

# WebSphere MQ Update

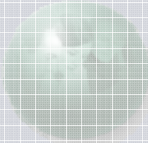
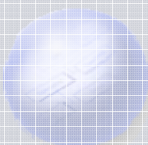
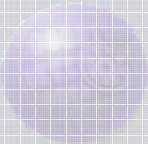
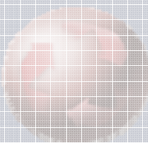



**John Kinchen**

***john\_kinchen@uk.ibm.com***

Messaging Clients Development Manager

# SOA Entry Points Help Customers Get Started

	What is it?	Value
People 	Deliver role-based interaction and collaboration through services	Improved productivity and flexibility by enabling targeted user interactions for improved business operations and collaboration
Process 	Achieve business process innovation through treating tasks as modular services	Greater innovation and flexibility through faster deployment and modification of business processes
Information 	Provide trusted information in business context by treating it as a service	Better business operations, more informed decisions and reduced risk with information delivered in-line and in-context
Reuse 	Service-enable existing assets and fill portfolio gaps with new reusable services	Lower risk and faster time to market by leveraging proven, time-tested functionality
<b>Connectivity</b> 	<b>Connect systems, users, and business channels based on standards</b>	<b>Reduced maintenance costs and greater reliability and consistency through flexible, any-to-any linkages</b>

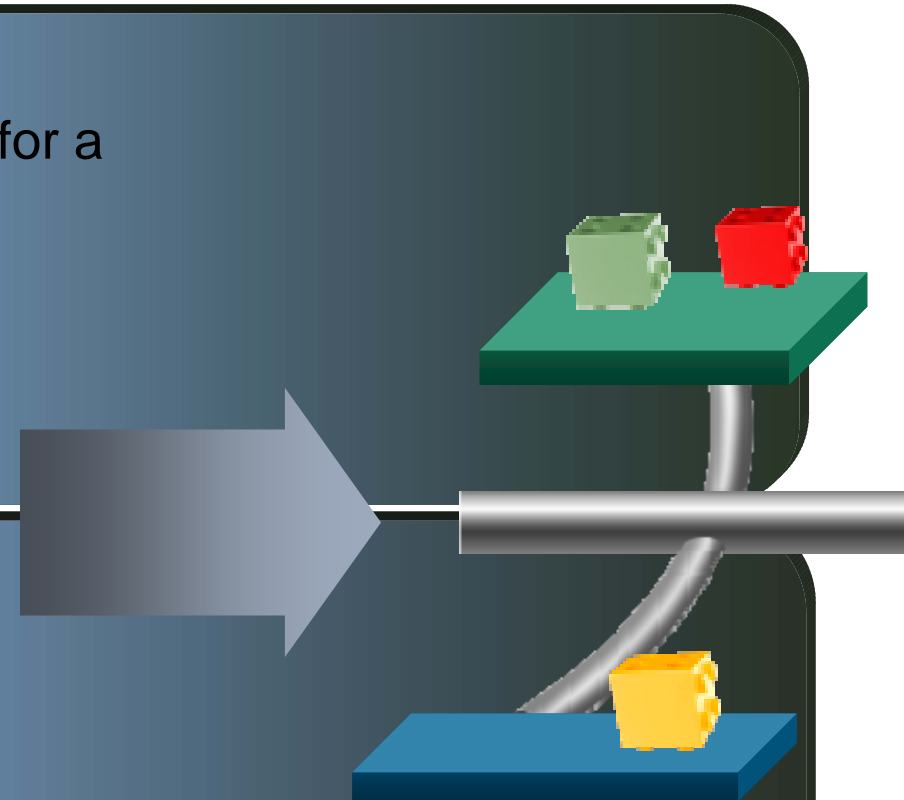
# SOA Entry Point to Connectivity

## Business Value

- Deliver services through new business channels for a secure, consistent user experience
- Service-based connections with trading partners
- Potential savings of 2 – 4x over custom-built integration or FTP

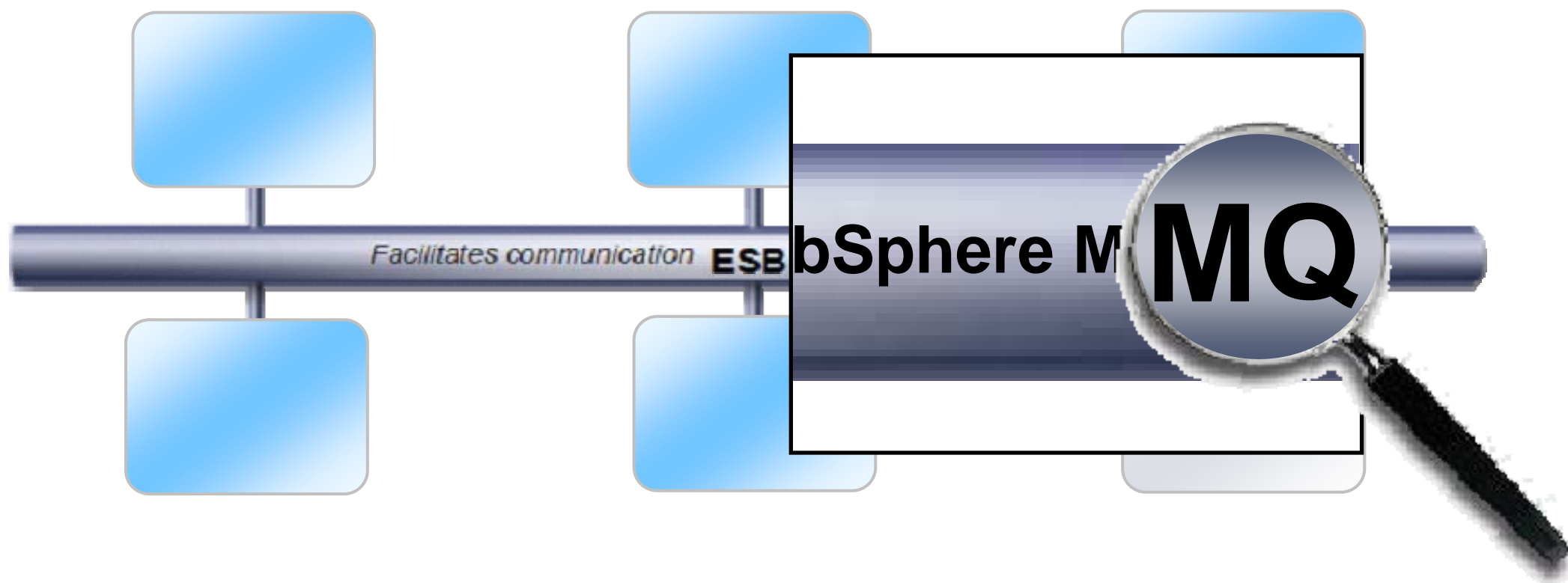
## Start with

- Messaging backbone as the foundation for SOA connectivity
- Enable mediated exchange between services, by leveraging an ESB
- SOA appliances integrate with ESB functions in a hardware form factor



# Messaging Transport for SOA Connectivity

- SOA Connectivity (ESB) integrates services and resources
- SOA Connectivity itself uses internal services and resources
- WebSphere MQ provides a proven messaging transport for SOA connectivity



