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
- DB2
- E-business logo
- ESCON
- eServer
- FICON
- IBM
- Smart SOA
- IBM Logo
- IMS
- iSeries
- MVS
- OS/390
- pSeries
- Rational
- RS/6000
- S/390
- Tivoli
- VM/ESA
- VSE/ESA
- WebSphere
- z/OS
- zSeries
- 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 and recent SupportPac highlights
- WebSphere MQ and the Web
- Summary

WebSphere MQ – What and Why?

Transport for both service-oriented and non-service-oriented IT assets

WebSphere MQ moves:

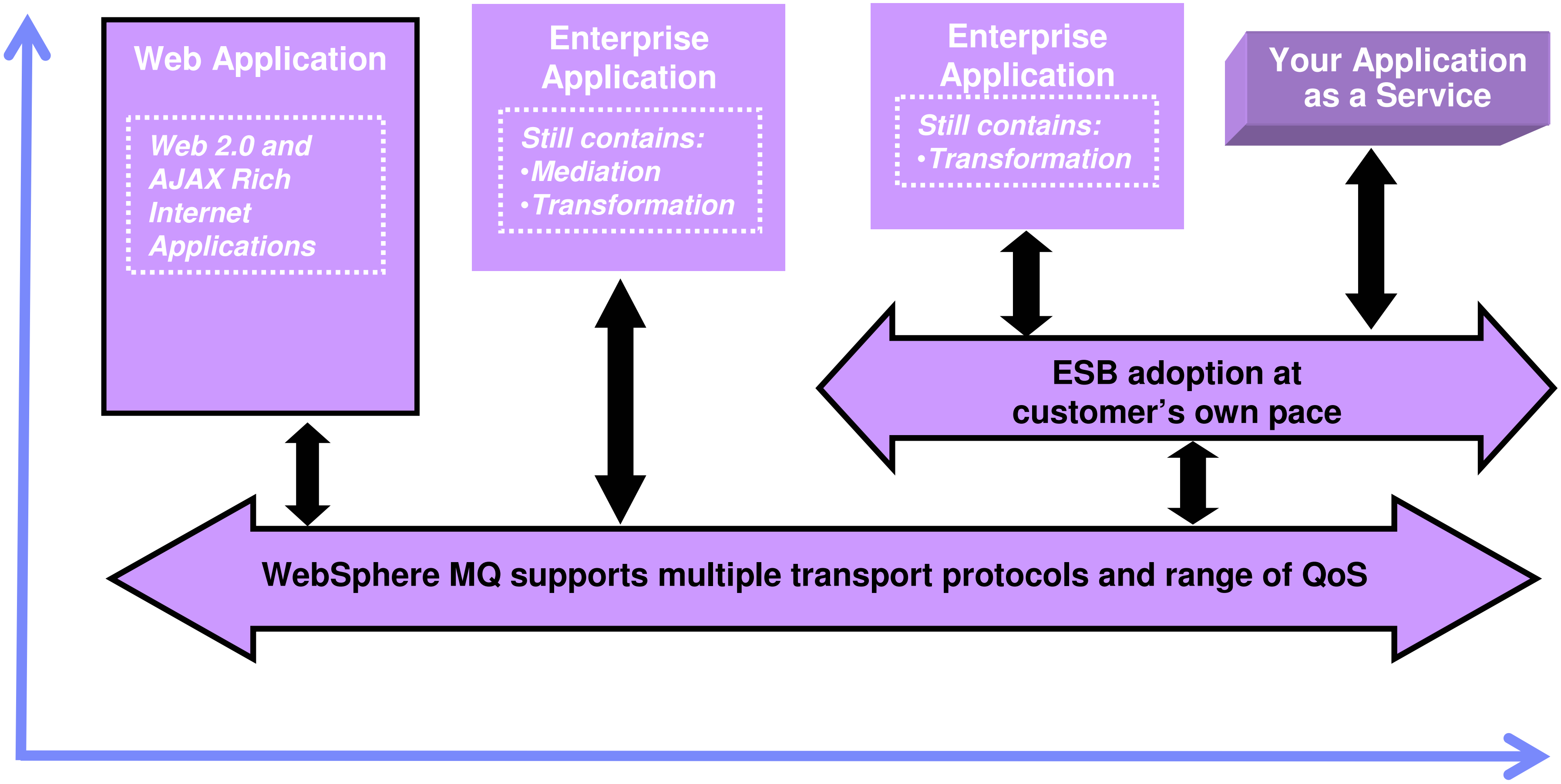
- *Data*
- *Messages*
- *Events*
- *Files*
- *Web service requests / responses*

7 Keys to SOA Connectivity –

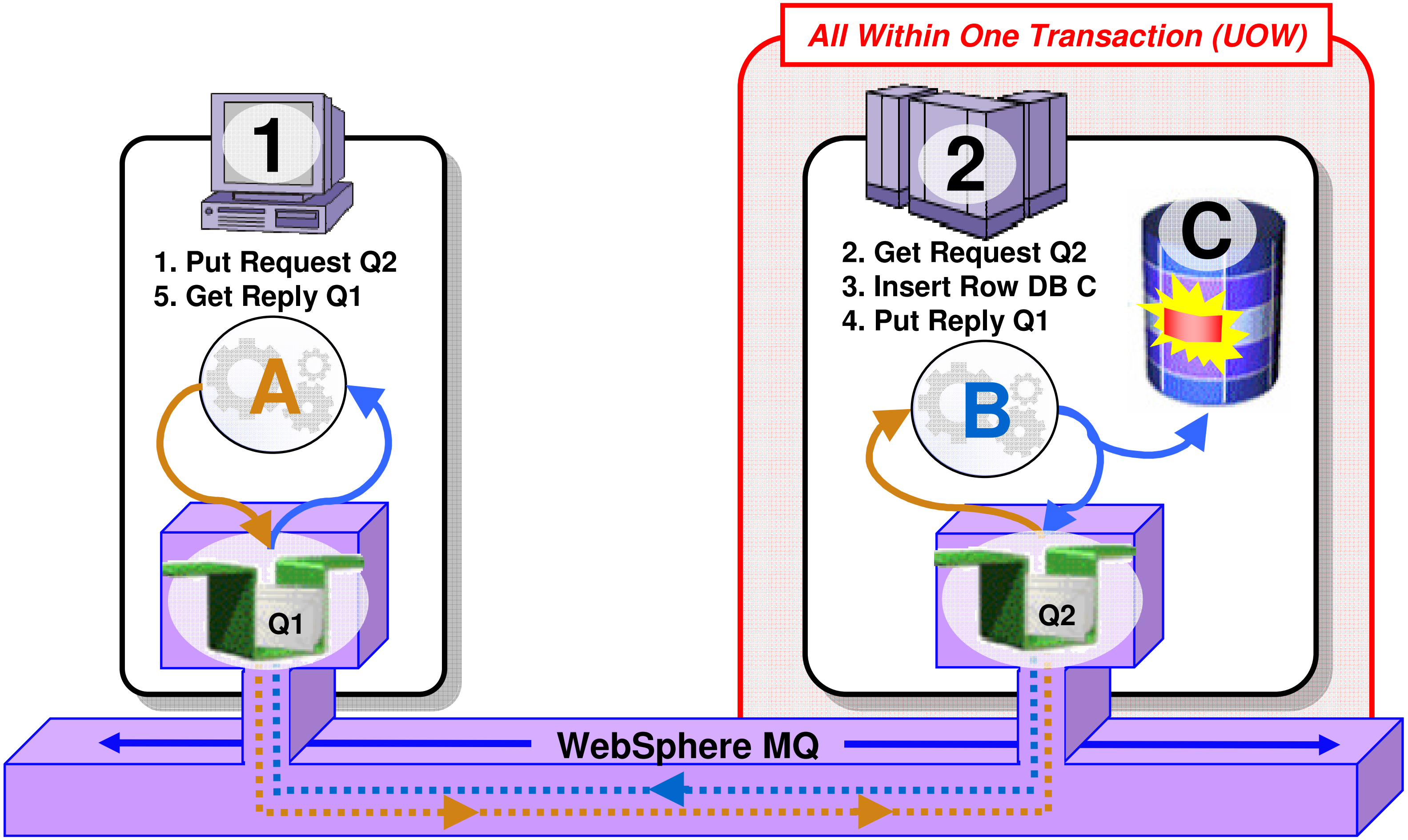
Reliable – proven, trusted, the standard
Secure – protects data end-to-end
Time-flexible & Resilient – connect whenever
Transactional – preserves integrity of data
Incremental – grows with your needs
Ubiquitous – connect virtually anything
Basis for ESB – underpins and extends`

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

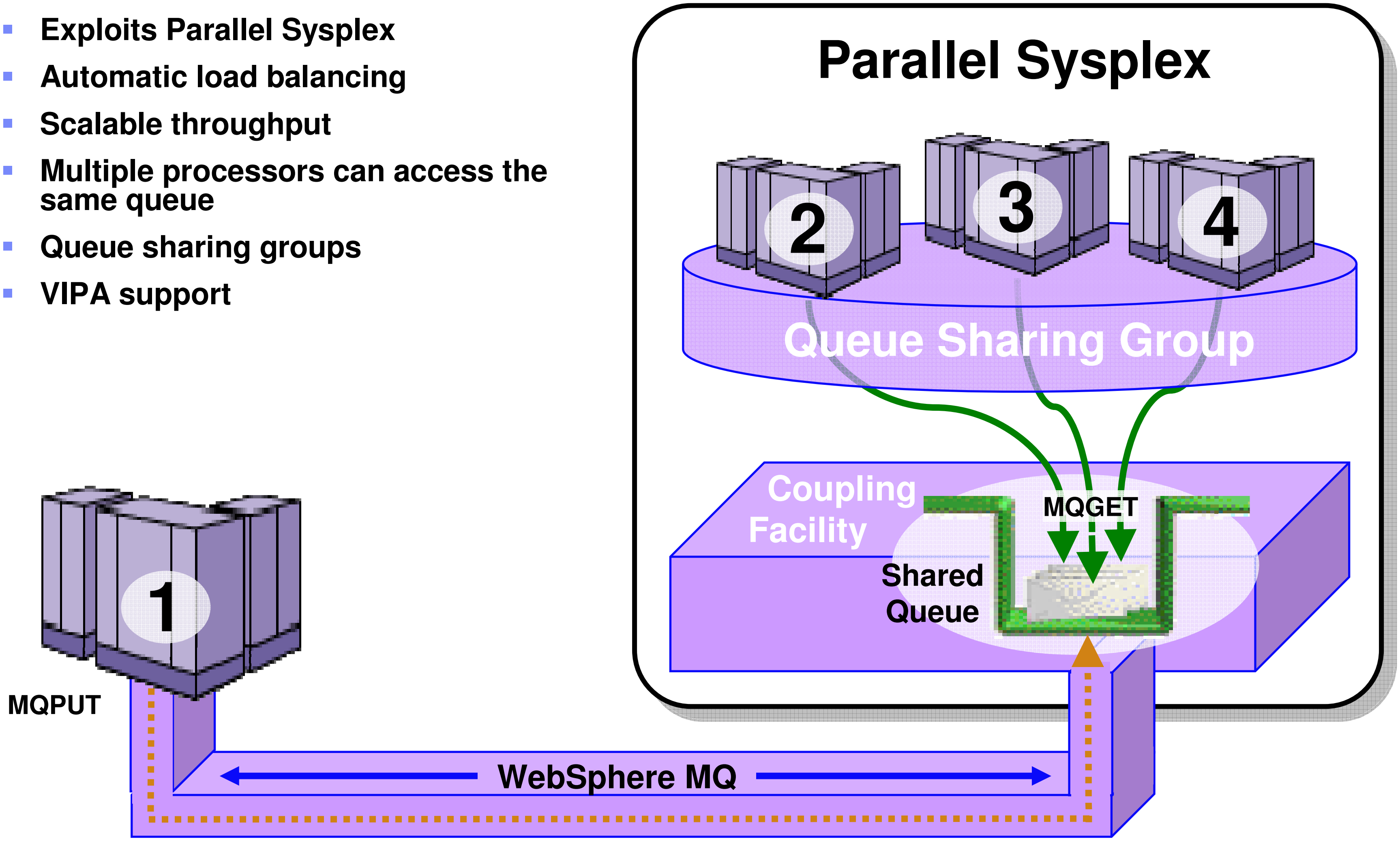
Full support for transactional environments



