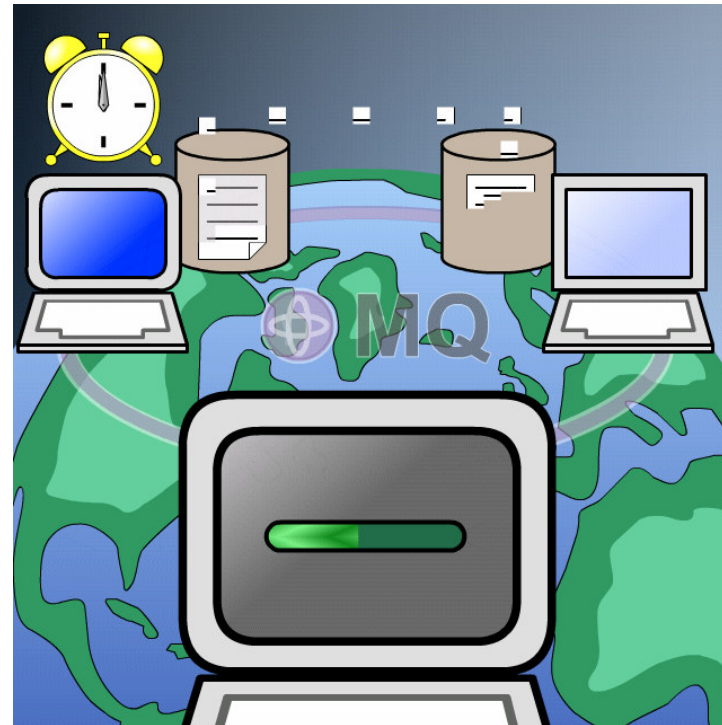
Architecture

- Enables remote GUI configuration and admin using same tooling as MQ
- Tooling publishes transfer requests to Backbone
- "Agents" running alongside Queues managers publish audit trail to Coordination Center
- "Agents" monitor file directories, load/unload files & perform pre- & post-transfer activities
- Coordination Queue manager publishes transfer status, process and audit trail
- Coordination Queue manager requires MQ V7.0
- Multiple Coordination Queue manager could control transfers, capture audit log and publish status

Coordination Queue Manager

MQFTE Eclipse Tooling

MQFTE Eclipse Tooling

MQFTE
MQ V7

Servers

MQFTE
MQ V6

MQFTE
MQ V7

MQFTE
MQ V6

MQFTE

MQFTE

Clients

MQFTE

MQFTE

Exploits MQ V7.0 pub-sub engine to enable flexible admin and logging from multiple, remote Eclipse tooling panels

Key Themes for WebSphere MQ File Transfer Edition

Auditable
Audit logs of transfers at source and target
Audit data persisted to MQ queues and/or relational database.
Captures time-stamped log at source and target

Ease-of-Use
Remote console for transfer initiation, unattended operation, scripting, scheduling, restart policies, status display
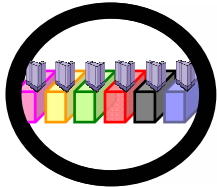Integrated with MQ Explorer configuration tooling
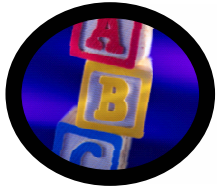
Simplicity
Small footprint, fast install
No need to write code or use API to configure transfers – Enabled via GUI
Leverages WebSphere MQ – no other technology pre-reqs

Security
Access to individual files subject to file system permissions
Link level security (inheriting MQ SSL security)

Breadth
Support WebSphere MQ V6 and V7 for transfers
Core Platform support (Windows, z/OS, Linux (32 Bit), Solaris, AIX, HP)
Good file type support (ASCII/EBCDIC, CR/LF, Flat files, z/OS QSAM, BPAM, VSAM)

Automated Transfers
Transfers can be scheduled to repeat at predetermined intervals
Transfers can be triggered by range of file system events e.g. new files, updated file, etc.

Apply to Join the Early Access Program

- What is it?
  - Opportunity to get early access to pre-GA code
  - Influence product directions and features
 - Build product knowledge and skills

- When does it start?
  - You can apply to join anytime after July 8th 2008

- How can I apply?
  1. Email filetransfer@uk.ibm.com to request to join
  2. If accepted by IBM, receive your unique numerical key and URL address
  3. Use your Key to log on to the Early Access Program site
  4. Read & accept the Program terms

- What is required to join?
  - A specific Non-Disclosure Agreement (NDA) must be signed
  - Can be signed electronically via the Betaworks Web site

- What do I receive?
  - Download regular code drops of the latest pre-GA iterations
  - Opportunity to provide product feedback direct to labs
  - Product roadmap details and materials

- Who can I contact with more questions?
  - Send an email to filetransfer@uk.ibm.com
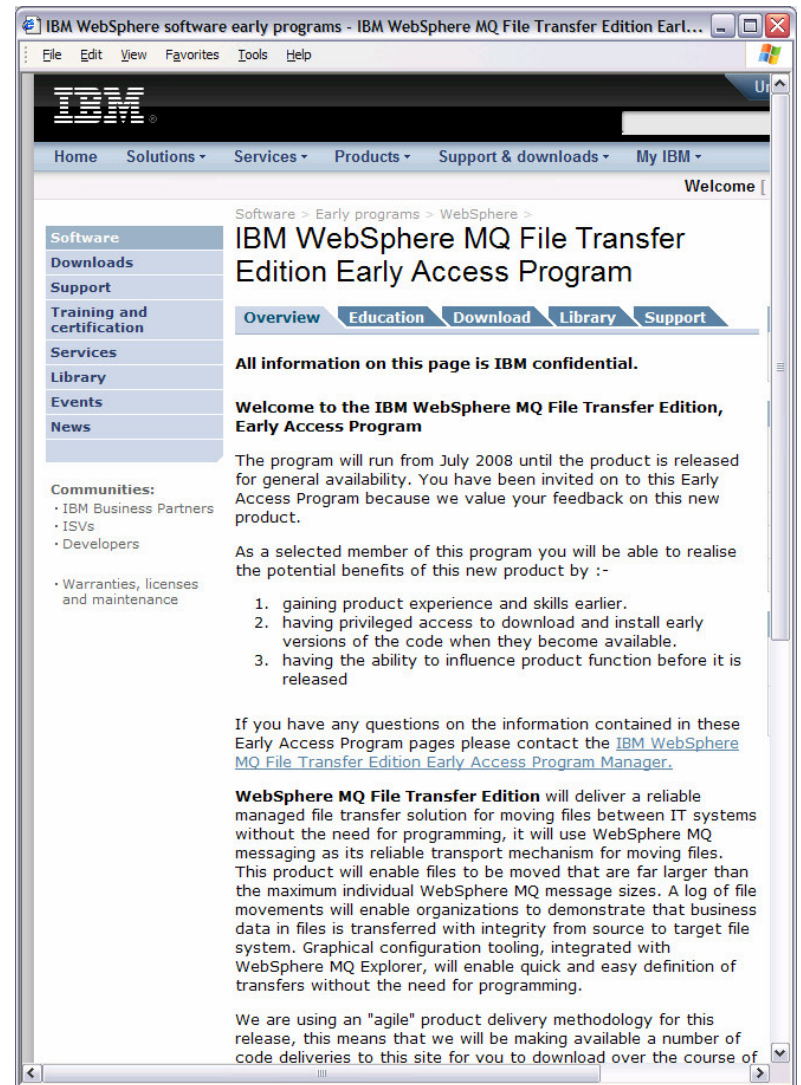
WebSphere MQ Version 7 - Summary

- **Improvements to publish/subscribe**

- **Improvements to JMS layer**

- **Ease-of-use for administrators**

- **Ease-of-use for MQI programmers**

- **Performance**

- **Continues to extend the enterprise messaging foundation in the SOA world**

- **Key Dates:**
  - Distributed V5.3 End of Service in September 2007
  - z/OS V5.3.1 End of Service in April 2008
  - Fee-based service extensions normally available for an extra 2 years
  - V6 - No End of Service or Withdrawal from Marketing dates set
  - V7 - Announcement 1 April 2008
  - V7 - GA 27 June 08
  - File Transfer edition V7.0 planned availability Dec 2008
  - Look at URL for most recent info
    - http://www-306.ibm.com/software/info/supportlifecycle/

Next Steps and more information

- **Talk with your IBM representative and IBM Business Partners to identify ways to help achieve your business goals with WebSphere MQ**

- **More information about WebSphere MQ can be found here**
  - www.ibm.com/webspheremq
  - www.ibm.com/software/integration/wmq/zos/
- **WebSphere MQ File Transfer Edition**
  - http://www-306.ibm.com/software/integration/wmq/filetransfer/

- **Additional materials**
  - White paper: The Value of WebSphere MQ for z/OS for the CICS customers ftp://ftp.software.ibm.com/software/integration/wmq/WSW11355-USEN-00.pdf
  - Article: Connecting System z applications with IBM WebSphere MQ on z/OS http://www-306.ibm.com/software/tivoli/features/ccr2/ccr2-2007-12/connecting.html
  - Telecinference: Exchanging business data and transactions with WebSphere MQ on z/OS http://www-306.ibm.com/software/os/zseries/telecon/may2/