# 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*

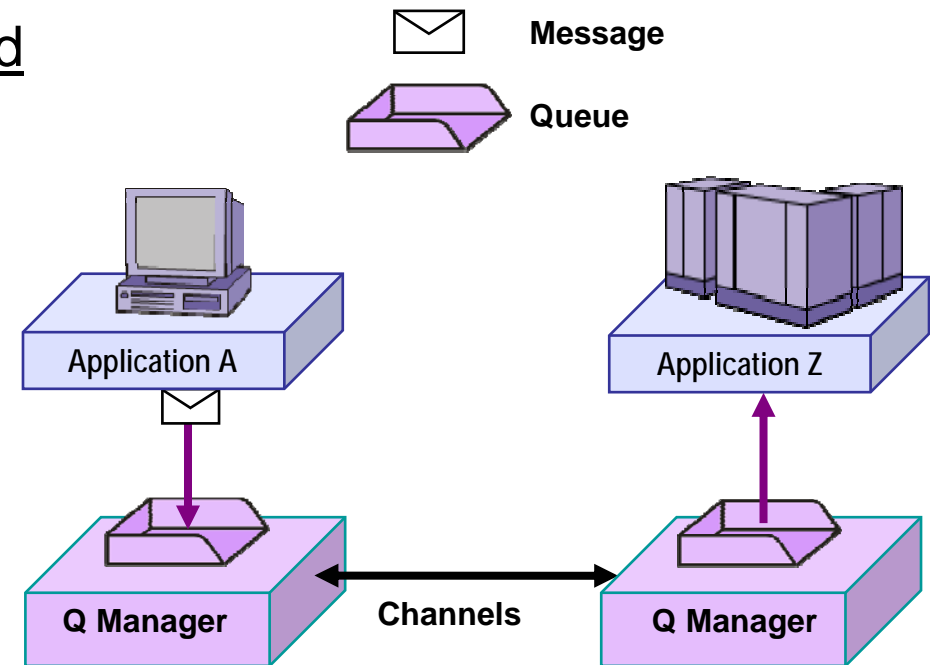
## 7 Keys to SOA Connectivity

1. Reliable – **proven, trusted, the standard**
2. Secure – **protects data end-to-end**
3. Time-flexible & Resilient – **connect whenever**
4. Transactional – **preserves integrity of data**
5. Incremental – **grows with your needs**
6. Ubiquitous – **connect virtually anything**
7. Basis for ESB – **underpins and extends**



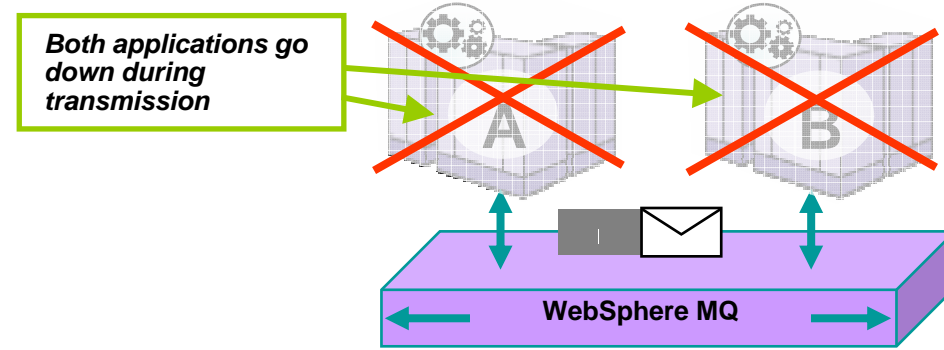
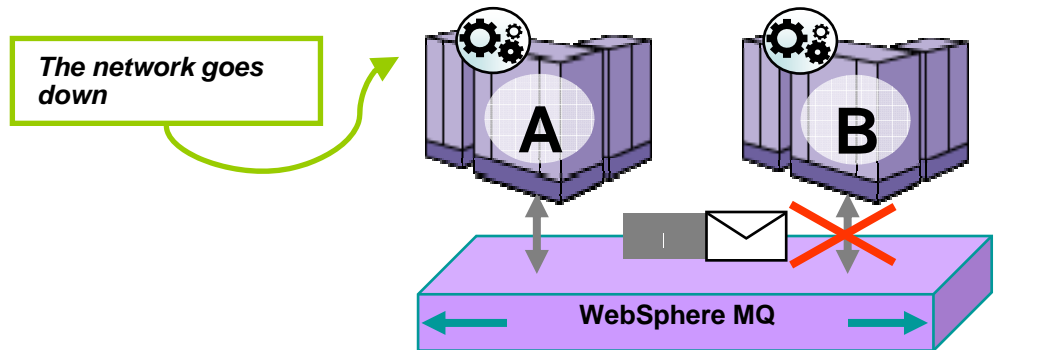
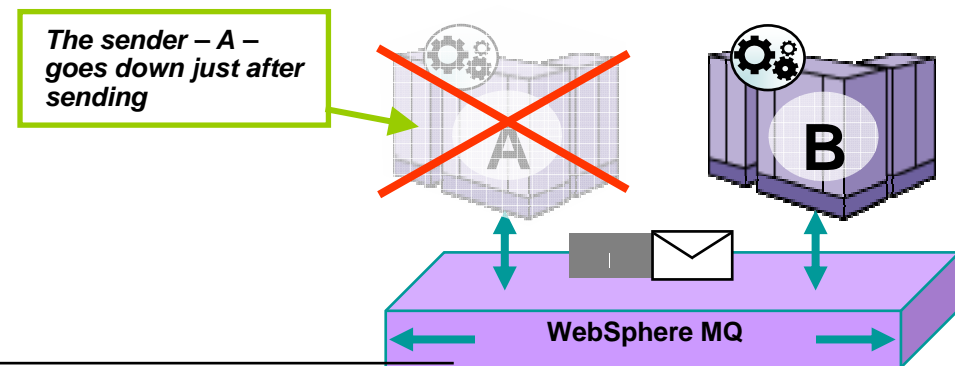
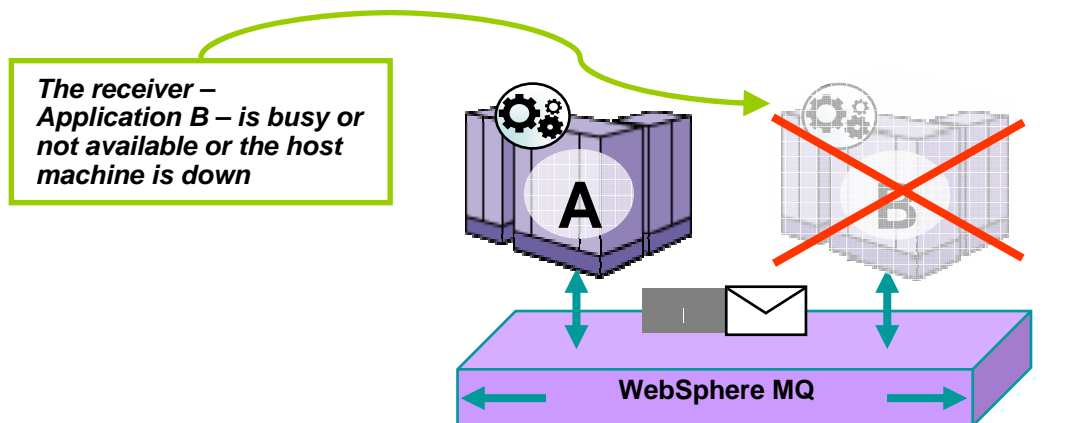
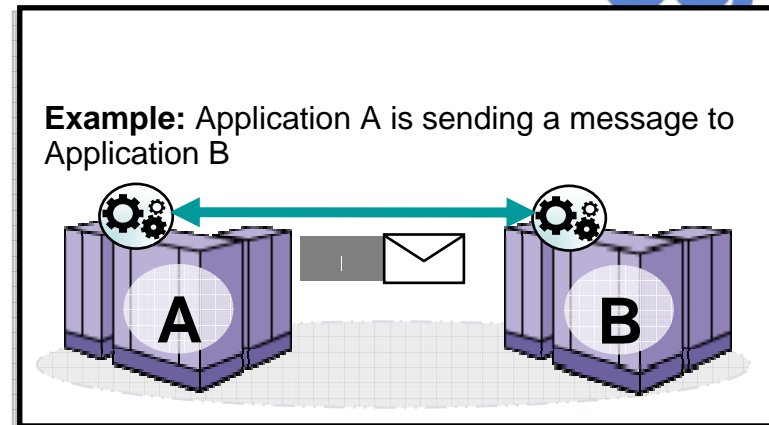
# How does WebSphere MQ work?