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

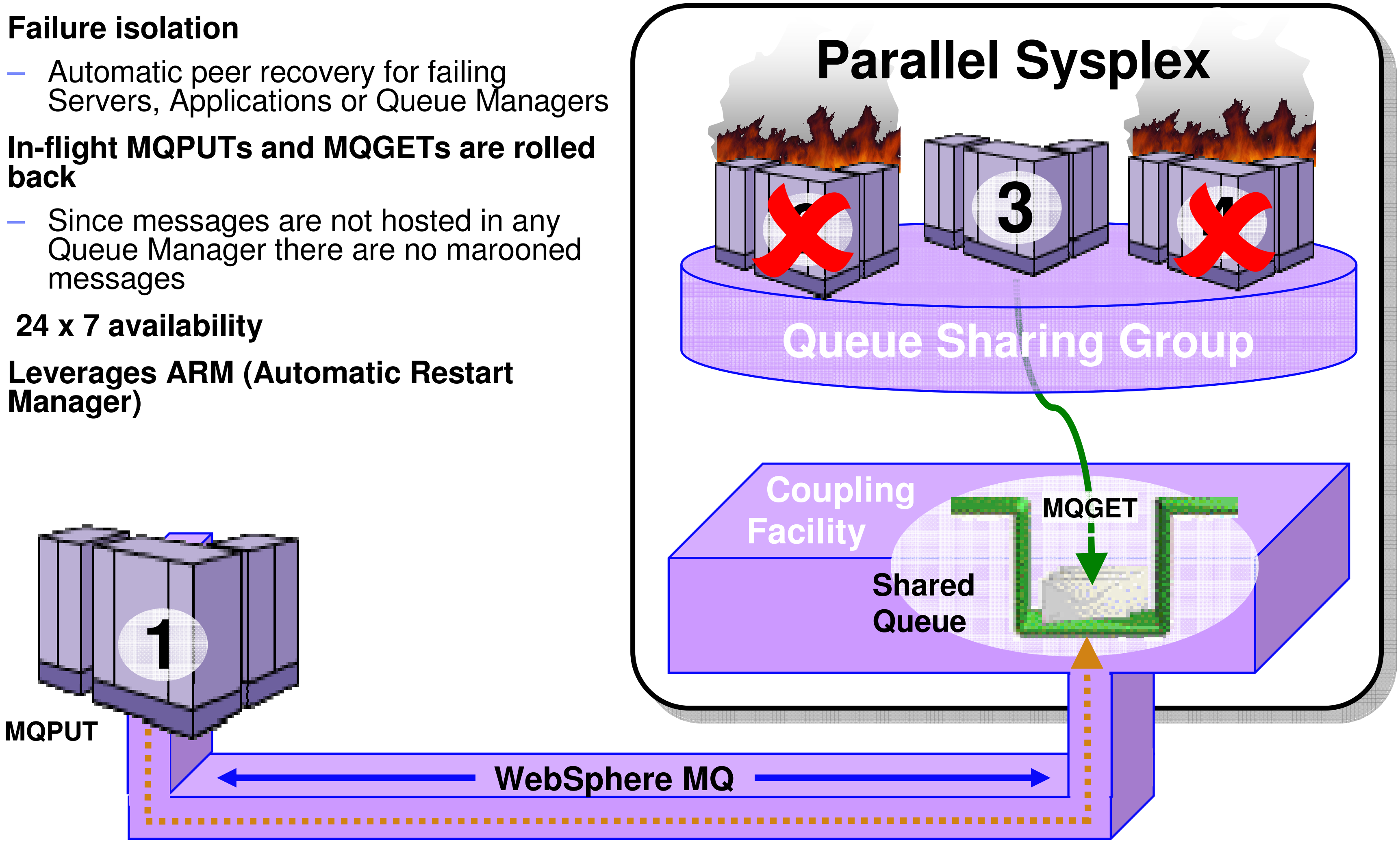
Shared queues on z/OS

- 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)**



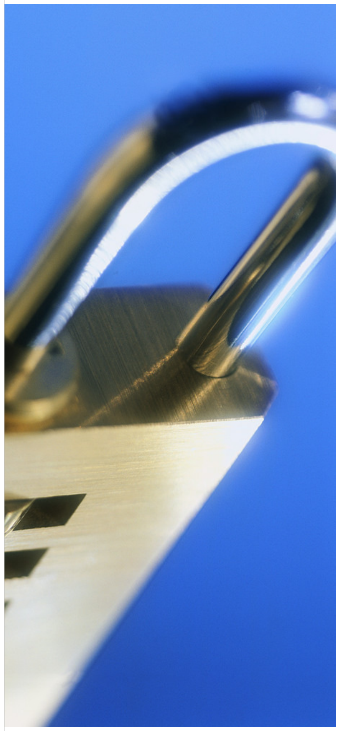
Security

- **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



WebSphere MQ for z/OS V6 and recent SupportPac highlights

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
  - Making it easier to use full JMS function without downloading or buying extras

XMS

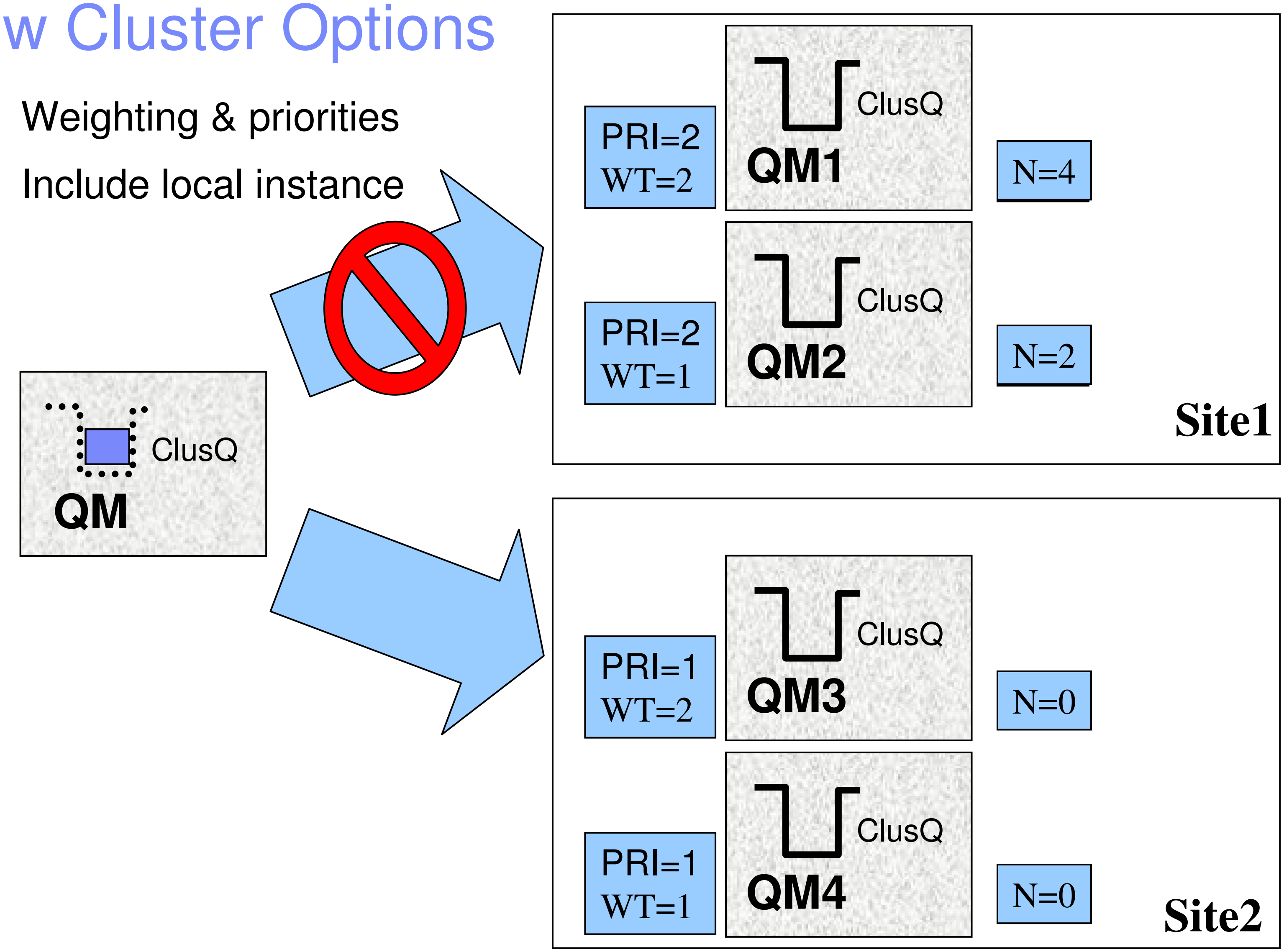
- Common messaging API across IBM providers
  - Transport neutral - WMQ, Real-Time, and WAS V6 Messaging
  - JMS-like API for other languages
  - Simplified Point-to-Point and Publish/Subscribe messaging
- Decoupled administration
  - XMS and JMS use the same administered Connection Factories and Destinations
- 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

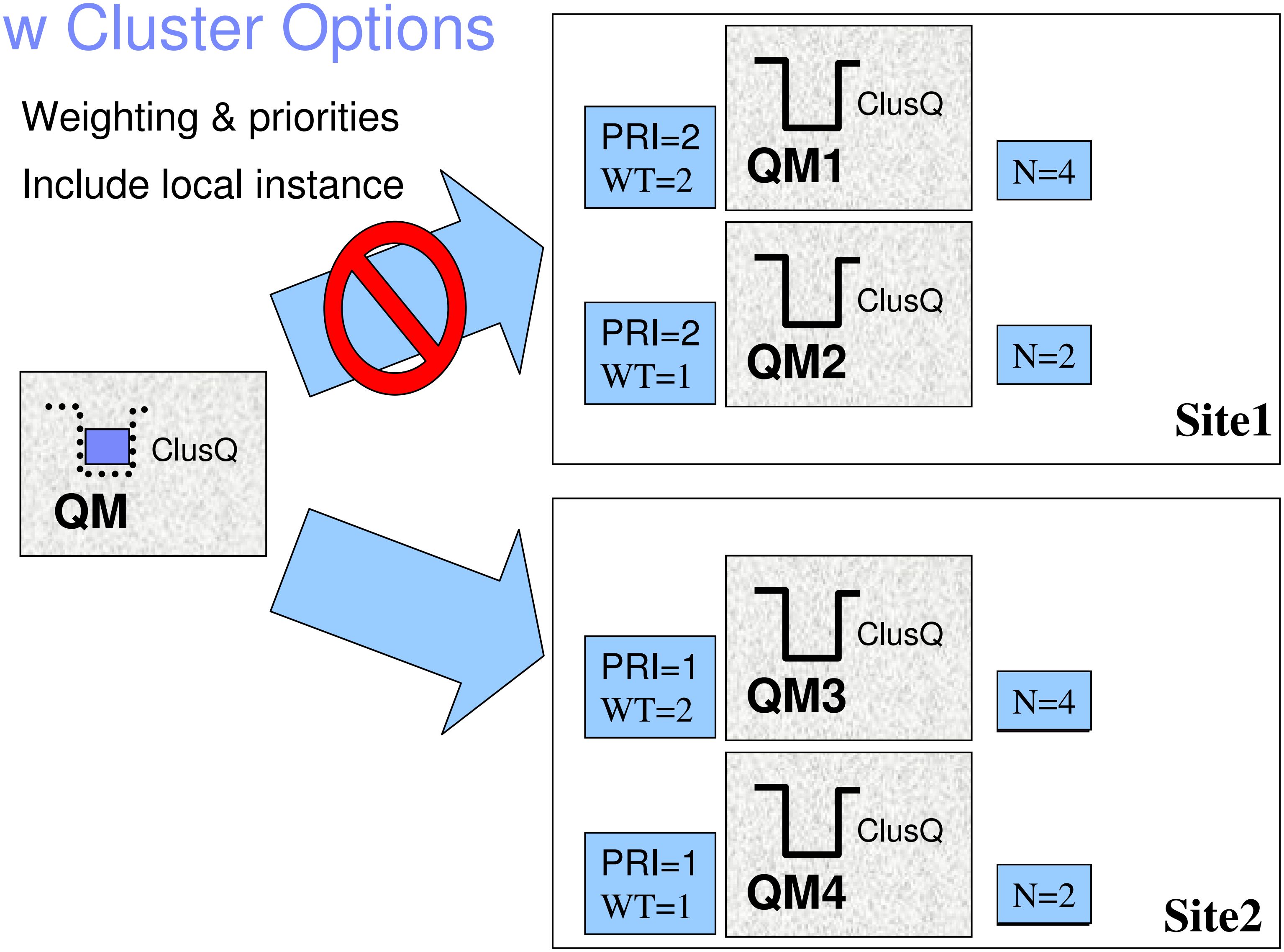
New Cluster Options

- Weighting & priorities
- Include local instance



New Cluster Options

- Weighting & priorities
- Include local instance



Improved Monitoring

- Who is using the Qmgr
  - Already have DISPLAY QSTATUS to show who is using a queue
  - New command DISPLAY CONN to show who is connected and which queues they have opened
  - New command STOP CONN to force a disconnect (not on z/OS)
  - Also shows UOW timestamps
- Additional status information
  - Problem determination (and problem avoidance) indicators
    - Show "instantaneous" information about activity
  - DISPLAY CHSTATUS
    - Cluster transmission queue sub depth and “average” throughput times
    - “Average” batch size
    - "Substate" - sending data on wire, waiting for nameserver response, in an exit
    - Response time from remote end
  - DISPLAY QSTATUS
    - “Average” time messages lived on queue (short & long-term indicators)
    - Last GET/PUT times
    - Oldest message still on queue
  - Adds new fields to the STATUS display for Channels
  - Adds new fields to the QSTATUS display for local queues