Additional Information

Introduction to WebSphere MQ
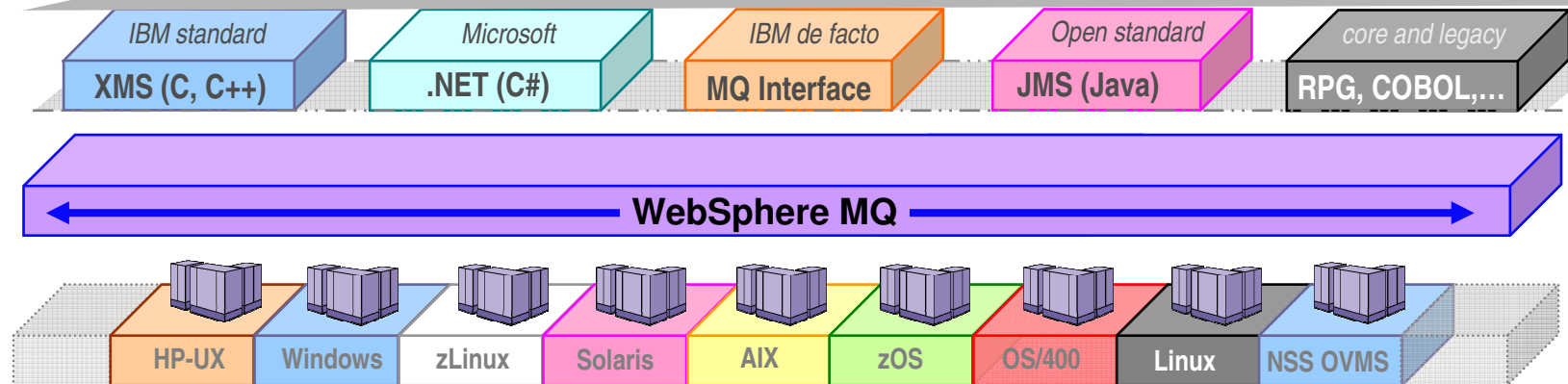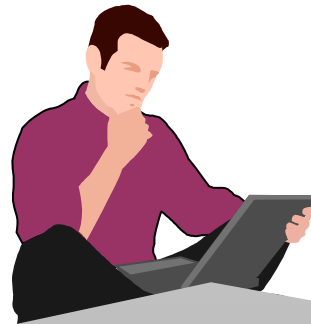# for z/OS concepts

WebSphere MQ connects virtually anything

WebSphere MQ has probably the software industry's <u>broadest</u> support for:

- *programming languages*
- *messaging interfaces*
- *application environments*
- *OS platforms*

Ubiquity of support gives developers the freedom to choose the technologies <u>they prefer</u> and <u>already have skills</u> in and can connect together what they <u>already</u> have…

IBM standard	Microsoft	IBM de facto	Open standard	core and legacy
XMS (C, C++)	**.NET (C#)**	**MQ Interface**	**JMS (Java)**	**RPG, COBOL,…**

WebSphere MQ

HP-UX	Windows	zLinux	Solaris	AIX	zOS	OS/400	Linux	NSS OVMS

80+ platform configurations

How does WebSphere MQ work?

- **Connects applications using** Messages **sent via** Queues.

- **Queues are owned by** Queue Managers **which <u>store and forward</u> messages.**

- **Routing is dynamic and configurable**

- This allows applications to be very loosely coupled…

  - Cuts location dependencies
    - Sender does not need to know <u>where</u> the receiver is running

  - Cuts timing dependencies (asynchronous)
    - Sender does not need to know <u>whether</u> the receiver is running

  - Cuts platform dependencies
    - Sender does not need to know the <u>platform</u> the receiver is running

  - Cuts data dependencies
    - With a Broker they do not even need to agree on the data format

Message

Queue

Application A

Application Z

Q Manager

Channels

Q Manager
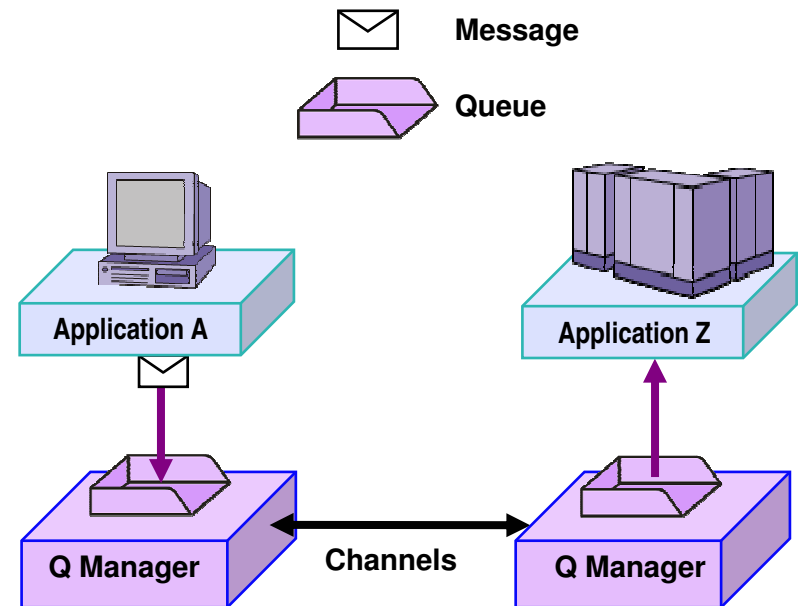
Pseudo-synchronous or asynchronous styles

Customer-facing transaction.
Pseudo-synchronous does NOT mean slow !!
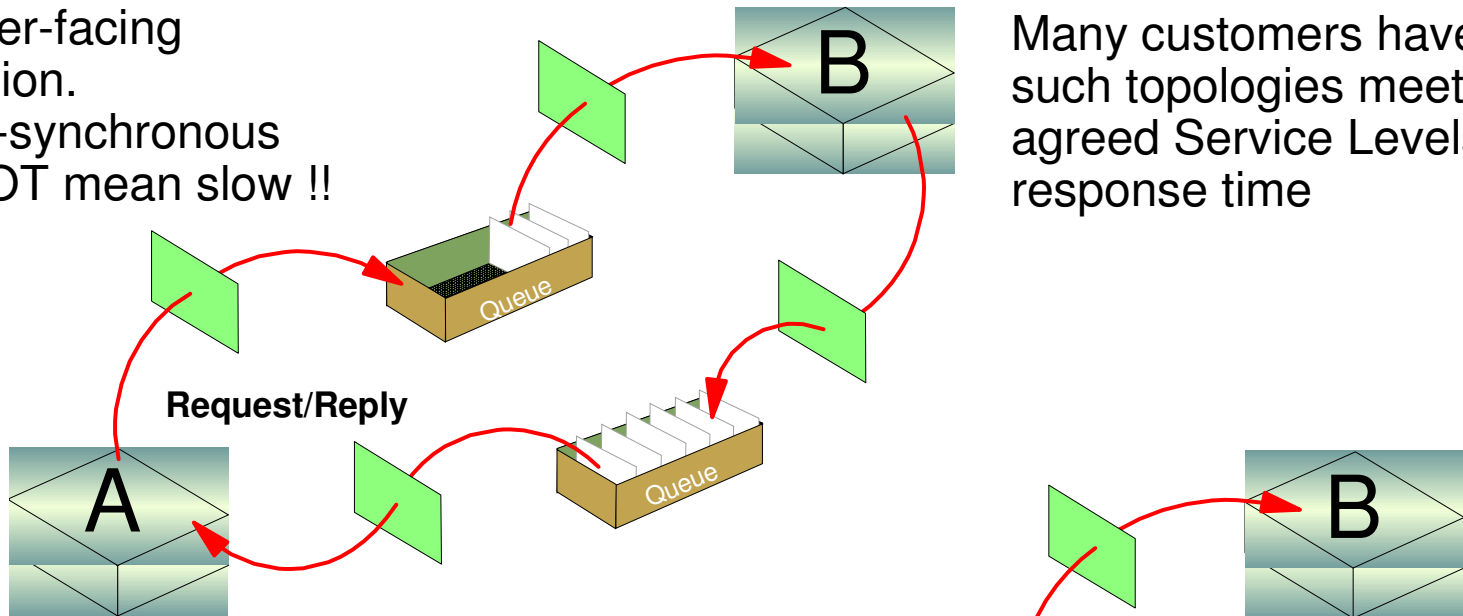
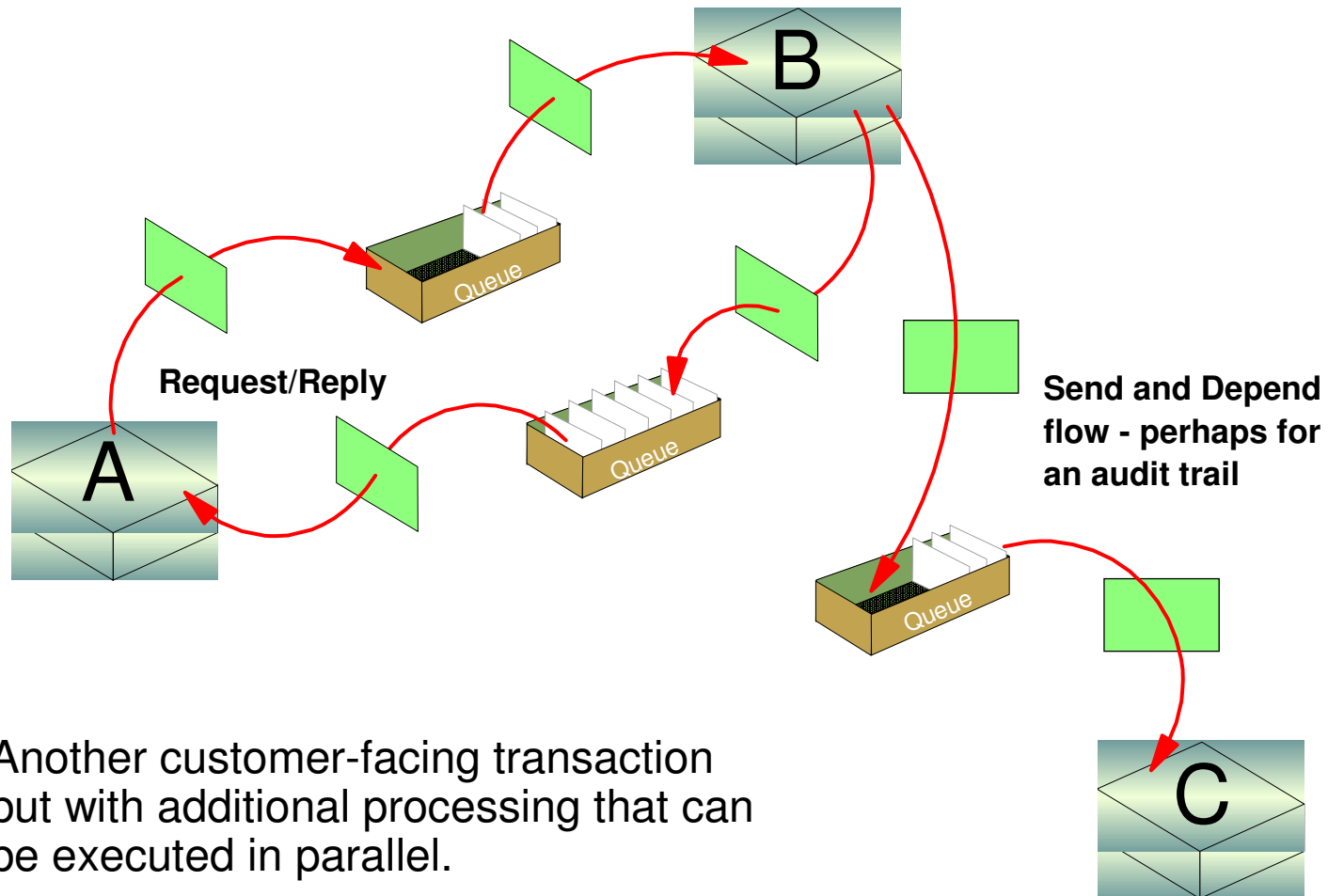Many customers have such topologies meeting agreed Service Levels of response time