- Connects applications using **Messages** sent via **Queues**.
- Queues are owned by **QMs** which store and forward messages.
- Routing is dynamic and configurable
- **This allows applications to be very loosely coupled...**
  - Cuts location dependencies
    - Sender does not need to know where the receiver is running
  - Cuts timing dependencies (asynchronous)
    - Sender does not need to know whether the receiver is running
  - Cuts platform dependencies
    - Sender does not need to know the platform the receiver is running
  - Cuts data dependencies
    - With a Broker they do not even need to agree on the data format

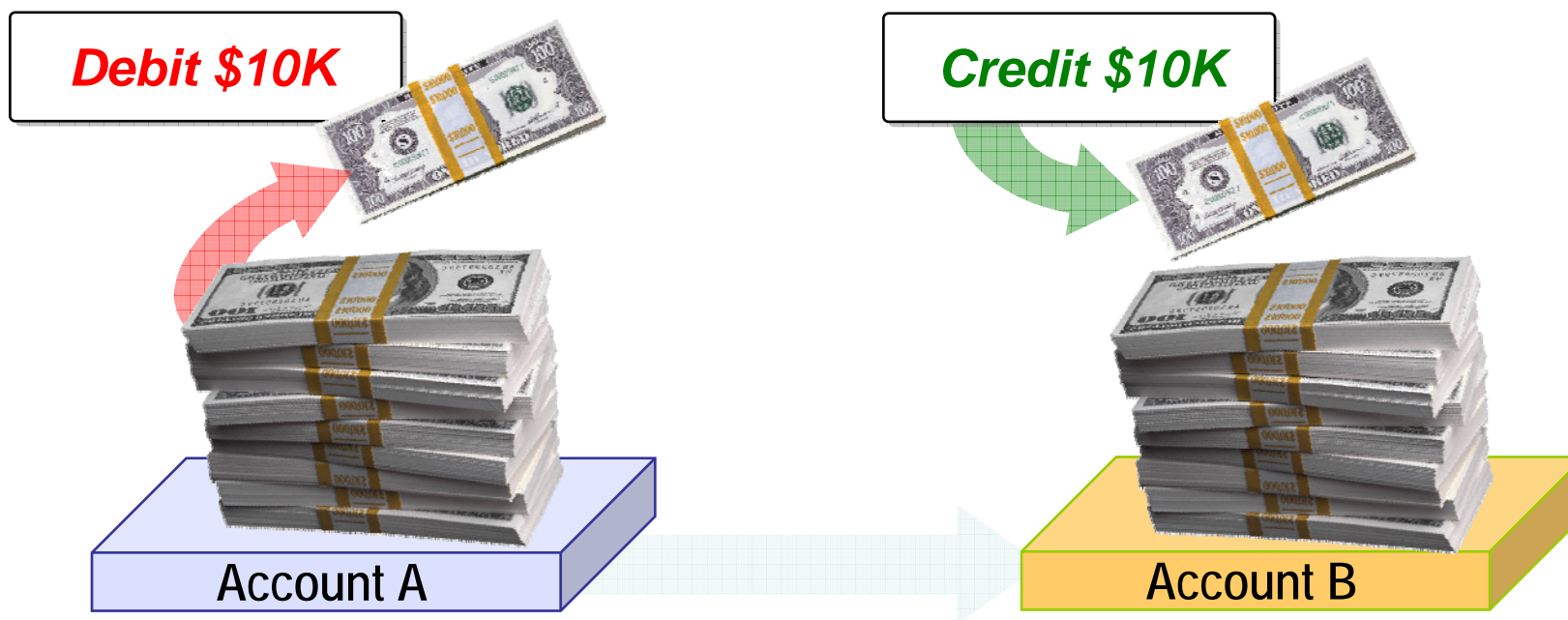


# Asynchronous = Flexible & Resilient

- SOA Connectivity must provide an always-connected experience for applications
  - Overcoming problems when SOA assets aren't available to talk or when IT systems and networks fail



# Transactions preserve the integrity of applications



**Both operations must succeed (or fail) exactly once  
Otherwise money goes missing!**

**WebSphere MQ is an XA compliant Transaction Manager. It can co-ordinate XA resources (e.g. messaging and database) in a single unit of work**

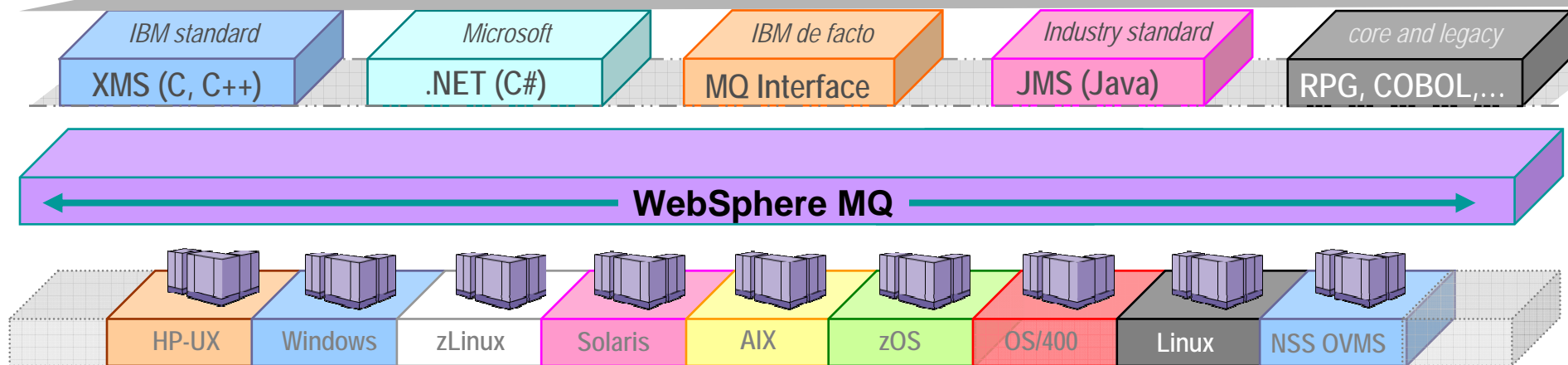
# Ubiquitous - WebSphere MQ connects virtually anything

- WebSphere MQ has probably the software industry's broadest support for:

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

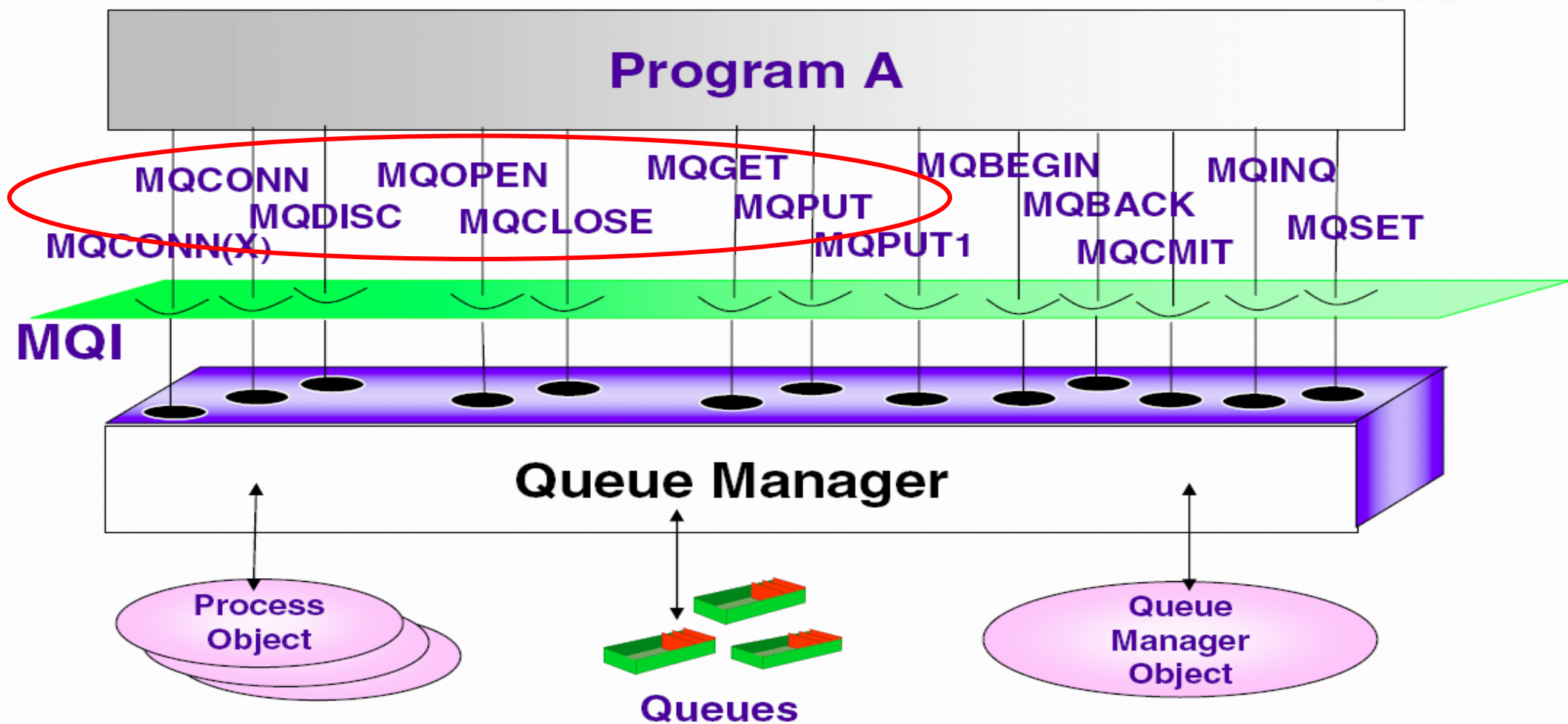


*Ubiquity of support gives developers the freedom to choose the technologies they prefer and already have skills in and can connect together what they already have...*



**80+ platform configurations**

# Cuts Platform Dependencies (single multi-platform API)

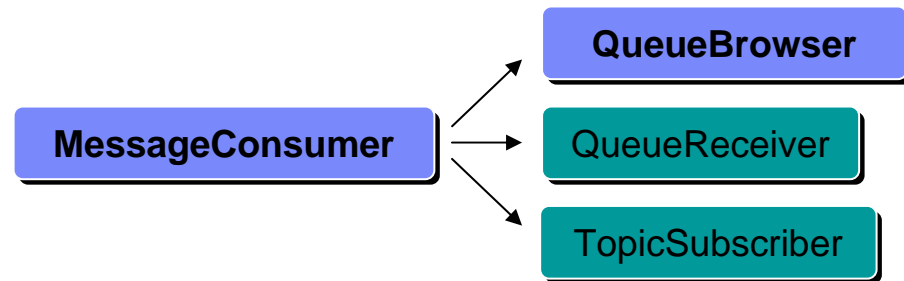
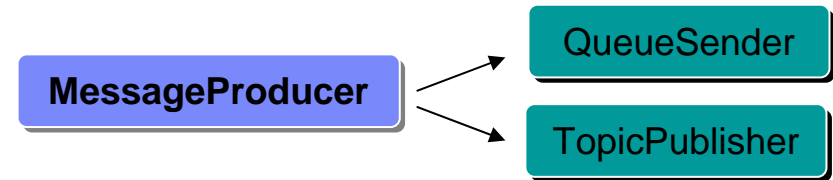
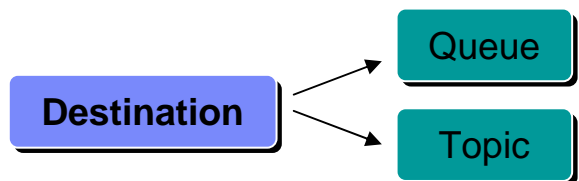
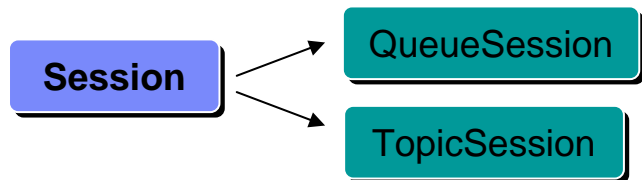
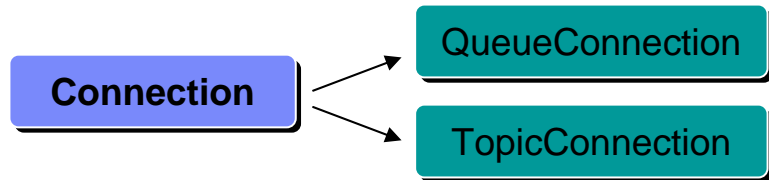
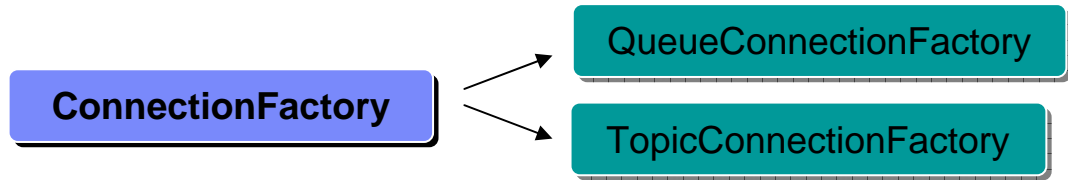


C, C++, C#, RPG, COBOL, PL/1, Java, JMS, Assembler (z/OS), Visual Basic, COM

# ...or Standards Based JMS 1.1 / XMS APIs

**Key**

- JMS 1.1 implemented in XMS
- JMS 1.02 classes not implemented

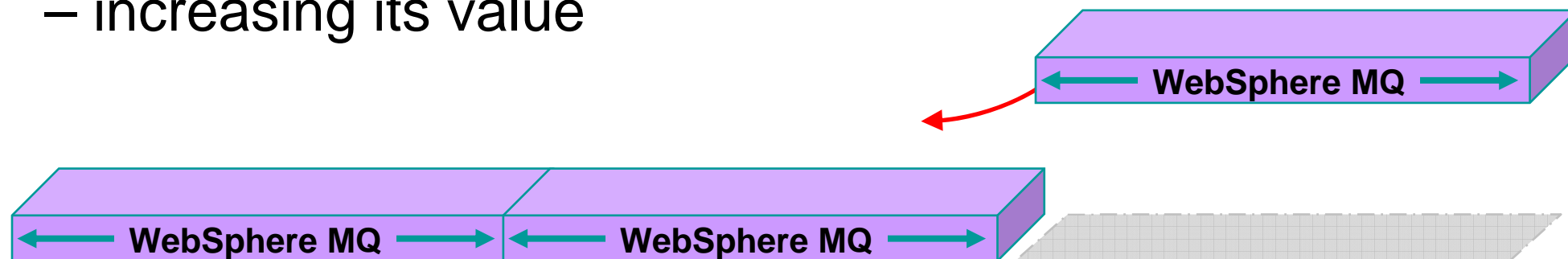


**JMS = Java**

**XMS = C, C++, C#**

# Connectivity that grows with your needs

- SOA Connectivity can start small and grow with your needs
- WebSphere MQ networks can grow incrementally
  - Adding a server, a client, even an application, one at a time
- As the MQ network grows more applications can participate
  - increasing its value



# WebSphere MQ

- Makes it easy to connect applications and systems
- Once-only assured delivery of data
- Many environments
  - Broad range of operating system and hardware platforms
  - Supports range of programming languages
  - Communications Protocols
  - Point-to-point and publish-subscribe styles
  - ... all available through simple APIs
- The industry standard for industrial messaging
- These values have not changed –
  - Still valuable concepts
  - A foundation requirement for SOA

# WebSphere MQ – Current Status

# Current Product Status

- V5.3 – Distributed platforms now out of service where V6 replacements exist
- V5.3.1 – z/OS end of service is April 2008
- V6 - No End of Service or Withdrawal from Marketing dates set
- V7 – Early Experience program has been running with customers and vendors
- V7 – GA June 2008
  - <http://www-306.ibm.com/software/integration/wmq/v7/>
- Look at URL for most recent info:
  - <http://www-306.ibm.com/software/info/supportlifecycle>