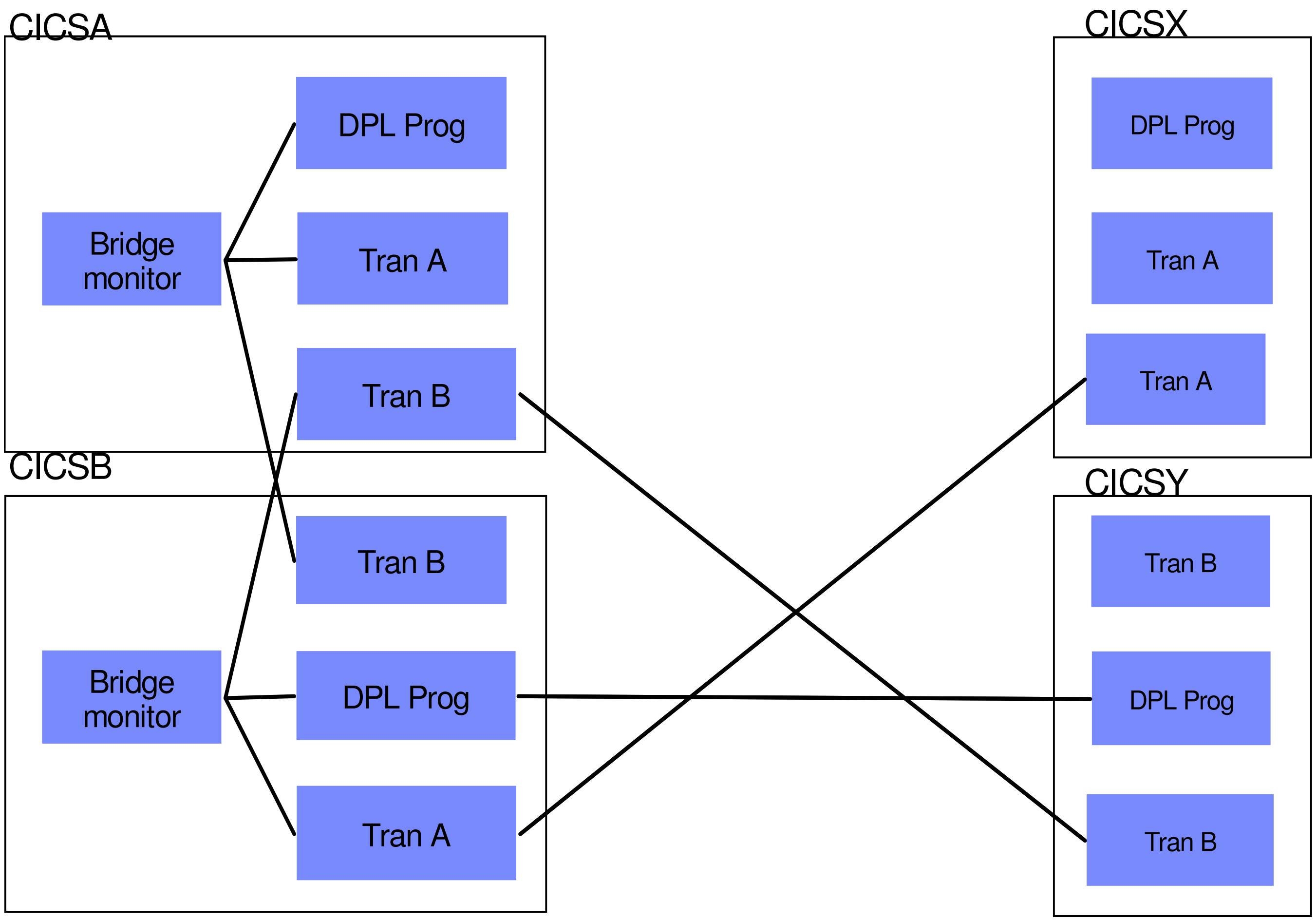
Improved Management for z/OS – PCF support

- Making it easier to write administration and monitoring programs
  - Same programs can drive z/OS and Distributed platforms
- It has always been comparatively easy to convert a PCF command into MQSC
- The difficult bit has been parsing the responses
  - PCF is structured, self-describing format
  - MQSC is pseudo-English, random ordering
- z/OS Command Server is now able to handle both formats
- PCF structures extended to handle the multiple sets of responses from Queue Sharing Groups
  - Other extensions to the “language” for new structures
  - E.g. 64-bit, name-value pairs
- All appropriate z/OS commands get PCF definitions
  - There are still a few that only make sense for local consoles
  - E.g. START/STOP TRACE

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

z/OS Availability – CICS Bridge with WMQ V6.0



Using the multiple bridge monitor and routing support allows the equivalent of the TOR - AOR set up.

Multiple Bridge Owning Regions (BORs) can take messages from a shared queue and feed work to a number of AORs thereby providing scalability and eliminating single points of failure.

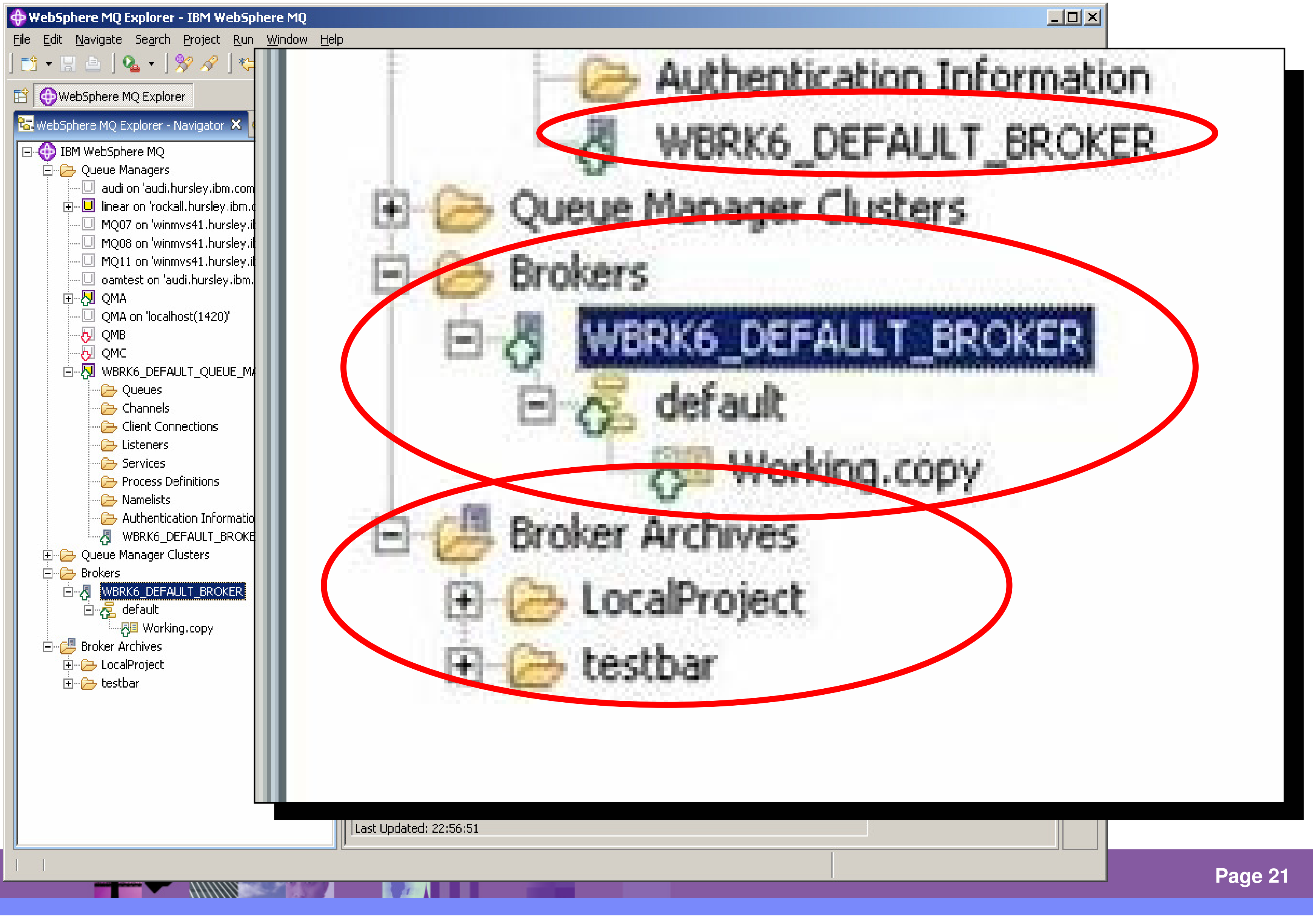
Improved support in CICS TS V3.2

- The following WMQ components that connect CICS and WMQ have been transferred and are now integrated in CICS TS V3.2 -
  - CICS-MQ Adapter
  - CICS-MQ Trigger Monitor
  - CICS-MQ Bridge
- As part of this transfer the following enhancements have been made -
  - Exploitation of CICS OTE
    - The above components have been made threadsafe and can exploit CICS open TCBs which for scenarios where multiple MQ calls are being made in a CICS transaction will result in reduced TCB switching and thus reduced CPU and pathlength
  - Improved diagnostics by using CICS facilities for trace, messages, etc.
  - Improved statistics which are now available within CICSplex SM Web UI
- See CICS TS V3.2 Announce Letter for more information

z/OS Availability – SUSPEND/RESUME IMS Bridge

- The problem
  - how to quiesce IMS bridge with a shared request queue
  - MQPUT inhibit of request queue stops all bridge instances
  - STOP QMGR leaves replies trapped in IMS or in bridge
- The solution
  - SUSPEND QMGR BRIDGE(IMS)
    - stops MQGETs by bridge for all bridge queues
    - allows MQPUT by bridge of replies from IMS
  - RESUME QMGR BRIDGE(IMS)
    - starts MQGETs by bridge for all active bridge queues
- Scenario to quiesce QMGR
  - issue SUSPEND QMGR BRIDGE(IMS)
  - wait until replies complete
  - no relevant IMS transactions running
  - MQ XMIT queues to remote QMGRs empty
  - issue STOP QMGR MODE(QUIESCE)

IS02 Supportpac: Message Broker Admin Plug-in



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

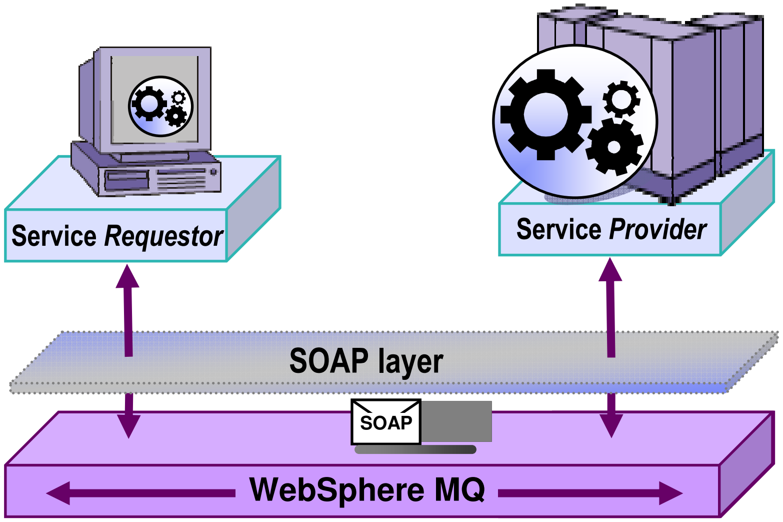
- WebSphere MQ can be used as a Web 2.0 enabler with the HTTP Bridge
  - Extending the reach of Enterprise applications to the Web

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 layer of messaging services to help make SOAP exchanges:

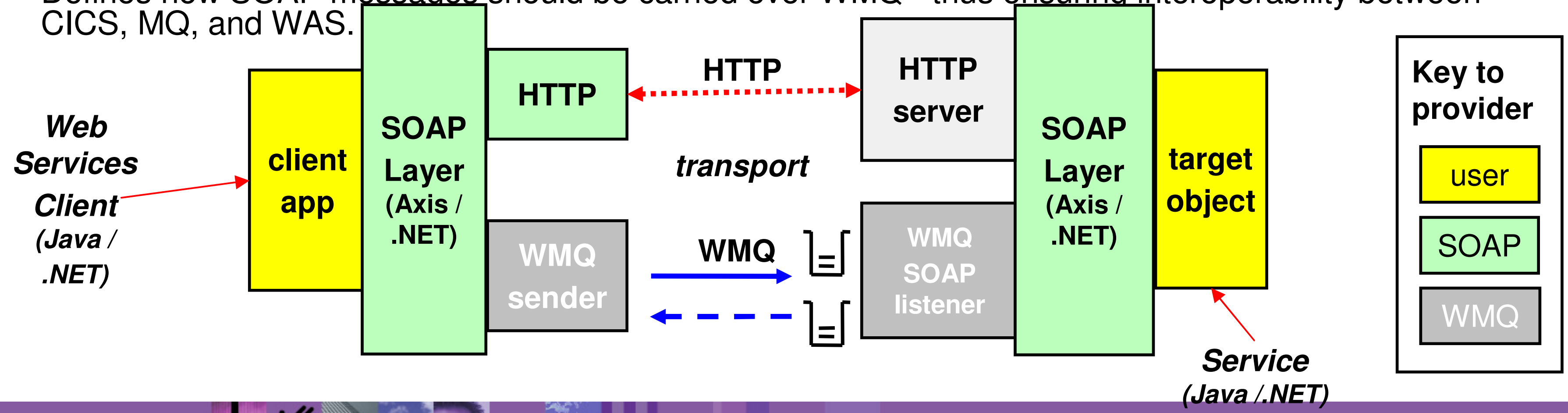
- ✓ *More reliable than those sent over HTTP*
- ✓ *Transactional – allowing service resources to be updated atomically so they retain integrity and failures can be recovered and retried*
- ✓ *Arrive in the same order as they were sent*
- ✓ *Decoupled in time – so that they can handle other requests whilst waiting for responses*
- ✓ *Buffered so that SOAP requests can be throttled for batch-style Web services*
- ✓ *Clustered so that SOAP requests can be processed in parallel by multiple service providers*
- ✓ *Easily integrated with applications that are not enabled for SOA*
- ✓ *More auditable*



- 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.