Request/Reply

Fire and Forget (or Send and Depend)

Could also be a Customer-facing transaction but no response is required.
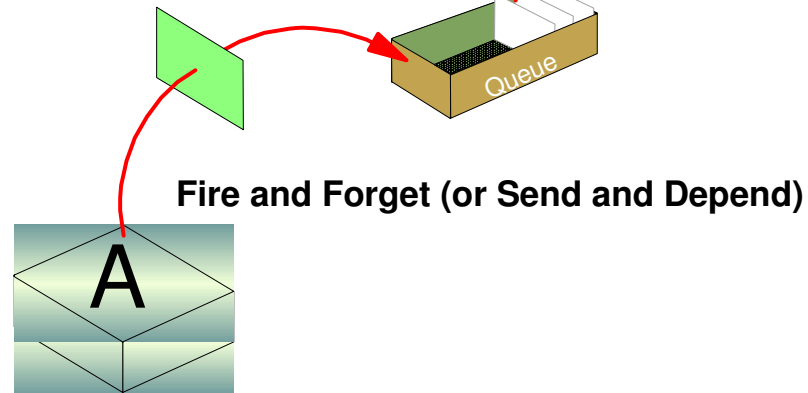
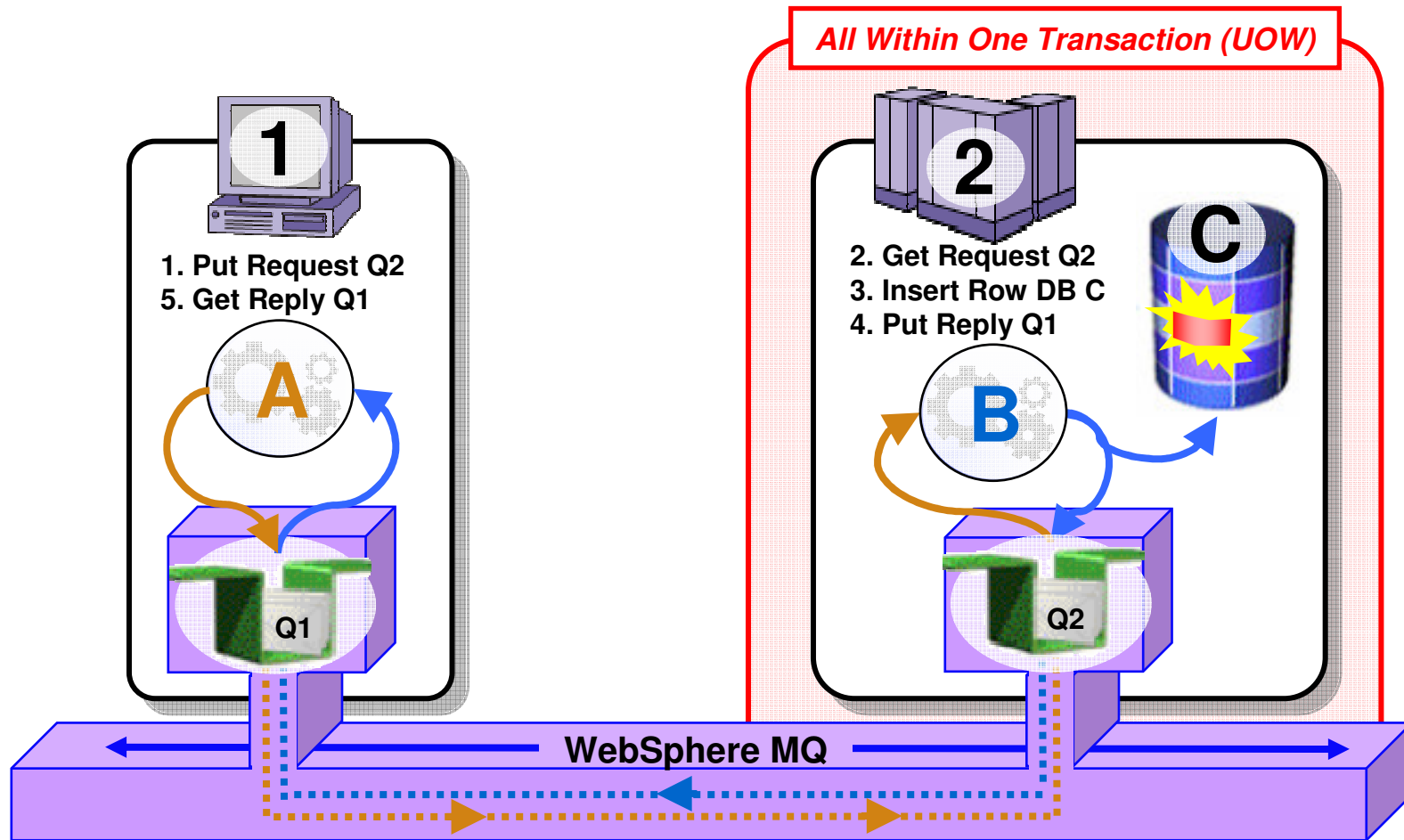# Pseudo-synchronous and asynchronous styles Enables parallel processing

Request/Reply

Send and Depend flow - perhaps for an audit trail

Another customer-facing transaction but with additional processing that can be executed in parallel.