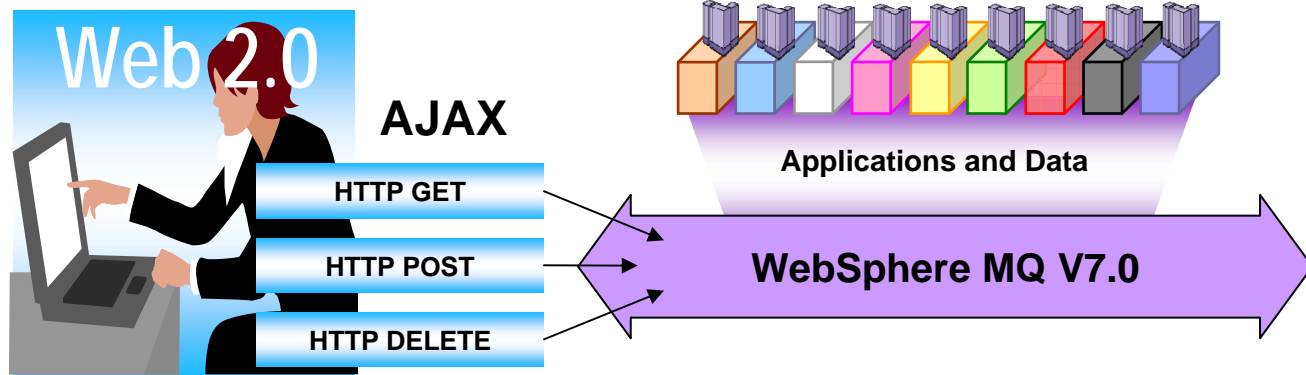
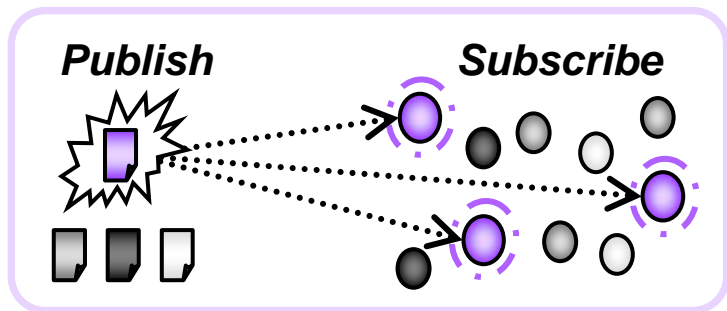
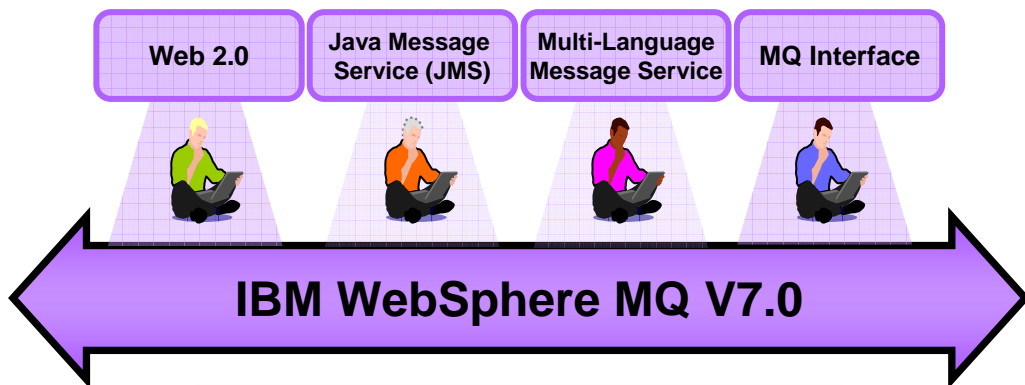
# WebSphere MQ V6 Updates

- XMS (1Q06: SupportPacs)
- IBM Support Assistant (1Q06: 6.0.1.1)
  - Makes it easier to access product information and raise defects
- New Plug-ins for the WMQ Explorer (3Q06: SupportPac & 6.0.2.0 )
- JCA for JMS (1Q07: 6.0.2.1)
  - Makes it easier to use WMQ JMS in any J2EE-compliant application server
- Support for 64-bit Windows on x86-64 and 64-bit JVM on z/OS (1Q07: 6.0.2.1)
- HTTP interface (2H07: SupportPacs MA0Y, MA94)

**Majority of these updates are now part of WMQ V7.0 base product**

# WebSphere MQ V7.0

## Universal Messaging Backbone for SOA and Web 2.0



# WebSphere MQ V7.0

- Central requirement was to improve JMS implementation
  - Better performance
  - Improved ease-of-use
  - Improved servicability
- Extension of publish/subscribe capabilities
  - Designed with Message Broker in mind
- Easier programming in any environment
  - Some features suggested by JMS requirements are useful in MQI

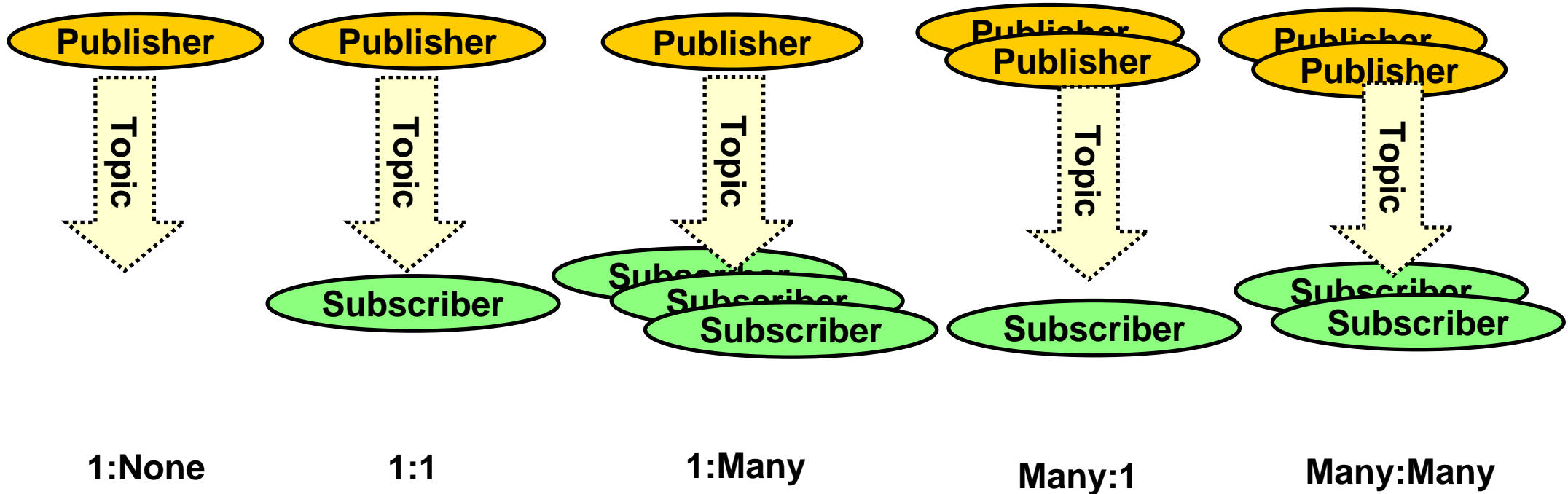
# Platforms

- Essentially the same platforms as V6
  - AIX, HP-UX (x2), Solaris (x2), Linux (x4), i5/OS, z/OS, Windows
  - Updates to base OS levels
  - See <http://www-306.ibm.com/software/integration/wmq/requirements/index.html>
- Only one product for Linux/zSeries
  - 31-bit version is removed; 64-bit edition continues
- Drop Windows 2000
  - Windows XP is base level
  - Vista supported
- Windows x64
  - Adds 64-bit application support to single Windows package
  - Supporting existing 32-bit applications
  - Some exits will require recompiling to support both 32 and 64-bit modes
- Java 1.4.2 and later