WebSphere MQ v6.0.2 Service Definition SupportPac

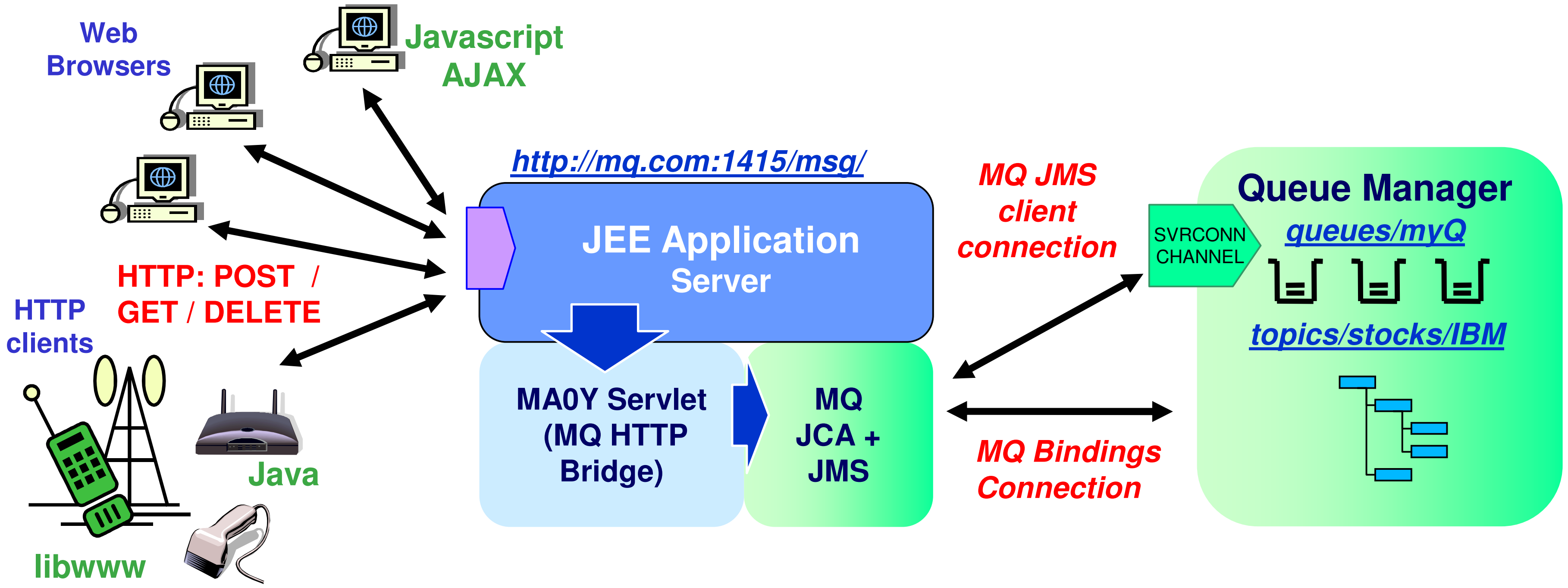
- **Available now as Cat 2 supportpac MA93**
- **Provides a Service Definition to kick-start your SOA**
  - Service Definition enables applications connected with WebSphere MQ to appear as Services
- **Promotes application re-use**
  - Enables applications connected to WebSphere MQ to be composed and re-used as Services
    - Directly (assuming client –side proxy generation capability)
    - Indirectly (by bridging from HTTP via ESB)
      - Redirect incoming requests from an HTTP URI to a WMQ endpoint which understands the WMQ application MEP, and trans-codes data from SOAP/HTTP binding to native message / MQ
- **Improves visibility and governance**
  - Enables applications connected by WebSphere MQ to be catalogued as Services
    - WSRR V6.1 supports generic WSDL extensions and additionally an MQ Endpoint
  - Aids auditing and change management to help assess the impact of changes to MQ applications
- **Encourages broader choice of management and monitoring**
  - Service definition is published for use by ISVs and customers to identify, manage and configure MQ applications as Services
  - Exploited by WebSphere Service Registry and Repository for governance of existing MQ-connected applications

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

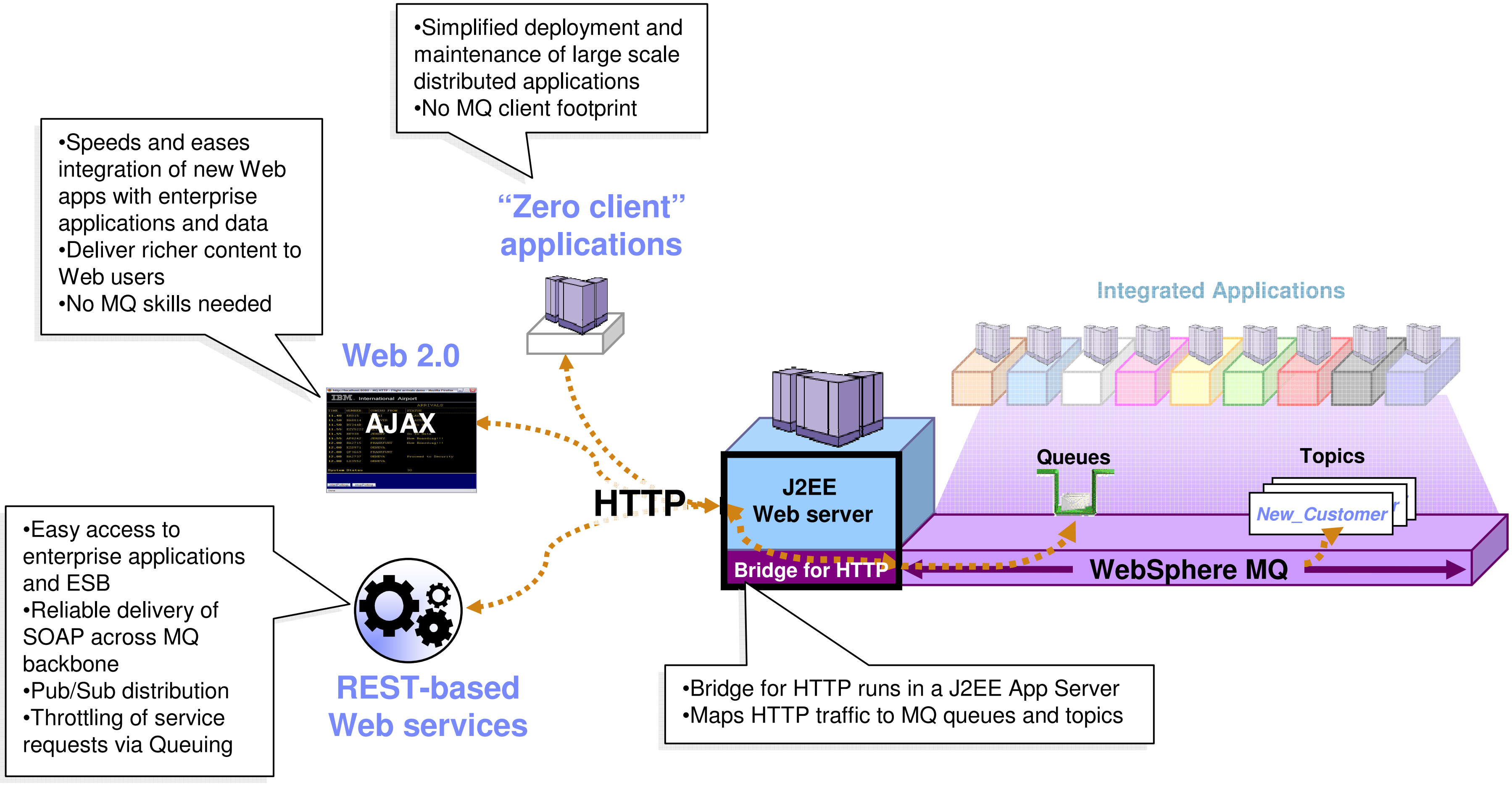
- **Enable MQ Application Connectivity from *any* Platform or Language with HTTP capabilities**
  - Significantly increase range of supported platforms
    - e.g. – Linux distros, POS terminal running Windows Services for Unix environment, RFID reader, Mobile devices
  - Reduce cost of MQ Client deployment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

Architectural Overview



- Key features of the WebSphere MQ Bridge for HTTP -
  - Maps URIs to queues and topics
  - Enables MQPUT and MQGET from
    - Web Browser
    - Lightweight client

Scenarios – WebSphere MQ Bridge for HTTP



- Simplified deployment and maintenance of large scale distributed applications
- No MQ client footprint

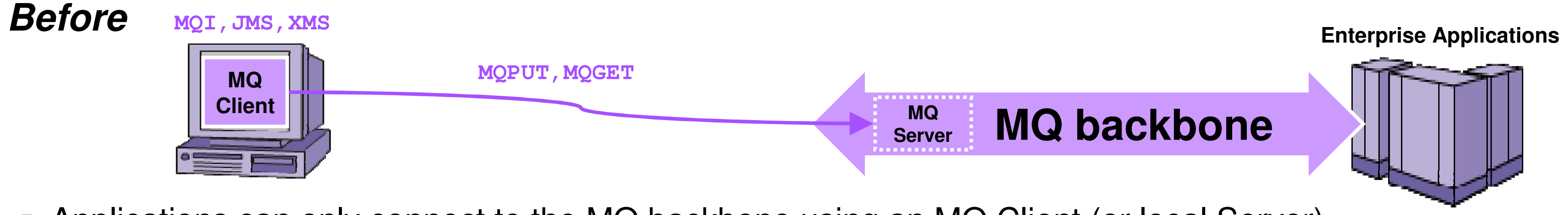
- Speeds and eases integration of new Web apps with enterprise applications and data
- Deliver richer content to Web users
- No MQ skills needed

- Easy access to enterprise applications and ESB
- Reliable delivery of SOAP across MQ backbone
- Pub/Sub distribution
- Throttling of service requests via Queuing

- Bridge for HTTP runs in a J2EE App Server
- Maps HTTP traffic to MQ queues and topics

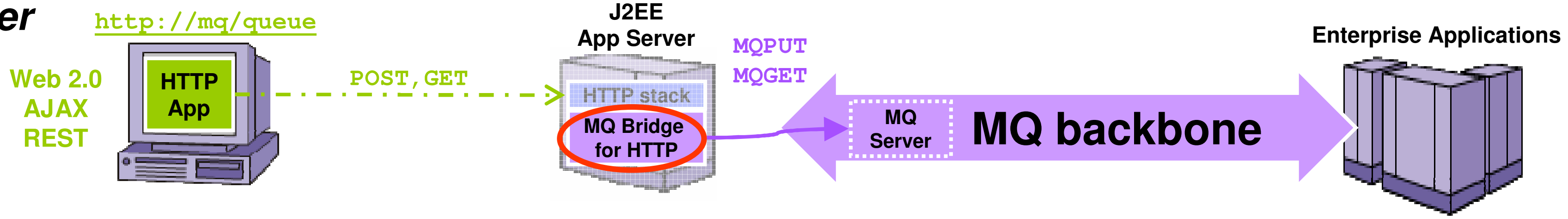
WebSphere MQ Bridge for HTTP

Before



- 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



- 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

Summary – WebSphere MQ Bridge for HTTP

Benefits

- No client libraries required – use stand-alone or in a Browser
- Simple API mapping HTTP to MQ
  - But highly extensible API
- Low cost of entry
  - develop in notepad, ad-hoc service descriptions / message formats