A

B

C

Queue

Queue

Queue

Transactional support ensures all or nothing
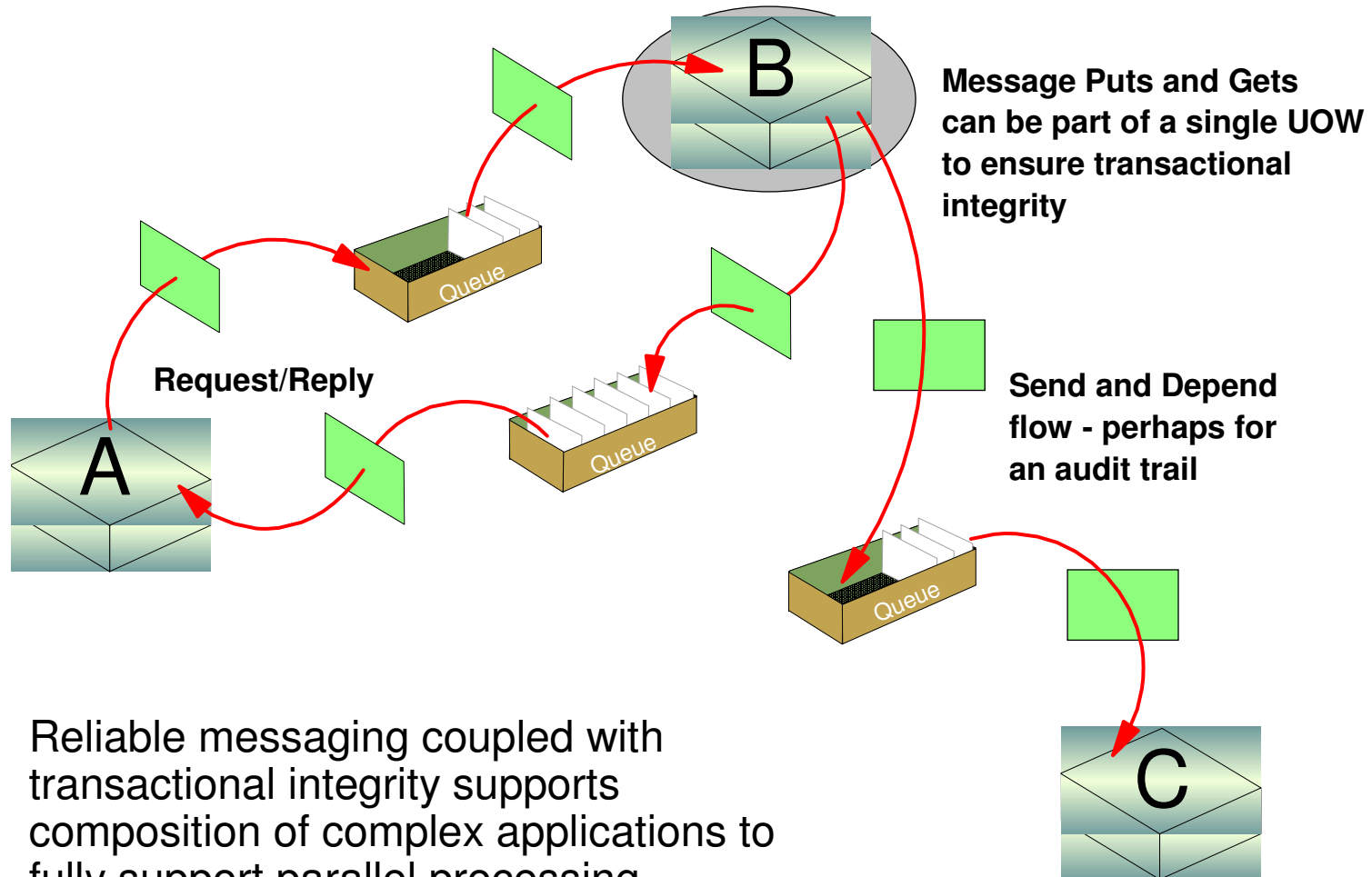
Message Puts and Gets can be part of a single UOW to ensure transactional integrity

Request/Reply

Send and Depend flow - perhaps for an audit trail

Reliable messaging coupled with transactional integrity supports composition of complex applications to fully support parallel processing.

Full support for transactional environments

All Within One Transaction (UOW)

1

1. Put Request Q2
5. Get Reply Q1

A

Q1

2

2. Get Request Q2
3. Insert Row DB C
4. Put Reply Q1

C

B

Q2

WebSphere MQ

- Full support for CICS, IMS, WAS, DB2 SPs & Batch RSS
- Reliable Two-Phase commit involving other Resource Managers e.g. DB2

Loosely coupled interaction style

Through correct use of ALIAS queue definitions, the application interaction can be augmented at any point in the flow to enable additional processing without application change.

This example shows a classic use of WebSphere Message Broker – Application D here - which in turn provides rich capabilities for augmentation, routing, transformation etc.

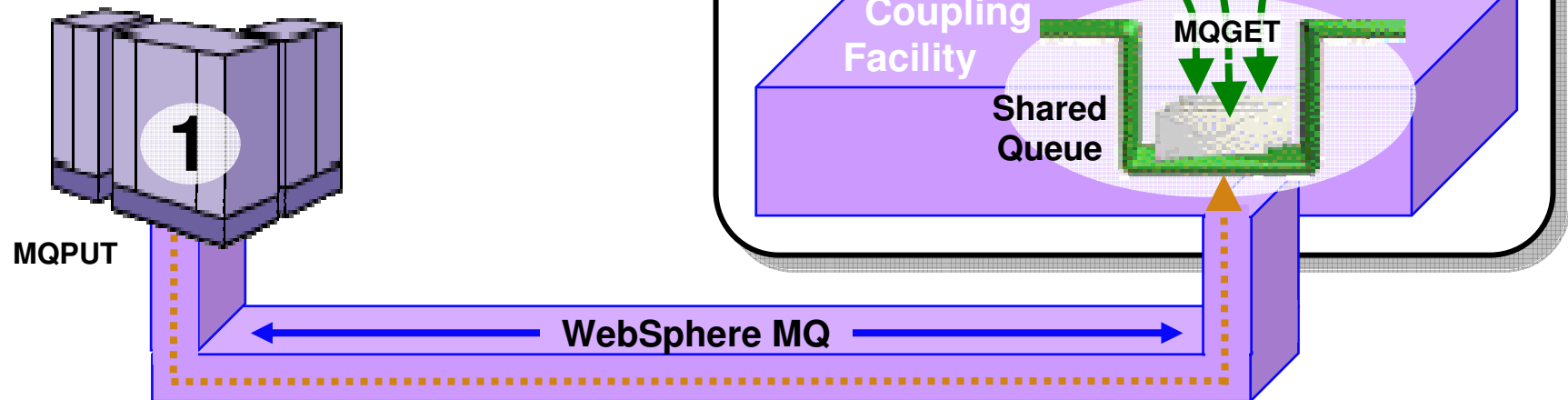All of the above provides flexibility in placement of business logic

D can intercept messages for additional processing without any change to A or B

Request/Reply
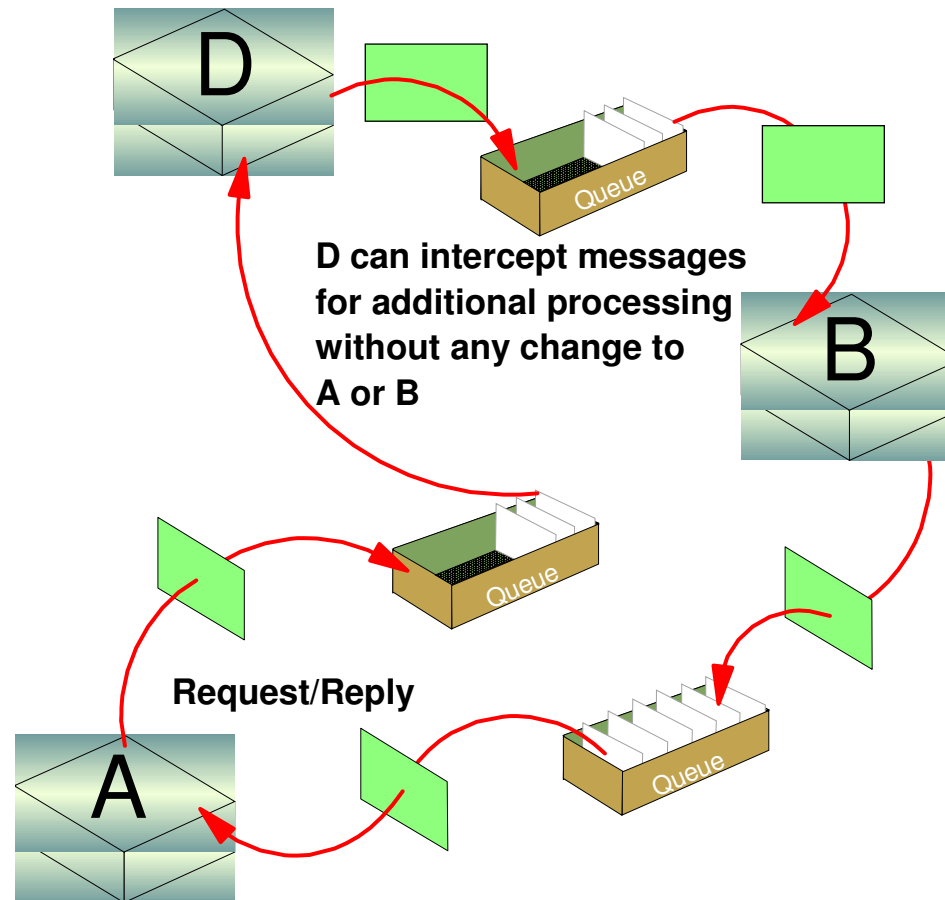
Workload Balancing with Clustering