# Publish/Subscribe

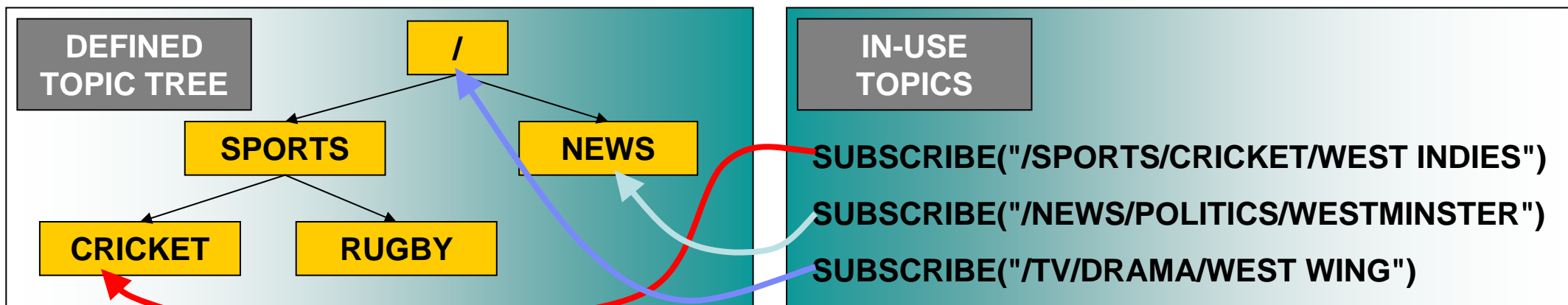
- A natural part of the JMS API
  - Combines both Publish/Subscribe and Point-to-Point patterns
  - Now also a natural part of the native MQI
- Point-to-point asynchronous messaging decouples applications
  - But still implies a one-one relationship between sender and receiver
- Publish/subscribe is 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

# Loose Coupling with Publish/Subscribe



# Publish/Subscribe Administration

- Based on Topic Strings
- Topic Objects
  - New object type, like queue or channel definitions
  - A 48-character name which has a longer attribute for full **topic string**
  - Defines major points in a topic tree
  - No additional definitions needed before applications can start using pub/sub
- In-use topics
  - The topic strings that applications are publishing or subscribing on
  - Inherit attributes (e.g. security) from the "closest" defined topic object
  - Not defined administratively, but can be viewed



## Publish/Subscribe Administration (2)

- Support for durable and non-durable subscriptions
  - With durable, a client can go away and come back later without missing messages
  - Non-durable exist only for the lifetime of the application
- Subscriptions
  - Able to see who is subscribing to topics: like DISPLAY QSTATUS
  - Able to create subscriptions on behalf of a third party
- Security
  - Use of a topic is restricted by permissions on the associated topic object
  - On z/OS drives need for mixed-case support in RACF
  - Follows existing WMQ model for security configuration (SAF or OAM)
- Conversion of point-to-point applications without code changes
  - Administrative changes to objects
  - A queue alias can point to a topic, not just a local queue

# Publish/Subscribe in the WMQ Explorer

The screenshot displays the WebSphere MQ Explorer interface. The 'Navigator' pane on the left shows a tree view of the IBM WebSphere MQ environment, including Queue Managers, All, Development Queue Managers, and v7. The 'Content' pane on the right shows the 'Topics' view for the selected v7 queue manager. A context menu is open over the v7 folder, listing various actions such as 'Compare with...', 'Status...', 'Delete...', 'Clear Retained Publication...', 'Topic Status - Subscribers...', 'Topic Status - Publishers...', 'Test Publication...', 'Test Subscription...', 'Create JMS Topic...', 'Object Authorities', and 'Properties...'. The 'Topics' table in the background lists several topics, including SYSTEM.ADMIN.QMGR.EVENT.TOPIC and SYSTEM.BASE.TOPIC.

Name	Topic String	Description
SYSTEM.ADMIN.QMGR.EVENT.TOPIC	SYSTEM.ADMIN.QMGR.EVENT.TOPIC	
SYSTEM.BASE.TOPIC		Base topic for resolving
		Admin stream for queue
		Default stream for queu