Limitations

- HTTP (as opposed to MQ) reliability
- No transactional capabilities
  - POST/DELETE operations equivalent to MQPUT/MQGET out of syncpoint
- Subset of MQI

Delivery

- MA0Y Cat2 SupportPac now available
  - First release is limited function - not suitable for production use
  - **THIS IS AN API PROTOTYPE – WE NEED FEEDBACK ON IT!!**
- Forthcoming supportpac (MA94) to provide Native HTTP Listener function
  - **Also Cat2 to solicit feedback**
- Intent is to ship both as Cat 3 with WMQ V7

WMQ and the Web - summary

- **WMQ can be used as a Web 2.0 enabler**
  - Extending the reach of Enterprise applications to the Web

- **The HTTP-WMQ bridge can be used for “Zero Footprint Clients”**
  - Server, Client , and Device applications can communicate over HTTP
  - No need to deploy client libraries
  - Extends MQ to many more client environments

- **WMQ can be used as a Web services transport**
  - Using SOAP over JMS standard, or by sending SOAP messages in raw MQ messages
  - As an integrated transport channel for Windows Communication Foundation

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

End of Service Dates for V5.3

- Distributed V5.3
  - Announced in June
  - Withdrawal from Marketing in September 2006
  - End of Service in September 2007
- z/OS V5.3.1
  - Announced February 6th 2007
  - Withdrawal from Marketing in March 2007
  - End of Service in April 2008
- Fee-based service extensions normally available for an extra 2 years
- 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/

- **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/

Thank
You

Additional Information

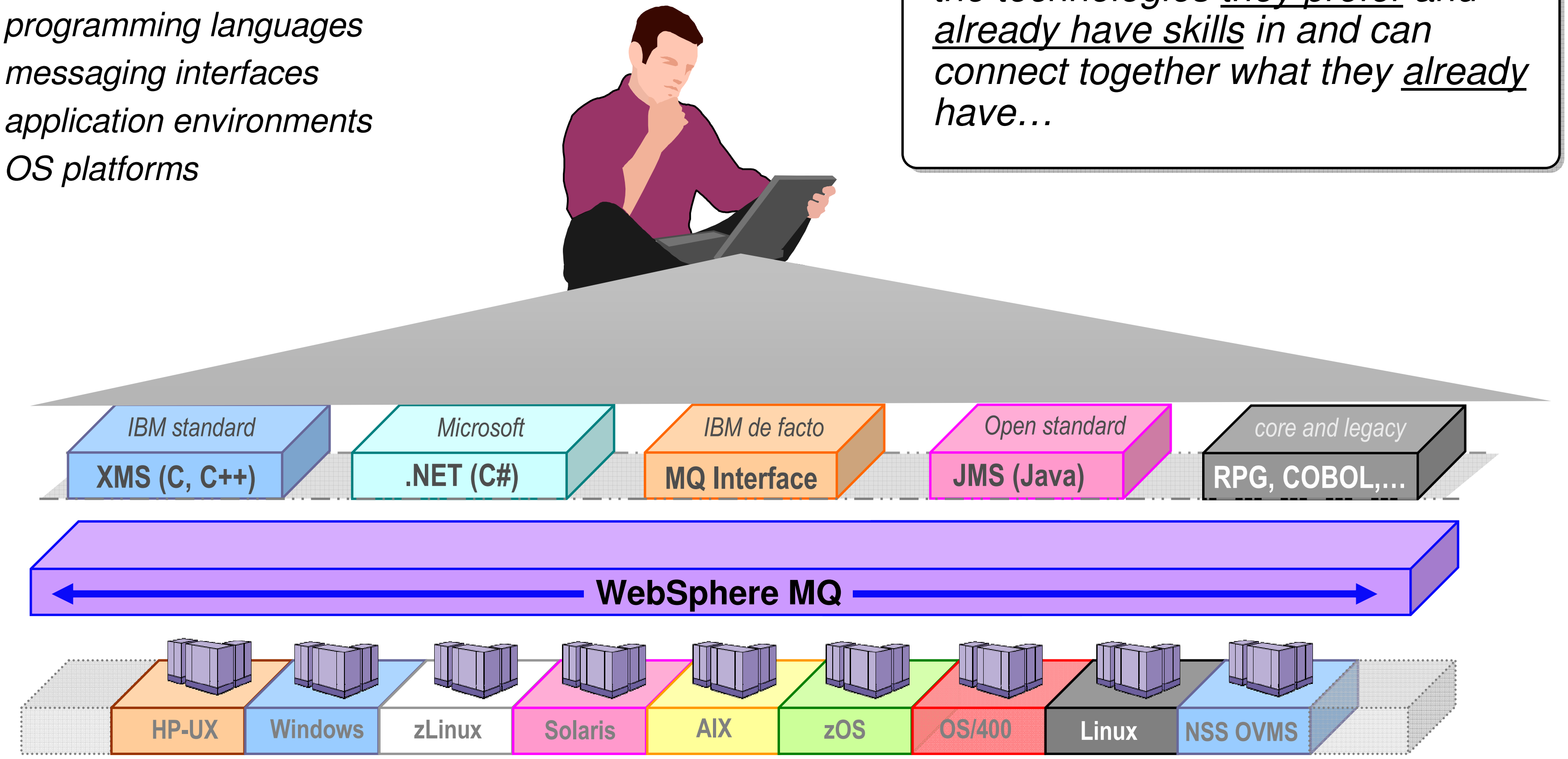
Introduction to WebSphere MQ for z/OS concepts

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
XMS (C, C++)

Microsoft
.NET (C#)

IBM de facto
MQ Interface

Open standard
JMS (Java)

core and legacy
RPG, COBOL,…

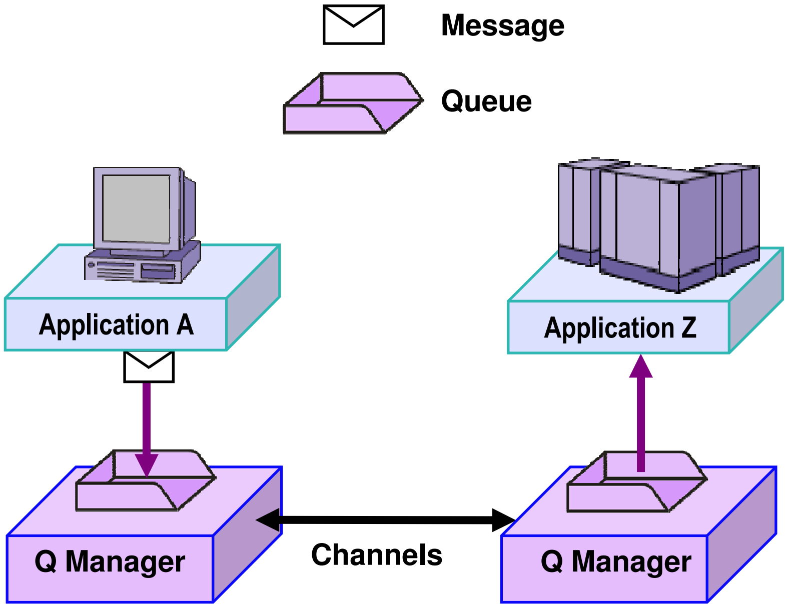
WebSphere MQ

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

80+ platform configurations

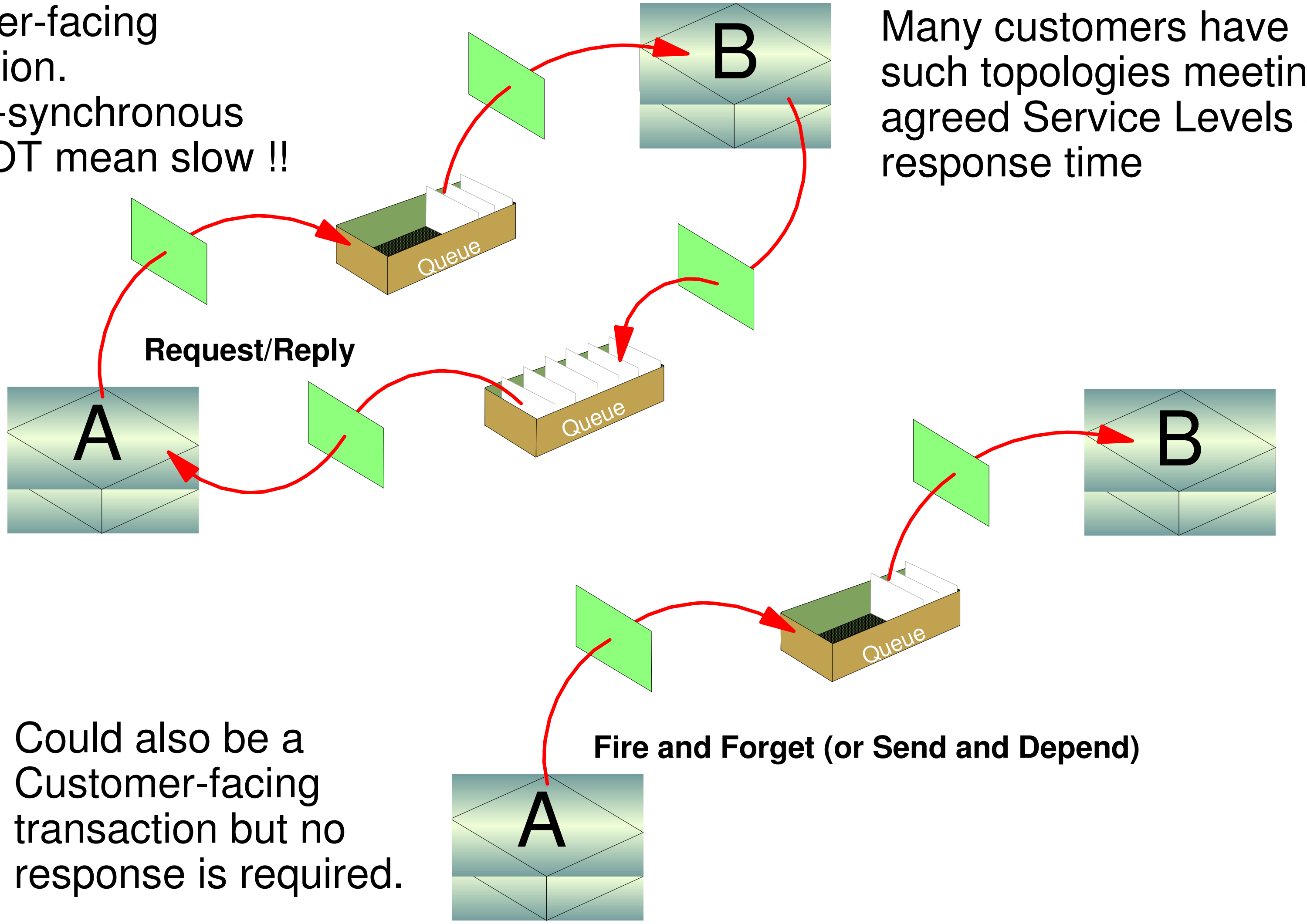
How does WebSphere MQ work?