- **"Round Robin" balancing by default**

- **Custom balancing algorithms can be added as workload exits**

- **MQ V6 provides additional balancing based on assigned priorities & weightings**
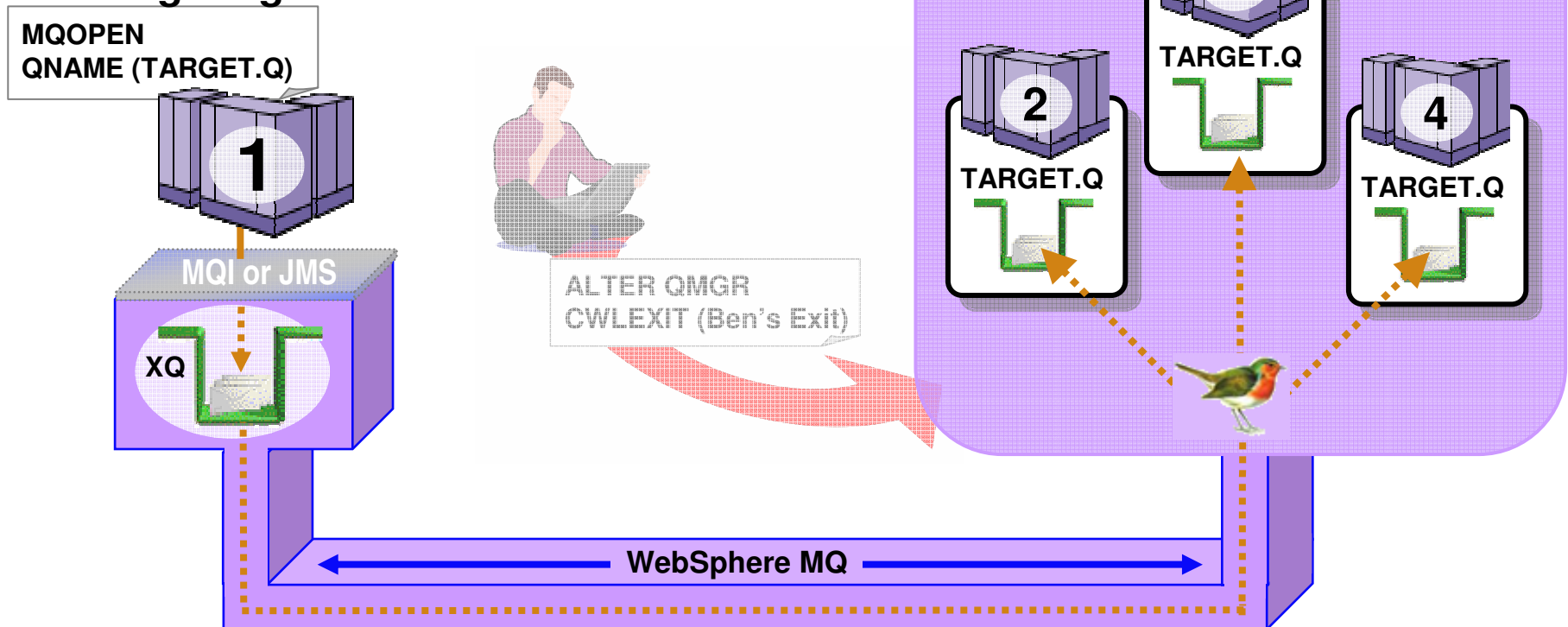
Cluster

MQOPEN
QNAME (TARGET.Q)

1

MQI or JMS

XQ

ALTER QMGR
CWLEXIT (Ben's Exit)

TARGET.Q
3

2
TARGET.Q

4
TARGET.Q

WebSphere MQ

Shared queues on z/OS

- **Exploits Parallel Sysplex**
- **Automatic load balancing**
- **Scalable throughput**
- **Multiple processors can access the same queue**
- **Queue sharing groups**
- **VIPA support**

Parallel Sysplex

2 **3** **4**

Queue Sharing Group

Coupling Facility

MQGET

Shared Queue

1

MQPUT

WebSphere MQ

Highly Available with Failover for z/OS

- **Failure isolation**
  - Automatic peer recovery for failing Servers, Applications or Queue Managers
- **In-flight MQPUTs and MQGETs are rolled back**
  - Since messages are not hosted in any Queue Manager there are no marooned messages
- **24 x 7 availability**
- **Leverages ARM (Automatic Restart Manager)**

Parallel Sysplex

3

Queue Sharing Group

Coupling Facility

MQGET

Shared Queue

1

MQPUT

WebSphere MQ
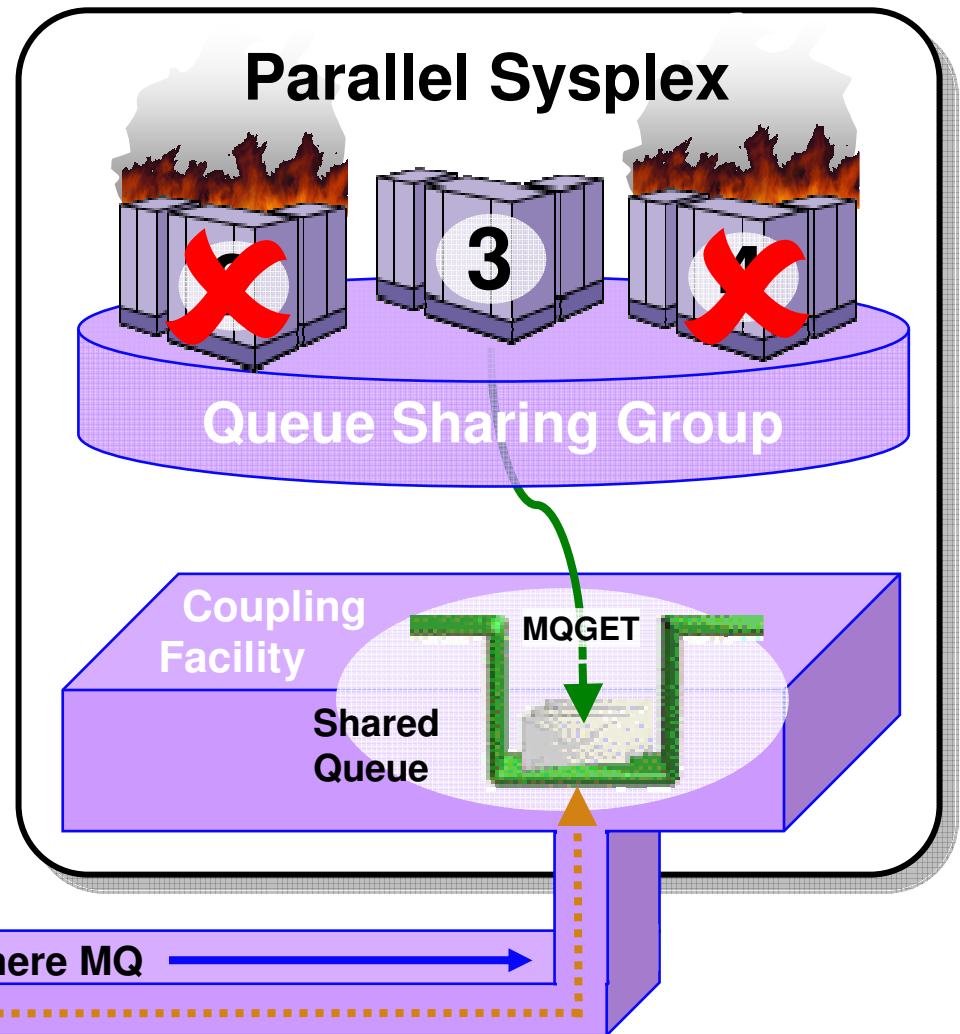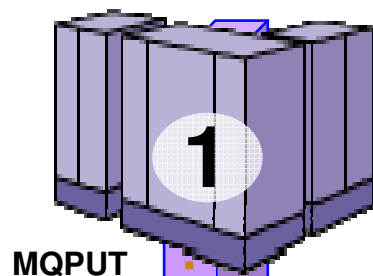
Accessing CICS, IMS, DB2 SPs and Batch/TSO
Explicitly – Using the MQI or JMS API

- **Gives CICS and IMS access to messages from other z/OS regions and other non-z/OS platforms**

- **Allows other applications to access CICS and IMS without using CICS or IMS APIs**