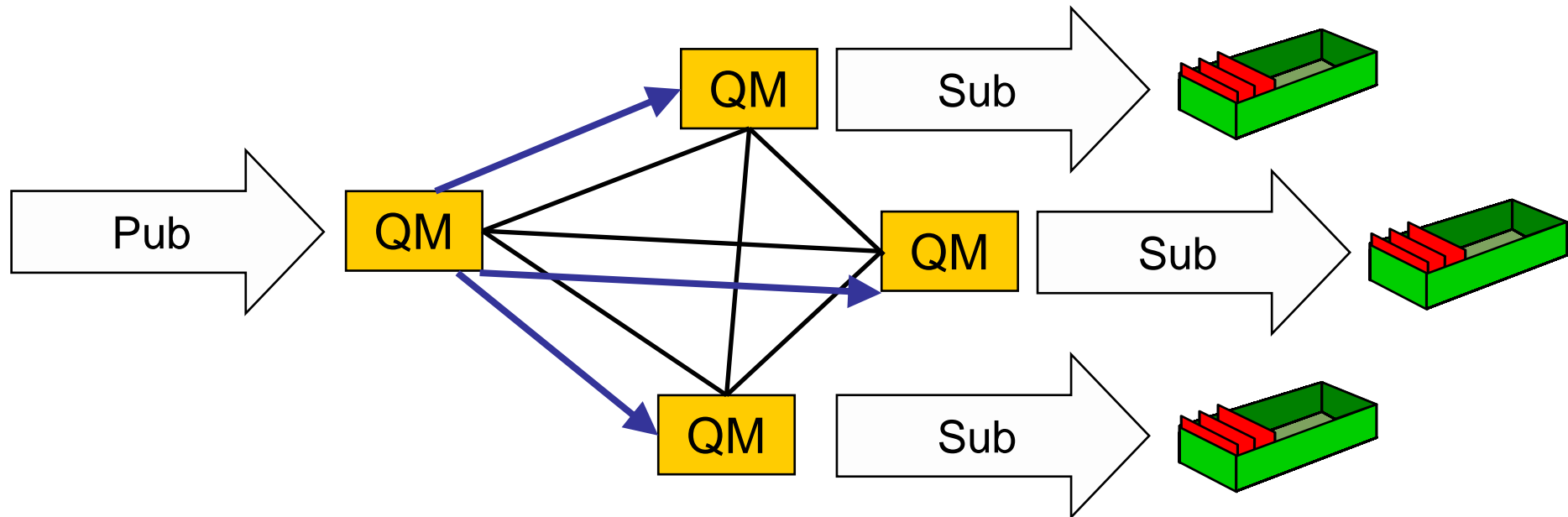


# Publish/Subscribe Topologies

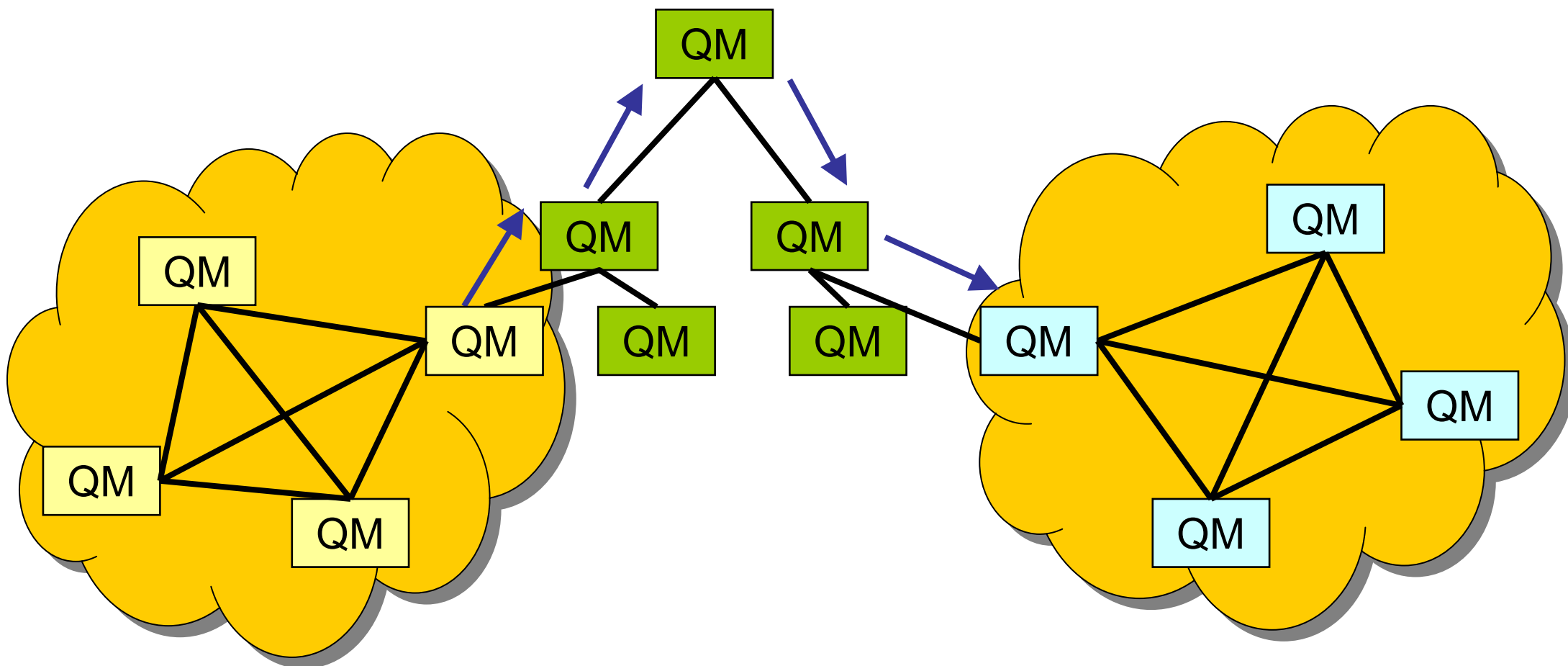
- WMQ V6 publish/subscribe networks based on **hierarchies**
- WMB publish/subscribe networks based on **hierarchies of collectives**
  - All systems in a collective are connected to each other (mesh)
- WMQ V7 publish/subscribe networks based on **hierarchies and clusters**
  - Clusters are built on existing WMQ cluster technology
  - With interoperability to other pub/sub systems through hierarchies

# Publish/Subscribe in a Cluster

- Consistent topic definitions in cluster
- Multiple routes across cluster



# Publish/Subscribe in Combined Hierarchy & Clusters

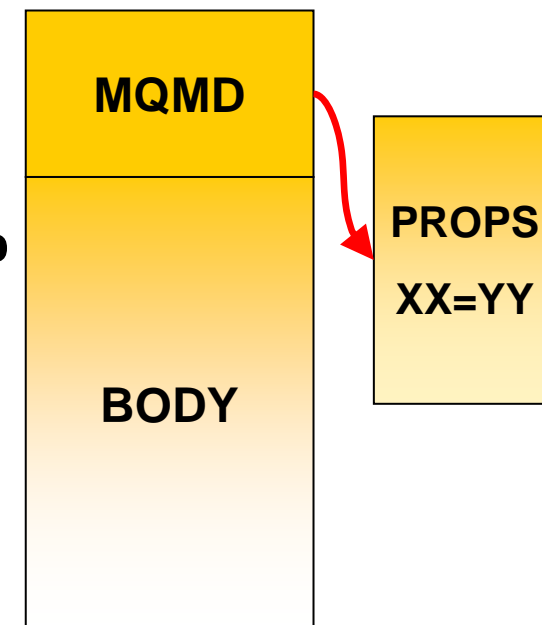


# Publish/Subscribe Application Programming

- Cannot significantly change the JMS API
  - But we want some of its facilities more easily available in the MQI
  - JMS implementation exploits new MQI functions
- New verbs for subscribing
  - **MQSUB** - Manage subscription
    - Retained publications delivered immediately after subscribing
  - **MQSUBRQ** – Request Subscription
- 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

# 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
  - Object attributes define behaviour
  - Defaults will create RFH2 folders



# 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 QM
  - Not looking inside message body
    - Message Broker still required for content filtering
- Cooperative Browsing
  - Efficient interface for applications reading from the same queue
  - No races – messages locked but available to any cooperating process

# Programming in Java

- JMS read/write access to all MQMD fields as properties
  - Have to explicitly enable this in the application program
  - Allows the application to go beyond the JMS specification
- JMS access to the raw message content
  - Can treat the whole body as a byte array property
  - Can see RFH2 folders that would normally be stripped
- Message Header Classes for Java
  - Updated and supported version of MS0B SupportPac
  - Makes it easy to build and parse PCF structures
  - Extended to handle other MQI message header formats
    - e.g. MQCIH, MQDLH classes

# Application Migration

- New verbs remove need for some of the older interfaces
  - Which will be deprecated – though not removed immediately
  - PCF and RFH facilities for WMQ publish/subscribe applications: Identity, Streams
- A single application cannot mix new verbs with old options
  - Can't use RFH(Register Publisher) and MQPUT(topic) in same program
- Option available to translate old-style pub/sub commands
  - Both RFH and RFH2 interfaces for WMQ and WMB pub/sub applications
  - Not needed when all publish/subscribe applications have been converted
- Migration step to convert existing WMQ and WMB subscriptions
- Most common pub/sub "application" is IBM-supplied JMS layer
  - Client design to automatically take advantage of new facilities when they exist
- New MQI operations are not available for old VB and ActiveX programs
  - Use .Net classes instead

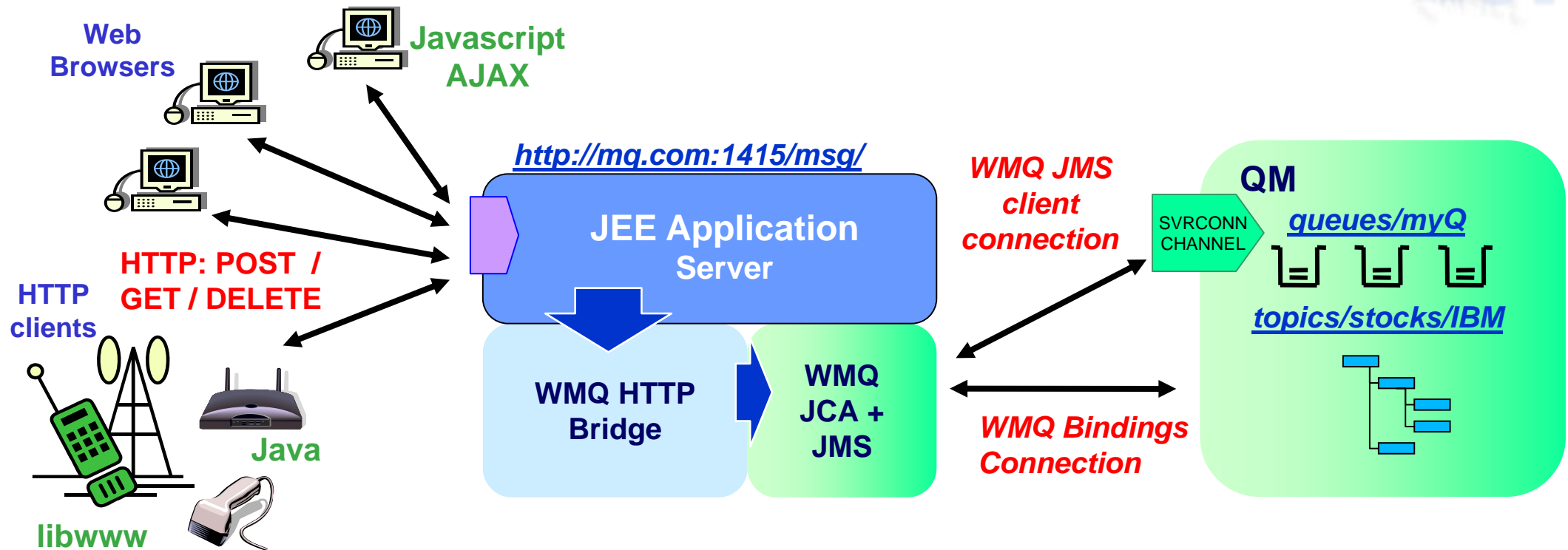
# Client Performance

- Traditional WMQ non-persistent messages more reliable than some need
- "Read Ahead" for Receiving Messages/Publications:
  - Messages sent to a client in advance of MQGET, queued internally
  - Administrative choice – no application changes needed
  - Higher performance in client
- "Asynchronous Put" for Sending/Publishing Messages:
  - Application can indicate it doesn't want to wait for the real return code
    - Maybe look for return code later – **MQSTAT** verb
  - Maintains transactional semantics
  - Higher performance in client