- **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



Pseudo-synchronous or asynchronous styles

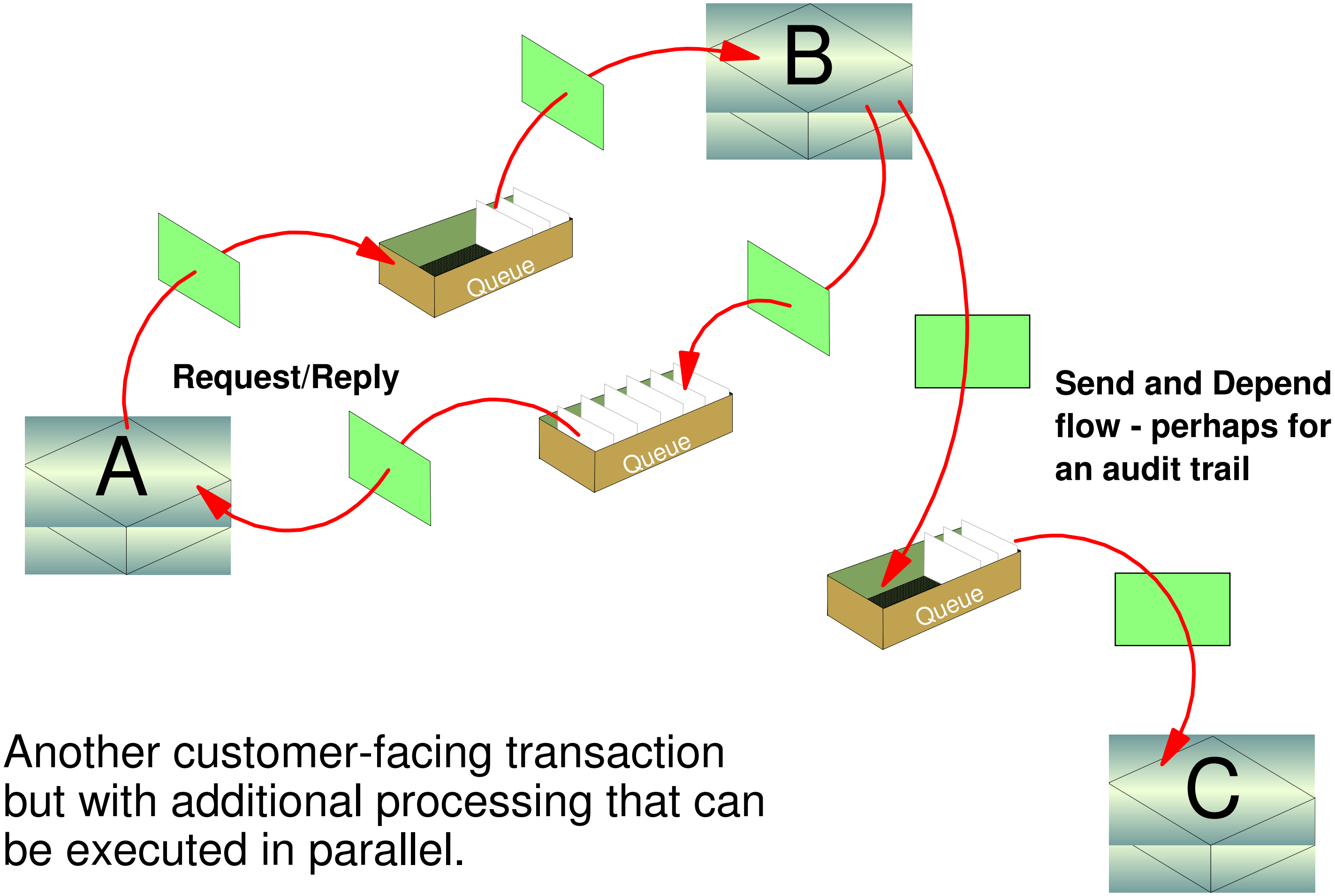
Customer-facing transaction. Pseudo-synchronous does NOT mean slow !!



Many customers have such topologies meeting agreed Service Levels of response time

Could also be a Customer-facing transaction but no response is required.

Pseudo-synchronous and asynchronous styles
Enables parallel processing

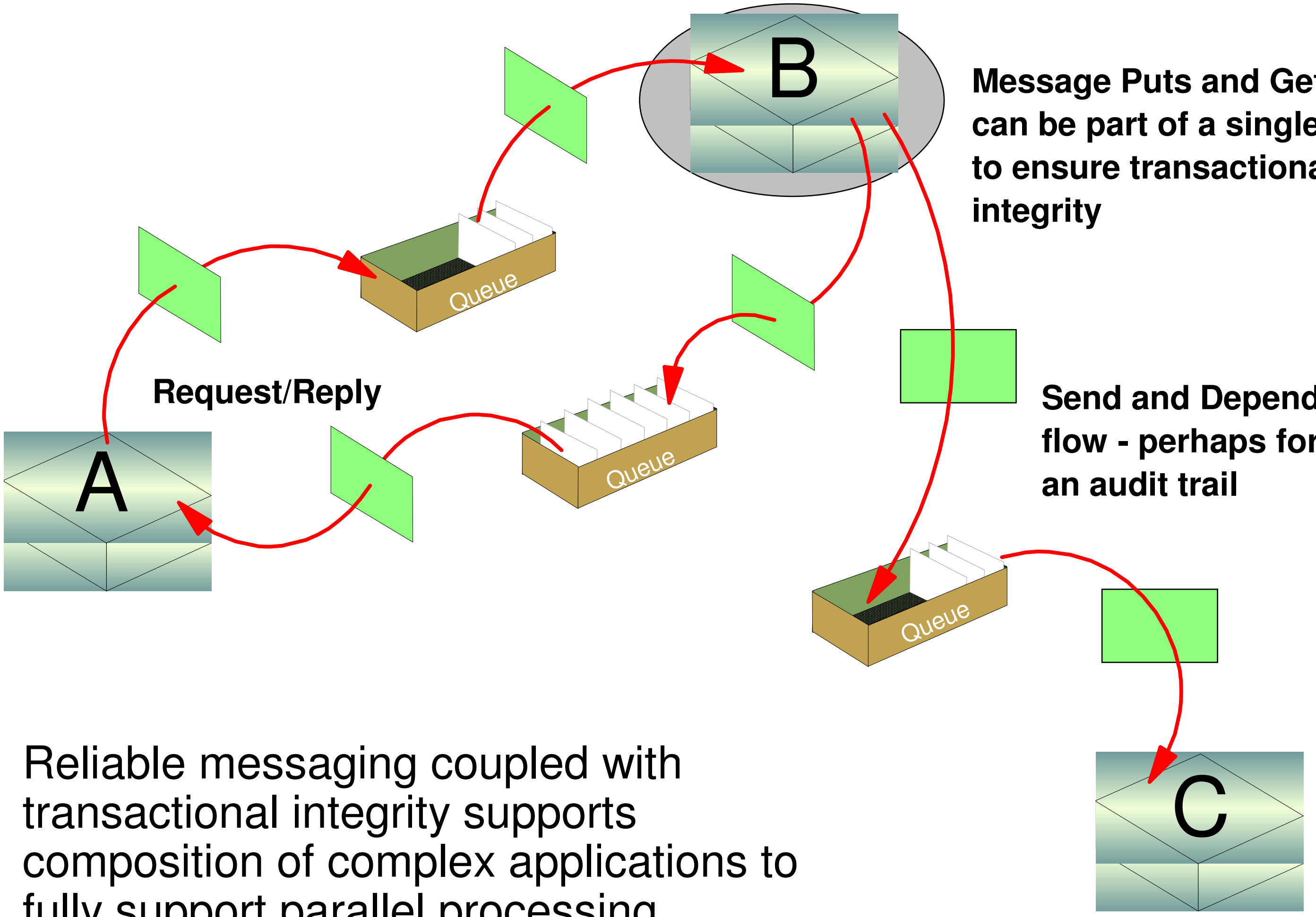


Request/Reply

Send and Depend flow - perhaps for an audit trail

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

Transactional support ensures all or nothing



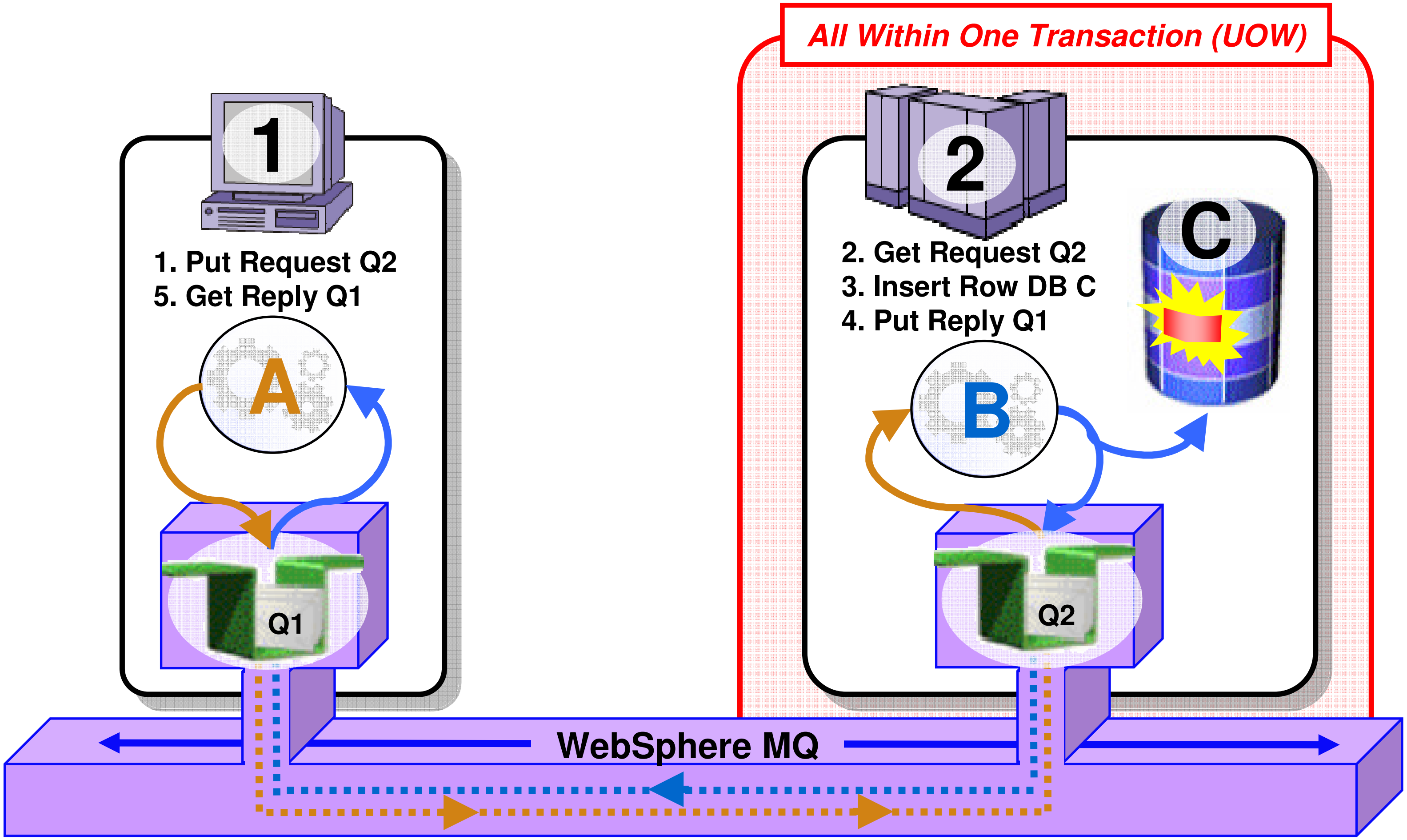
Message Puts and Gets can be part of a single UOW to ensure transactional integrity

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



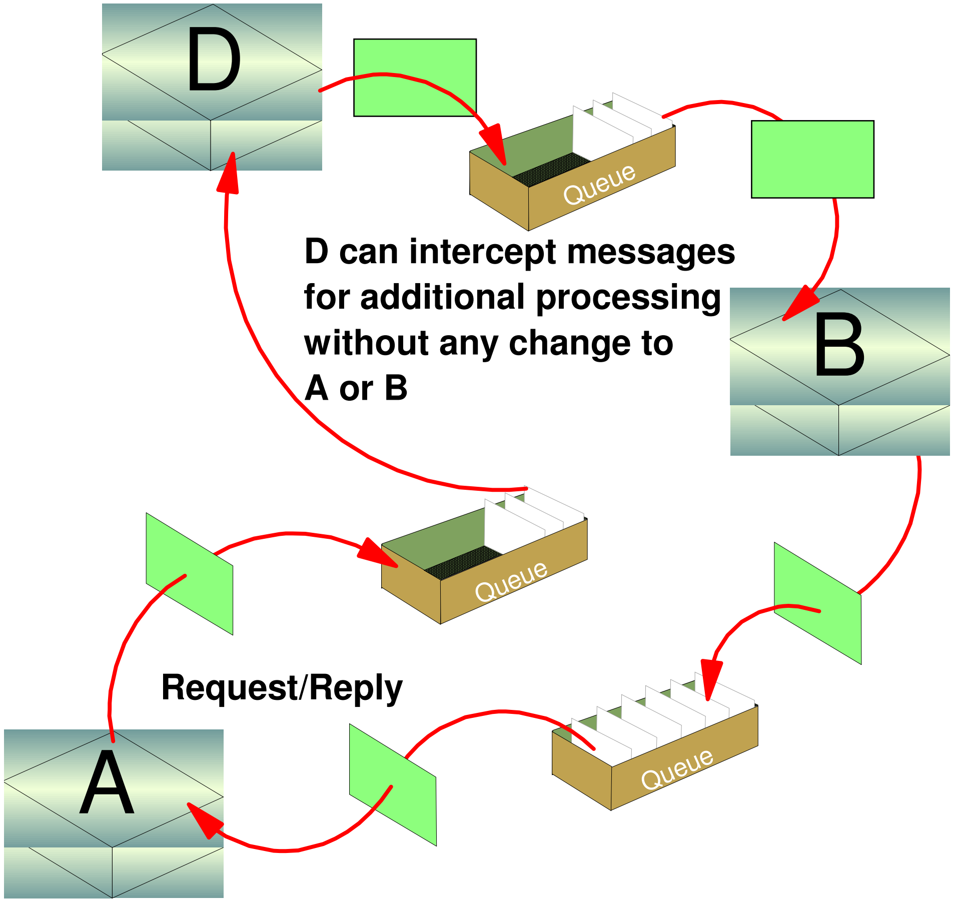
- Full support for CICS, IMS, WAS, DB2 SPs & Batch RSS
- Reliable Two-Phase commit involving other Resource Managers e.g. DB2