z/OS

| CICS | IMS | DB2 | BATCH |

PUT & GET

PUT & GET
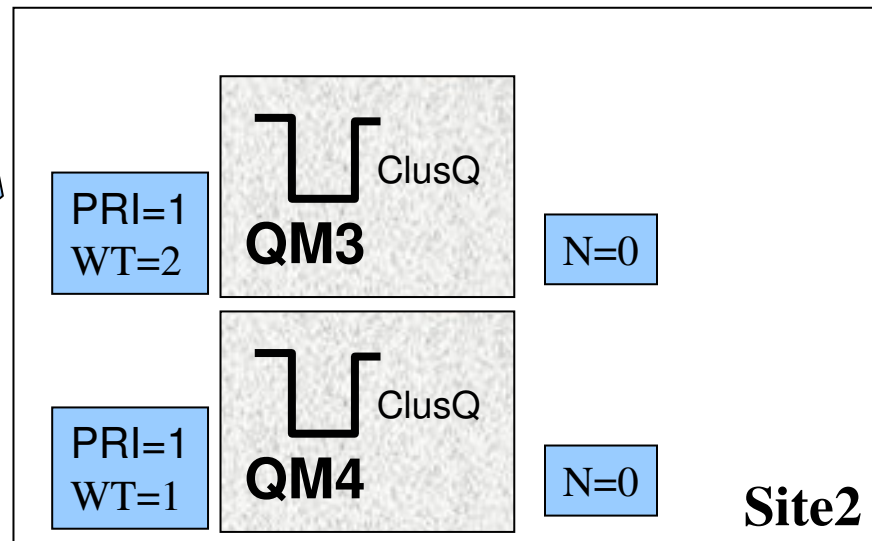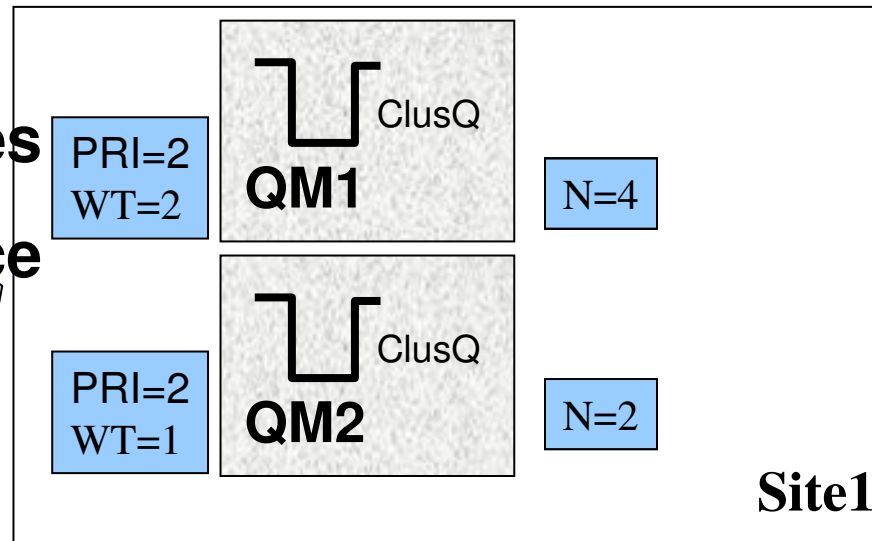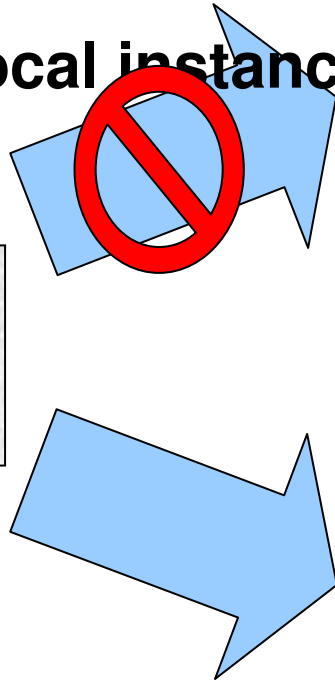
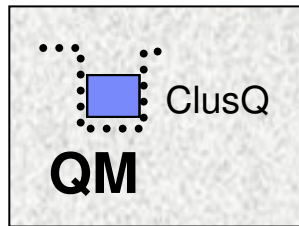**MQI or JMS**

WebSphere MQ
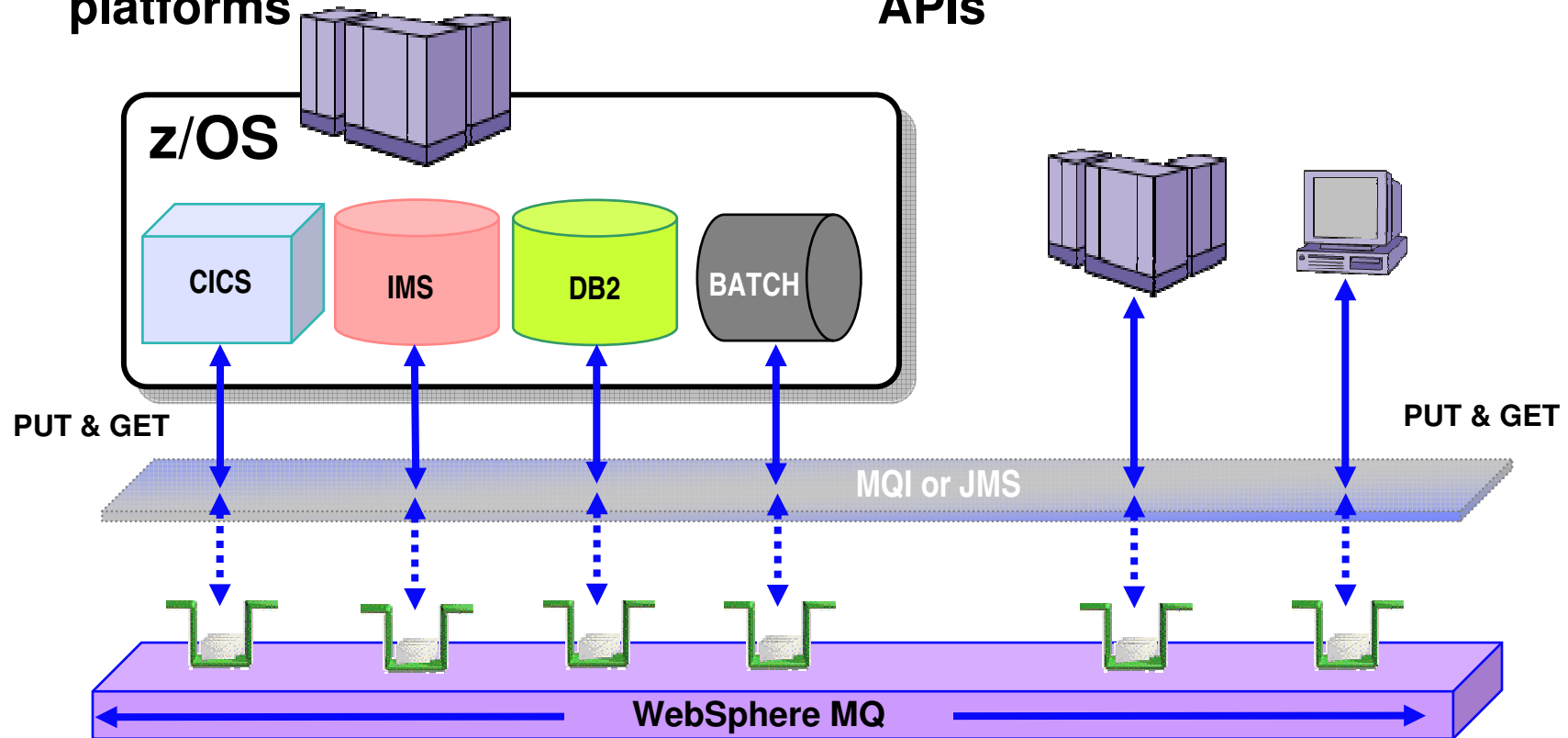
Accessing CICS and IMS
Implicitly – Using the Bridges supplied for CICS and IMS

- **CICS and IMS applications use the native APIs:**
  – EXEC CICS BMS, GU, ISRT...

- **Allows other applications to access CICS and IMS without using CICS or IMS APIs**
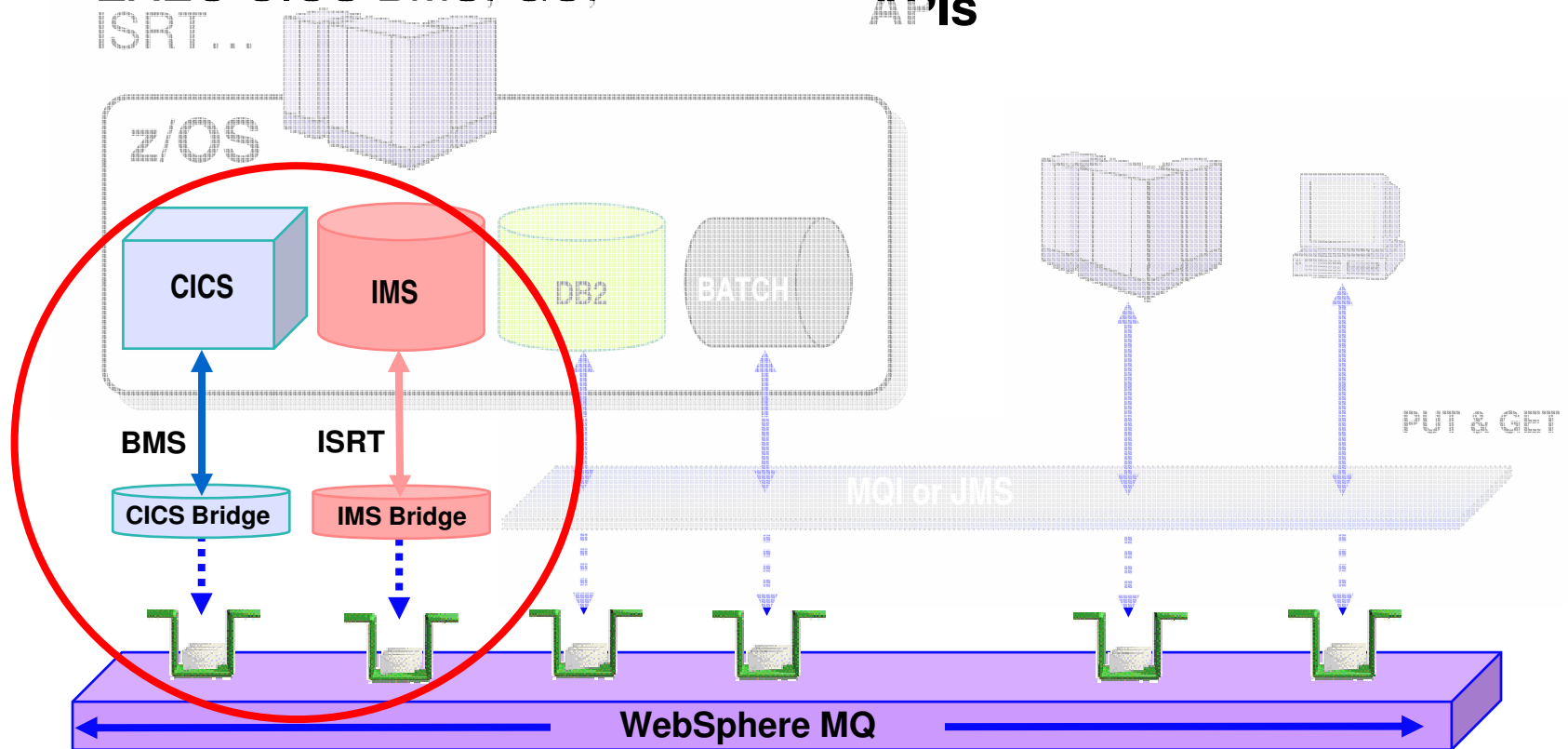
z/OS