# Client Connection Management

- Shared Client Conversations
  - Several connections from the same process can be handled on the same socket
- 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 QM
- Free connections to z/OS for administration programs like WMQ Explorer
  - Limited number of clients permitted by V7 license without CAF

# WebSphere MQ Bridge for HTTP - 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
  
- 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)

		HTTP verb mapping			
Resource	Sample URIs	GET	POST	PUT	DELETE
Messages	<a href="http://host/msg/queue/qname/">http://host/msg/queue/qname/</a> <a href="http://host/msg/topic/topic_path/">http://host/msg/topic/topic_path/</a>	MQGET w. browse	MQPUT	-	MQGET

# WMQ Explorer Enhancements

- Sets
  - QMs can be partitioned into sets within the Navigator
  - e.g. "Test", "Production"
- Security Configuration
  - Easy to define channel exits, userid/password configurations
  - Configured for each QM or for all QMs 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

# QM Sets

The screenshot displays the WebSphere MQ Explorer interface. On the left, a tree view shows the hierarchy: IBM WebSphere MQ > Queue Managers > Queue Managers. A context menu is open over the 'Queue Managers' folder, listing the following items:

- All
- Development Queue Managers** (highlighted with a dashed border)
- v7
- Production Queue Managers
- audi
- QMA

In the background, the 'Content' view shows a table titled 'Queue Managers Set Development Queue Managers [manual set]'. The table has the following columns:

Queue manager name	Command level	Queue manager status	Platform	Queue-sharing group
...	700	Running	...	...

# WMQ Explorer Preferences

The screenshot displays the 'Preferences (Filtered)' dialog box in WebSphere MQ Explorer. The 'User Identification' tab is active, showing the 'Enable default user identification' checkbox checked and the 'Userid' field set to 'user1'. In the left sidebar, the 'Client Connections' category is expanded, revealing a list of sub-categories: Security Exit, SSL Key Repositories, SSL Options, and User Identification. The 'User Identification' sub-category is highlighted with a grey background. A secondary dialog box is partially visible on the right, showing a 'Browse...' button and a text field.

## Some Performance Information

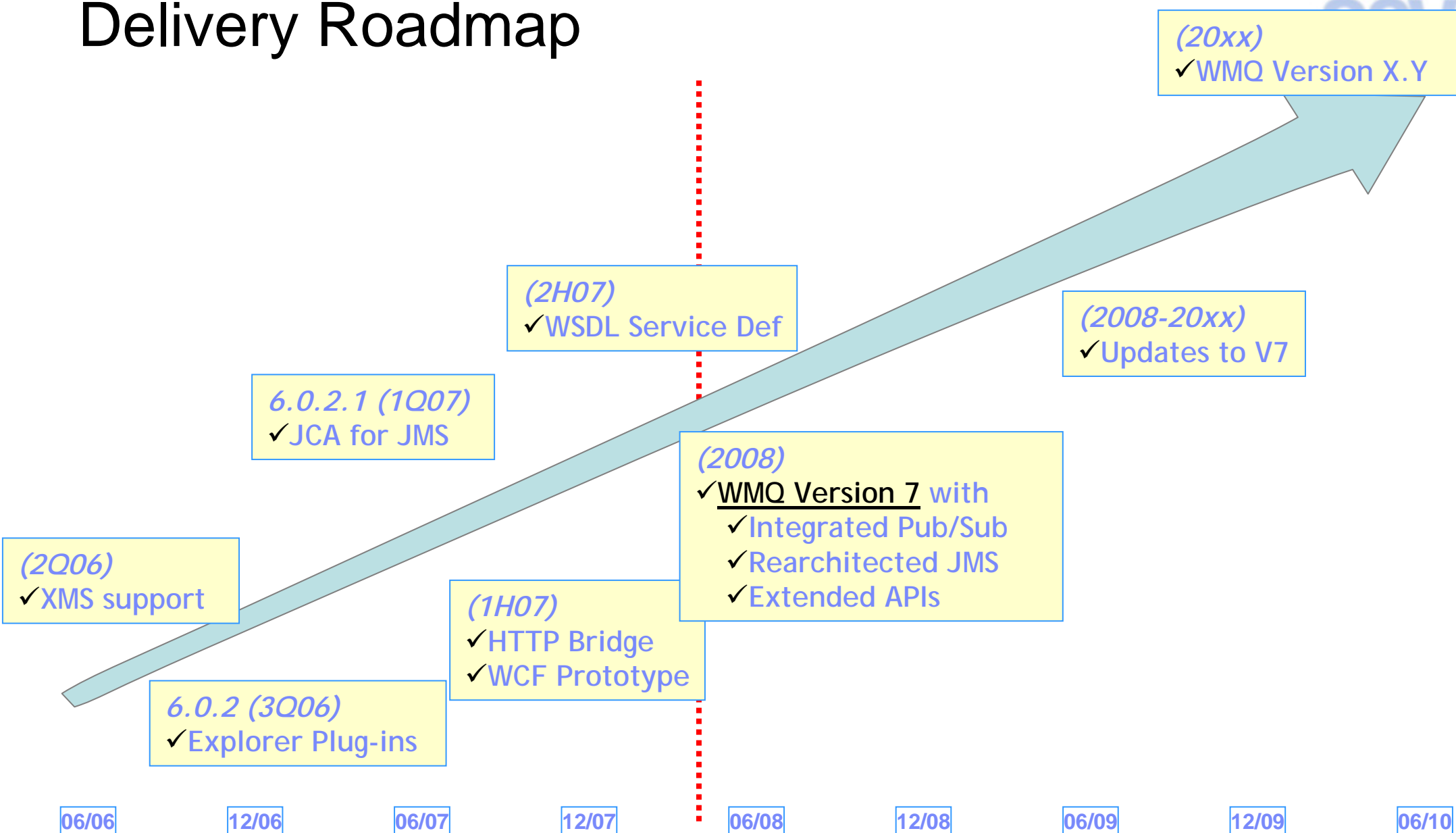
- Persistent pub/sub throughput increased up to 60%
- Non-persistent client throughput increased up to 300%
- JMS Selector rates improved up to 250%
- Message Listener throughput improved up to 45%
  - Latency also improved

**Note: Measurements are taken from pre-GA code & subject to change**

## Limitations in Initial Release

- A few items were not completed in time for GA on z/OS
  - Message Properties in CICS and IMS adapters
  - Using selectors for point-to-point messaging
- Intention is to release this function as soon as possible

# Delivery Roadmap



# Extending the Reach

# WMQ as an SOA Asset - Service Definition

- 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

# WCF (Indigo) Custom channel for MQ

- Windows Communication Foundation underpins .NET WS and Messaging
  - Built-in Transports e.g. MSMQ, HTTP(S), Named Pipes, TCP/IP, etc.
  - Transports can be extended with 'custom channels'
  - Allows alternative transports (like MQ) to be slotted into WCF seamlessly
- IBM has a prototype custom channel for WMQ
  - Initially shipped in 2007 on AlphaWorks & recently refreshed
  - Using SOAP/JMS message formats
    - For interoperability with WAS, CICS SOAP/JMS services
- We are looking for feedback to assess demand for this feature  
<http://www.alphaworks.ibm.com/tech/mqwcf>

# WebSphere MQ Version 7 - Summary

- Improvements to publish/subscribe
- Improvements to JMS layer
- Ease-of-use for MQI programmers
- Ease-of-use for administrators
- Performance
- Continues to extend the enterprise messaging foundation in the SOA world