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

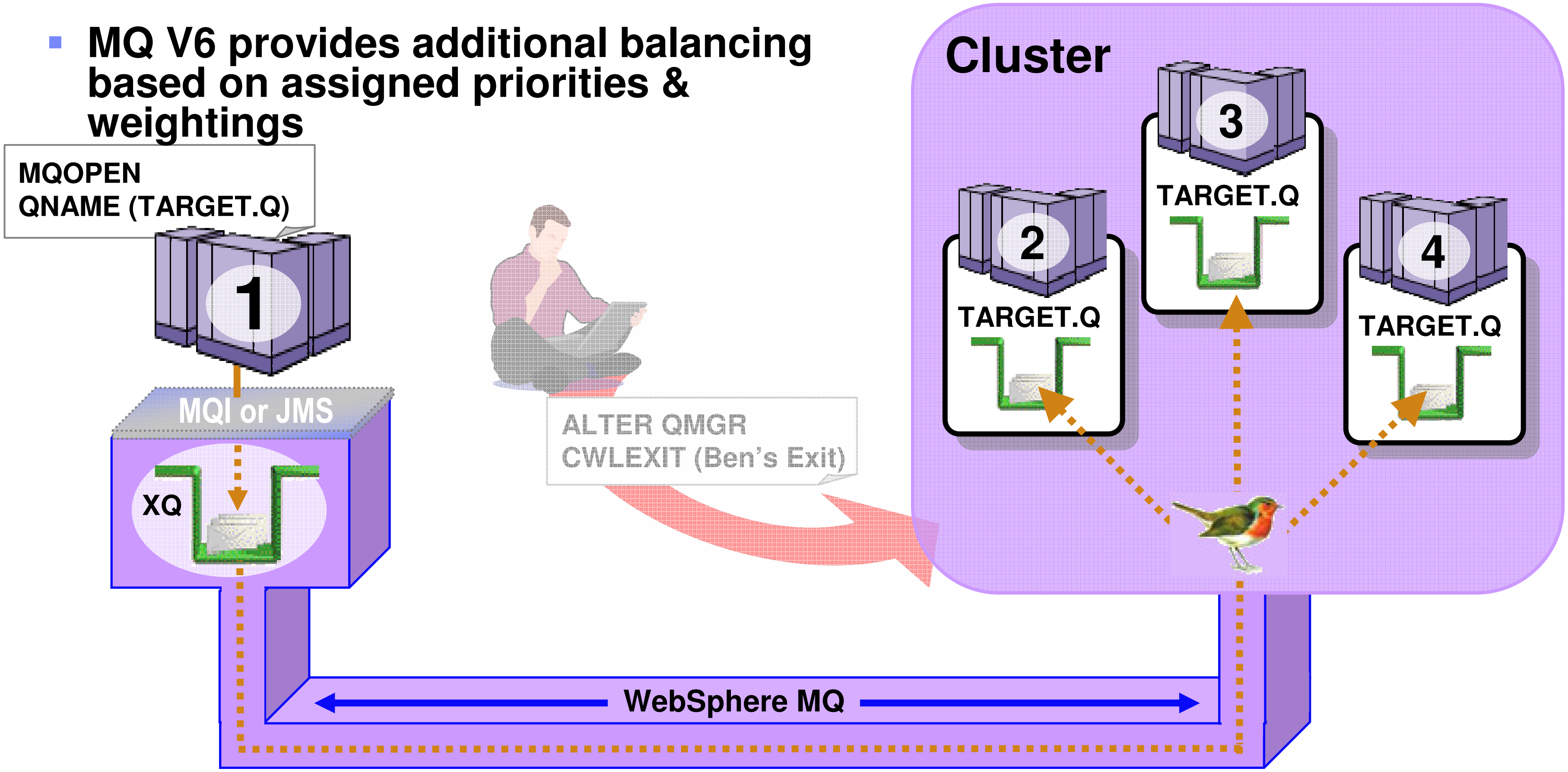


D can intercept messages for additional processing without any change to A or B

Request/Reply

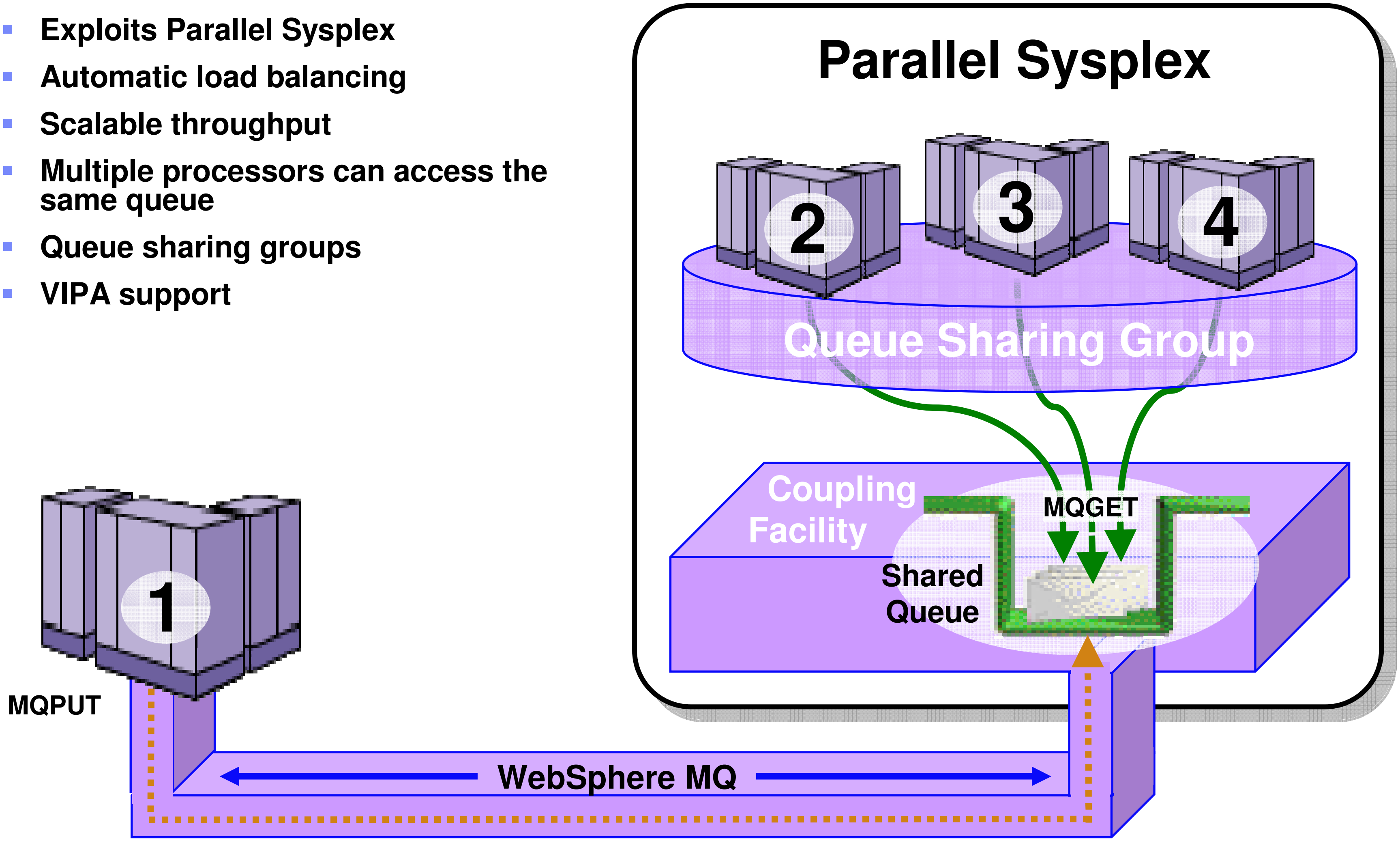
Workload Balancing with Clustering

- “Round Robin” balancing by default
- Custom balancing algorithms can be added as workload exits
- MQ V6 provides additional balancing based on assigned priorities & weightings