| CICS | IMS | DB2 | BATCH |

BMS — CICS Bridge

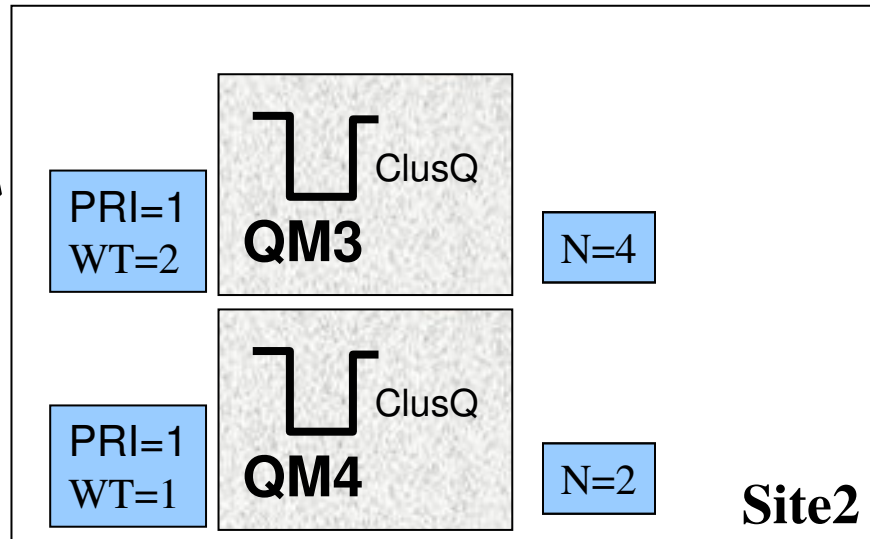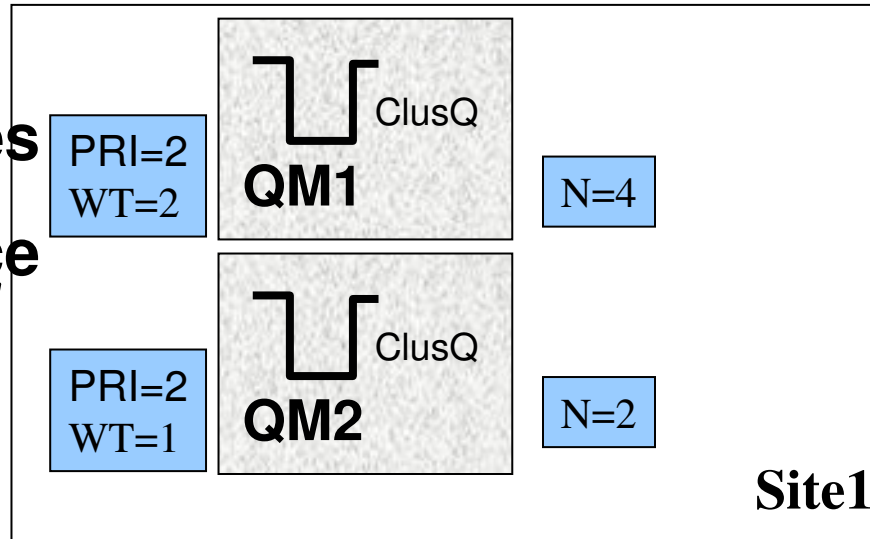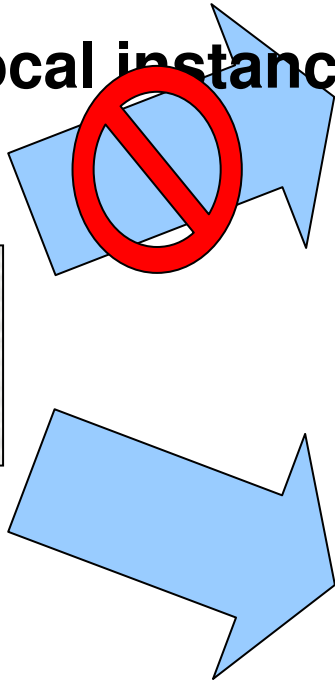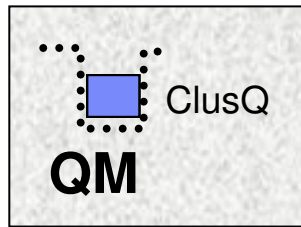ISRT — IMS Bridge

MQI or JMS

PUT & GET

WebSphere MQ

New Cluster Options

- **Weighting & priorities**
- **Include local instance**

ClusQ

QM

| PRI=2 WT=2 | ClusQ **QM1** | N=4 |

| PRI=2 WT=1 | ClusQ **QM2** | N=2 |

Site1

| PRI=1 WT=2 | ClusQ **QM3** | N=0 |

| PRI=1 WT=1 | ClusQ **QM4** | N=0 |

Site2

New Cluster Options

- **Weighting & priorities**
- **Include local instance**

ClusQ
QM

PRI=2
WT=2
QM1
ClusQ
N=4

PRI=2
WT=1
QM2
ClusQ
N=2

Site1

PRI=1
WT=2
QM3
ClusQ
N=4

PRI=1
WT=1
QM4
ClusQ
N=2

Site2

WebSphere MQ for z/OS V6 and recent SupportPac highlights

- **The details of z/OS availability**

z/OS Availability – Large shared queue messages

- **Removes the 63KB limit for shared queues so removing a significant constraint for WMQ application exploitation of shared queues**
  - Maximum shared queue message length now 100MB
    - same as that for non-shared messages
- **Uses DB2 to store large message payloads as BLOBs**
  - Header stored as V5.3 in CF with pointer to BLOB
  - Transparent to applications
  - Small messages stored as in V5.3, without DB2
- **There are performance implications ...**
  - Expect degradation (indications are >4x for large persistent messages)
  - All customer feedback indicates that only a small percentage of applications that use shared queues will have messages > 63KB so the performance cost is believed to be acceptable
- **Administrator control of whether it is permitted**

z/OS Availability – SQ Admin. Structure recovery

- **The Shared Queue Administration structure holds key WMQ control information across the queue-sharing group**
  - Transaction state information to enable peer recovery
  - Serialised application support information
  - Structure interest information
- **But with WMQ V5.3 a loss of this structure will cause each connected Queue Manager to fail resulting in a total loss of the queue-sharing group**
- **In WMQ V6 this limitation is removed**
  - Queue Managers will reconnect to/reallocate the structure
  - This is done automatically - no operator intervention required
  - Structure will be rebuilt from log data
  - Until the rebuild is complete some shared queue functions will not be available
    - no new shared channels can start
    - no Application structure rebuild is possible
    - no Queue Manager peer recovery is possible
  - BUT .. Existing applications will carry on running to completion and the vast majority of new applications can start and also run to completion