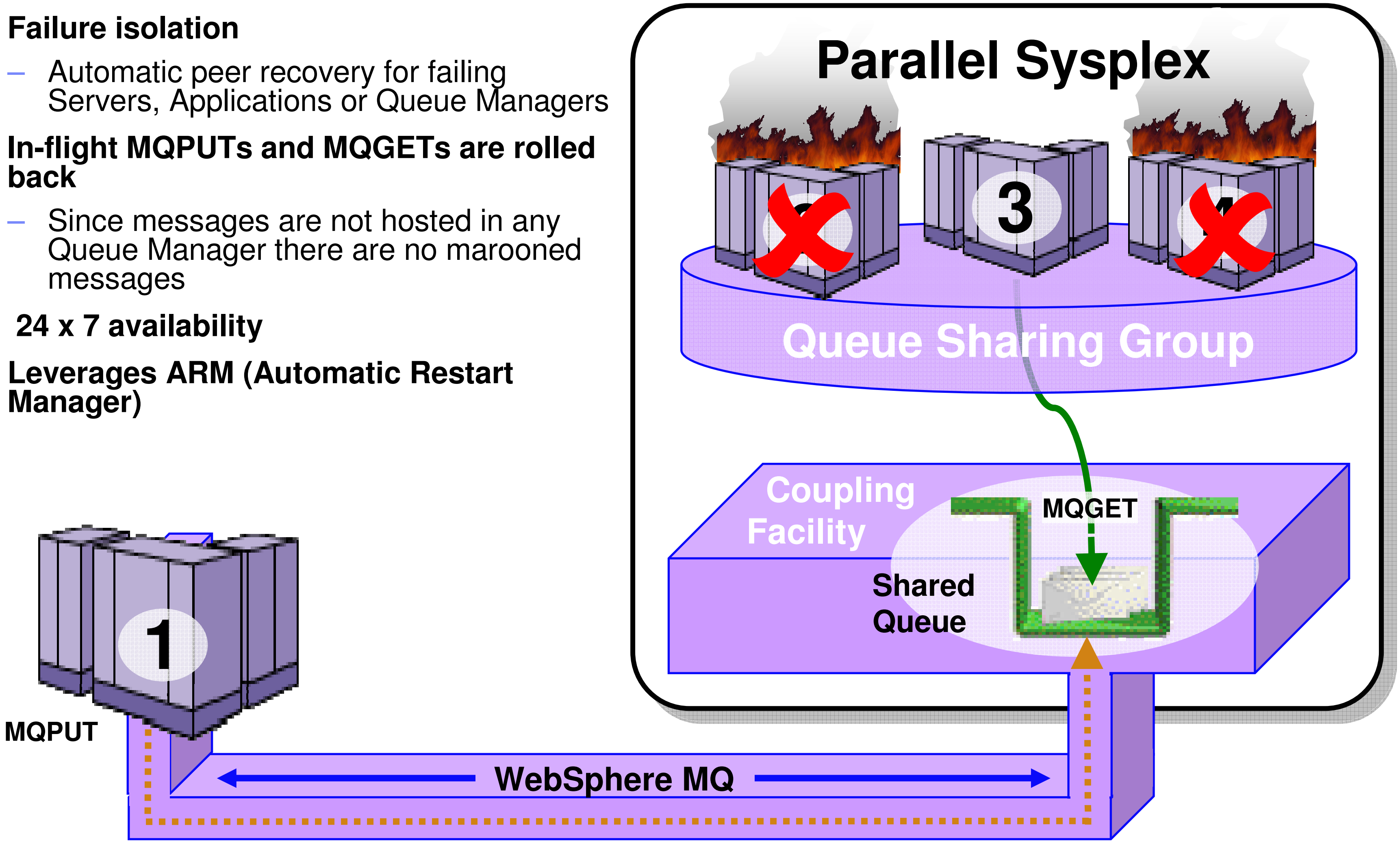
Shared queues on z/OS

- 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

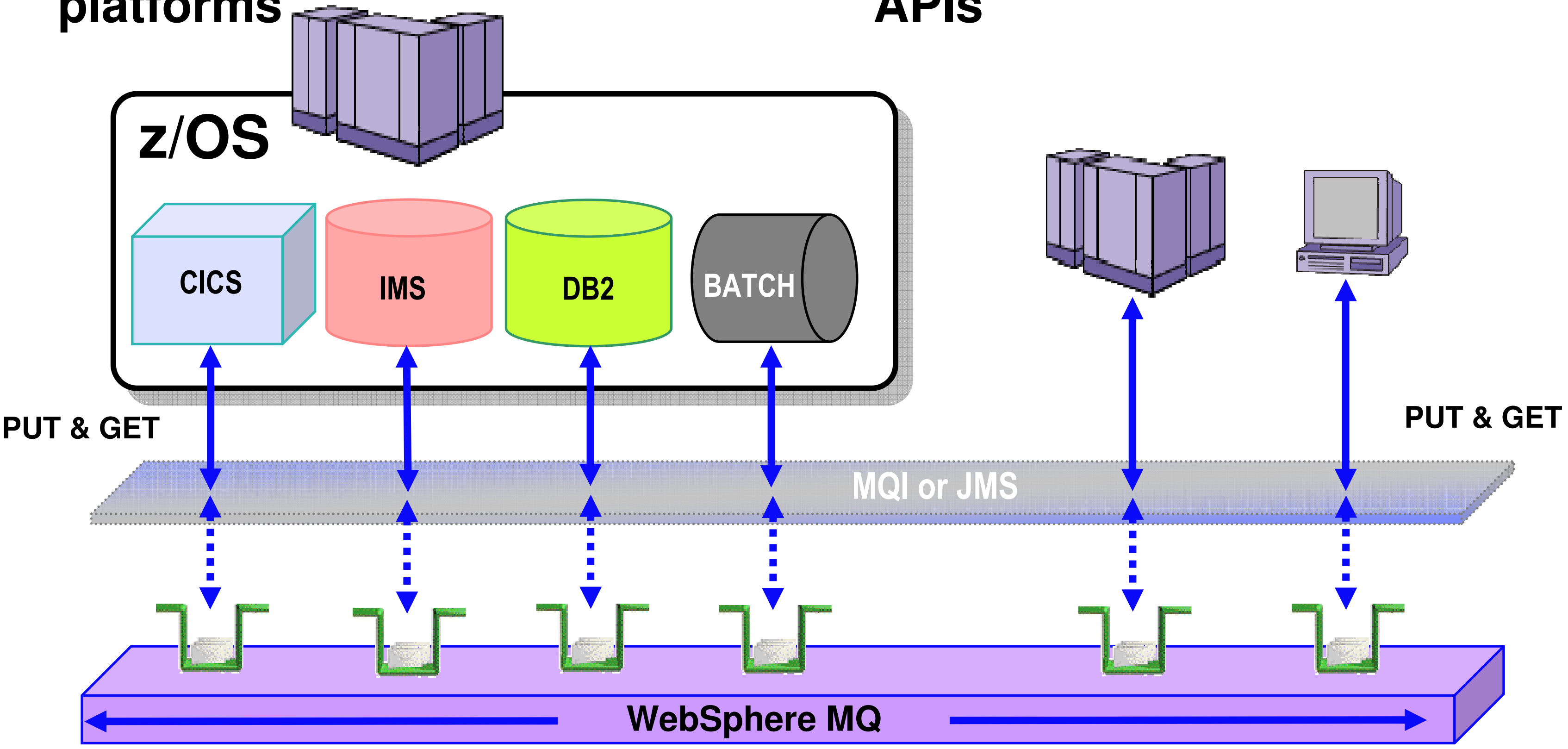
- **Failure isolation**
  - Automatic peer recovery for failing Servers, Applications or Queue Managers
- **In-flight MQPUTs and MQGETs are rolled back**
  - Since messages are not hosted in any Queue Manager there are no marooned messages
- **24 x 7 availability**
- **Leverages ARM (Automatic Restart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



Accessing CICS, IMS, DB2 SPs and Batch/TSO

Explicitly – Using the MQI or JMS API

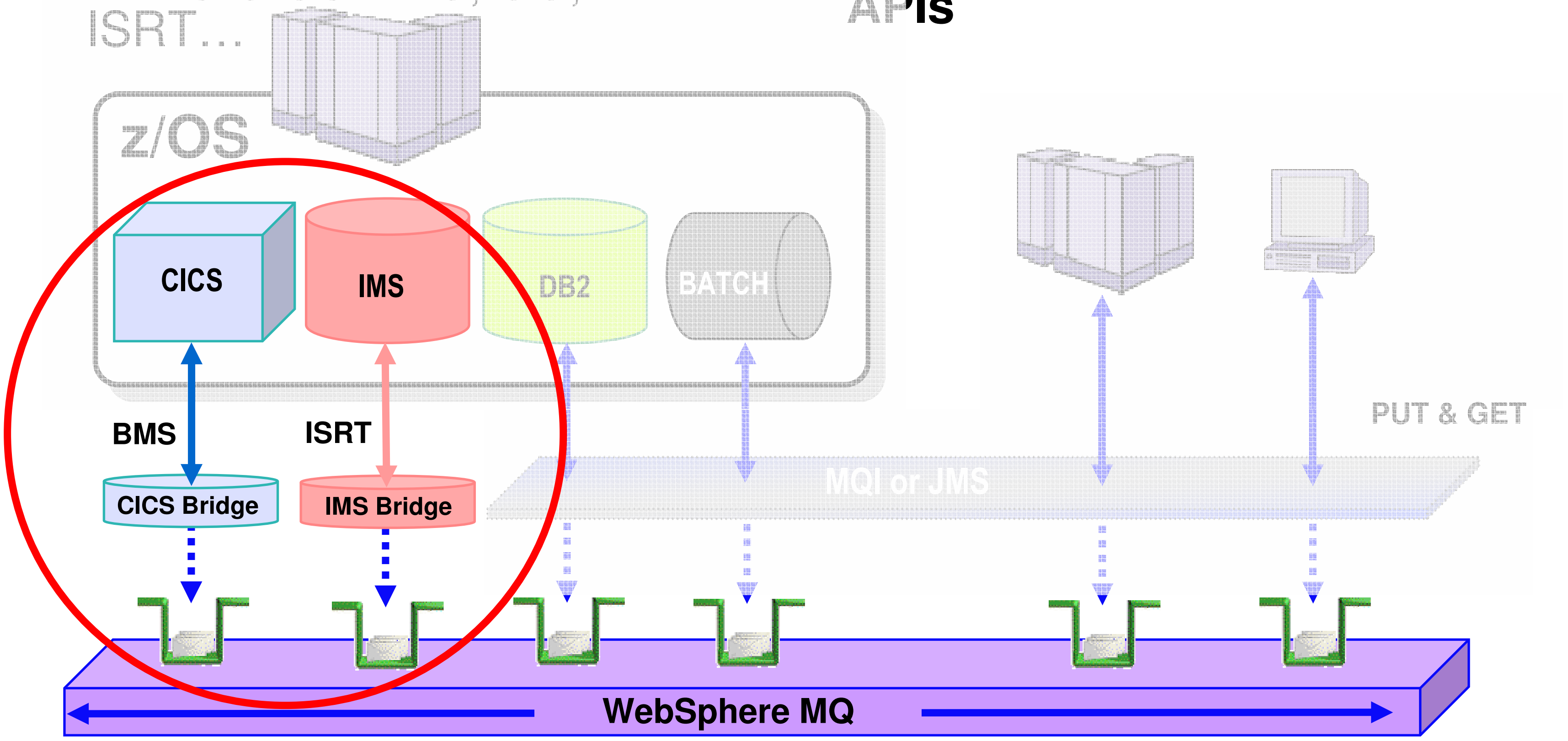
- Gives CICS and IMS access to messages from other z/OS regions and other non-z/OS platforms
- Allows other applications to access CICS and IMS without using CICS or IMS APIs



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**



WebSphere MQ for z/OS V6 and recent SupportPac highlights

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

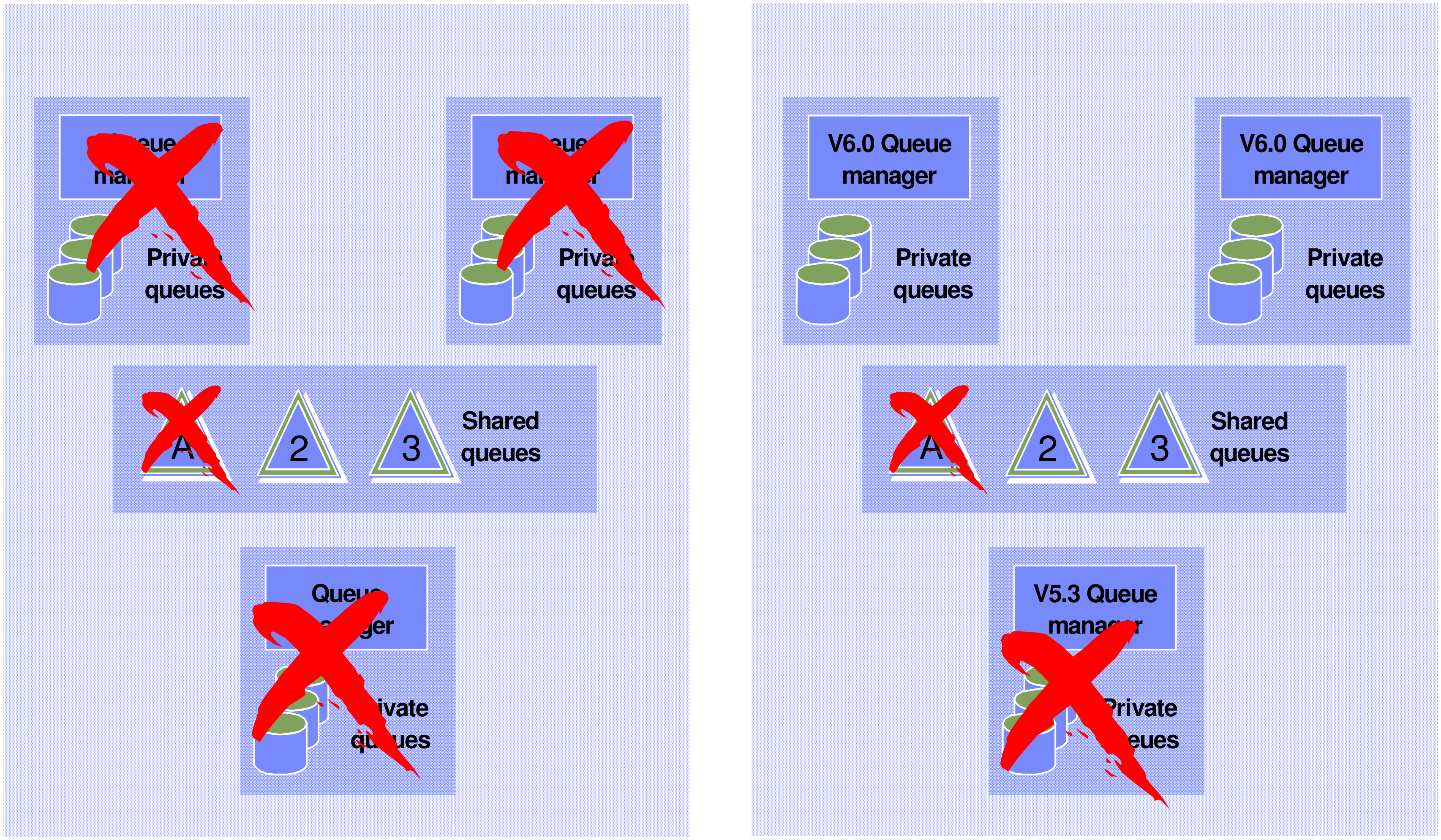
z/OS Availability – Large shared queue messages

- 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)
    ```

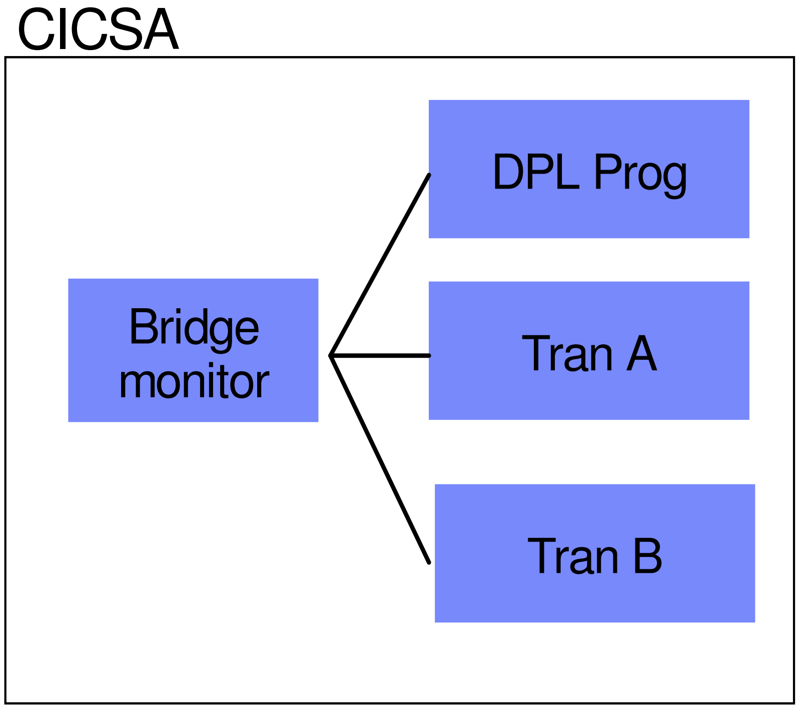
z/OS Availability – Multiple TCP/IP stack support

- 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



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.

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