z/OS Availability – SQ Admin V5.3.1 v V6
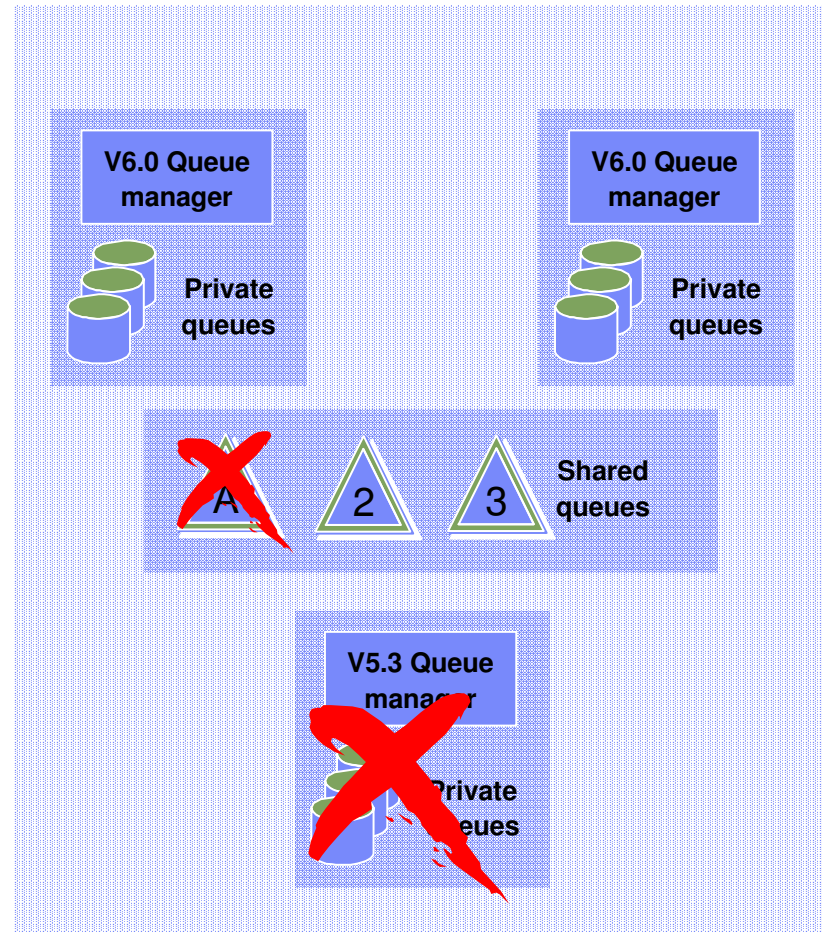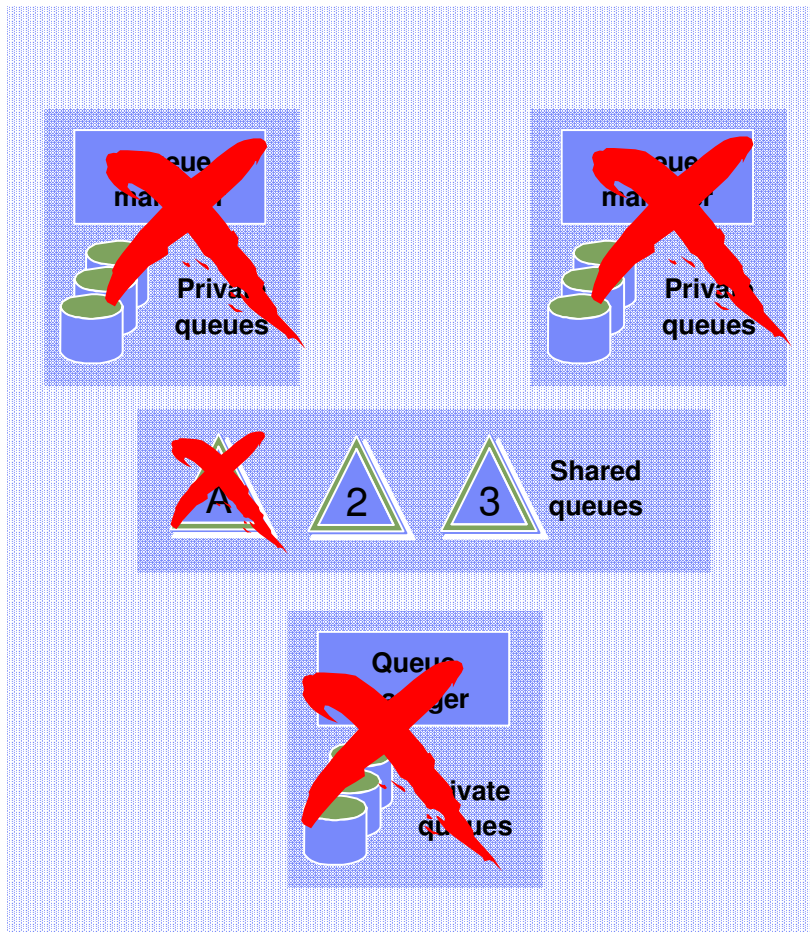
z/OS Availability – 64GB Page sets

- **In order to have page sets that exceed 4GB -**
  - NEW PAGE SETS MUST BE CREATED

- **Page sets MUST be DEFINED with an SMS DATA CLASS that specifies –**
  - "EXTENDED ADDRESSABILITY"

- **Create replacement page sets with the SAME SIZE as the page set that they replace, and REPRO old page sets to the new page sets - these page sets will 'grow' beyond 4GB by page set expansion**

- **Or brand new page sets can be allocated > 4GB and formatted ready for use**

- **Update QMGR JCL to point to new page sets**

- **Note: Existing page sets can continue to be used, but they will continue to have a maximum size of 4GB**

z/OS Availability – Dynamic CHIN parameter setting

- **CSQXPARM parameters are now attributes of the Queue Manager object**
  - Supports common methodology for "Queue Manager" parameter changes between z/OS and Distributed Websphere MQ

- **So with WMQ V6 no need for CHIN recycle**

- **Note: Some specific attributes may still require a recycle, determined on a case by case basis.**

- **Example –**
  - ALTER QMGR   ADOPTCHK(ALL)
    ADOPTINT(120)
    ADOPTMCA(NO)
    TCPKEEP(YES)
    RCVTIME(X2)
    RCVTMIN(120)

# z/OS Availability – Multiple TCP/IP stack support

- **In some recovery scenarios WMQ may be running on a different LPAR with a different TCP/IP stack**

- **In these scenarios need to be able to specify both a Listener stack and the stack that an Outbound Channel should use**

- **New CHIN parameter TCPSTACK -**

  - TCPSTACK=SINGLE

    - Restricts channels and listeners to the stack named in TCPNAME
    - CHIN will behave the same as with WMQ V5.3

  - TCPSTACK=MULTIPLE

    - Channels and listeners may bind to any available stack
    - TCPNAME stack will be the default

- **Listener IPADDR and channel LOCLADDR used as with WMQ 5.3 to indicate communications address within the stack**

z/OS Availability – V5.3 CICS Bridge Monitor

- **The CICS Bridge provides MQ interface to legacy CICS applications**

- **With WMQ 5.3 it comes in two flavours -**

  - DPL Bridge - Link to program with data in Commarea

  - 3270 Bridge - Transaction with BMS map

    - Uses CICS Attach bridge facility

- **Bridge monitor browses queue, performs authority checks and starts a transaction to process the message.**

- **BUT .. Significant restrictions**

  - Only one bridge monitor per bridge queue

  - 3270 transactions cannot be routed to AORs

  - Reduces reliability and scalability because tied to single region

- **Some enhancements introduced in MQ 5.3.1**

  - Reply expiration, null truncation, sync on return

z/OS Availability – CICS Bridge with WMQ V5.3

CICSA

DPL Prog

Bridge monitor
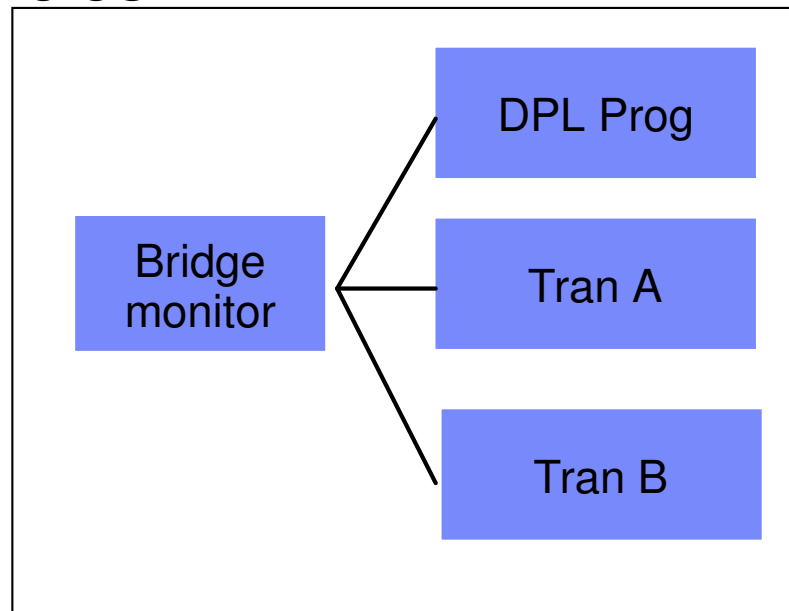
Tran A

Tran B

Each instance of the Bridge must have exclusive access to the Bridge Queue.

Also the form of the EXEC CICS START does NOT allow transaction routing so TRAN A and TRAN B can only run in the CICSA region.

WebSphere MQ Bridge for HTTP

Before   `MQI,JMS,XMS`

Enterprise Applications

MQ Client   `MQPUT,MQGET`   MQ Server   **MQ backbone**

- Applications can only connect to the MQ backbone using an MQ Client (or local Server)
- MQ Clients are free BUT their deployment can create an inhibitor to MQ adoption
  - *Clients need to be installed and configured – a potential provisioning challenge for large numbers*
  - *They have a small footprint (Approx. 3-10 MB) – but this could still pose problems for some applications / users*

After   `http://mq/queue`

J2EE App Server   `MQPUT MQGET`

Enterprise Applications

Web 2.0 AJAX REST   HTTP App   `POST,GET`   HTTP stack   MQ Bridge for HTTP   MQ Server   **MQ backbone**

- Provides a bridge between HTTP applications and WebSphere MQ
  - *Maps messages from HTTP URLs to MQ Queues and MQ Topics (provides Pub/Sub)*
- Uses the HTTP/HTTPS stack on J2EE App Servers – connects via JCA
  - *Works with any J2EE Server with JCA support including WAS Community Edition*

Key Benefits

- Zero client solution for connecting to an MQ network – eliminates need for MQ client for simple access
- Easy way for Web developers that create Web 2.0, AJAX, REST Web services to connect to MQ
  - e.g. Web pages can dynamically update themselves on-the-fly using data from the MQ backbone

z/OS Availability – V6 CICS Bridge Monitor enhancements

- **Multiple bridge monitors for single queue**

- **Transaction Routing**

- **Tolerate & return additional headers (e.g. RFH2)**

- **Honour Backout threshold and requeue queue name**

- **Error messages to CICS joblog option**

- **Optional reply to queue**

- **Enhanced passticket support**

- **Honour MQRO_DISCARD_MSG and MQRO_PASS_DISCARD_AND_EXPIRY**