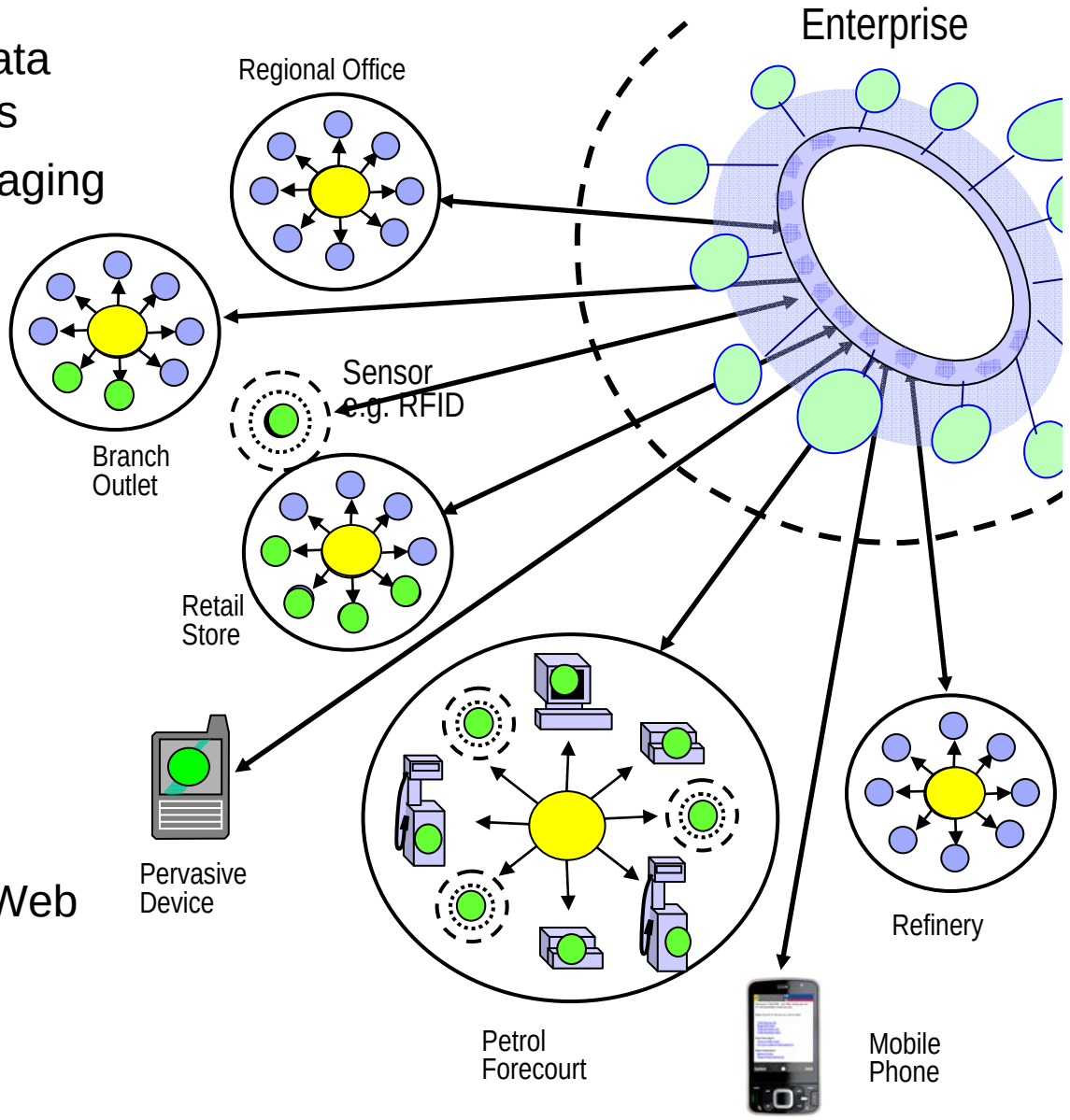
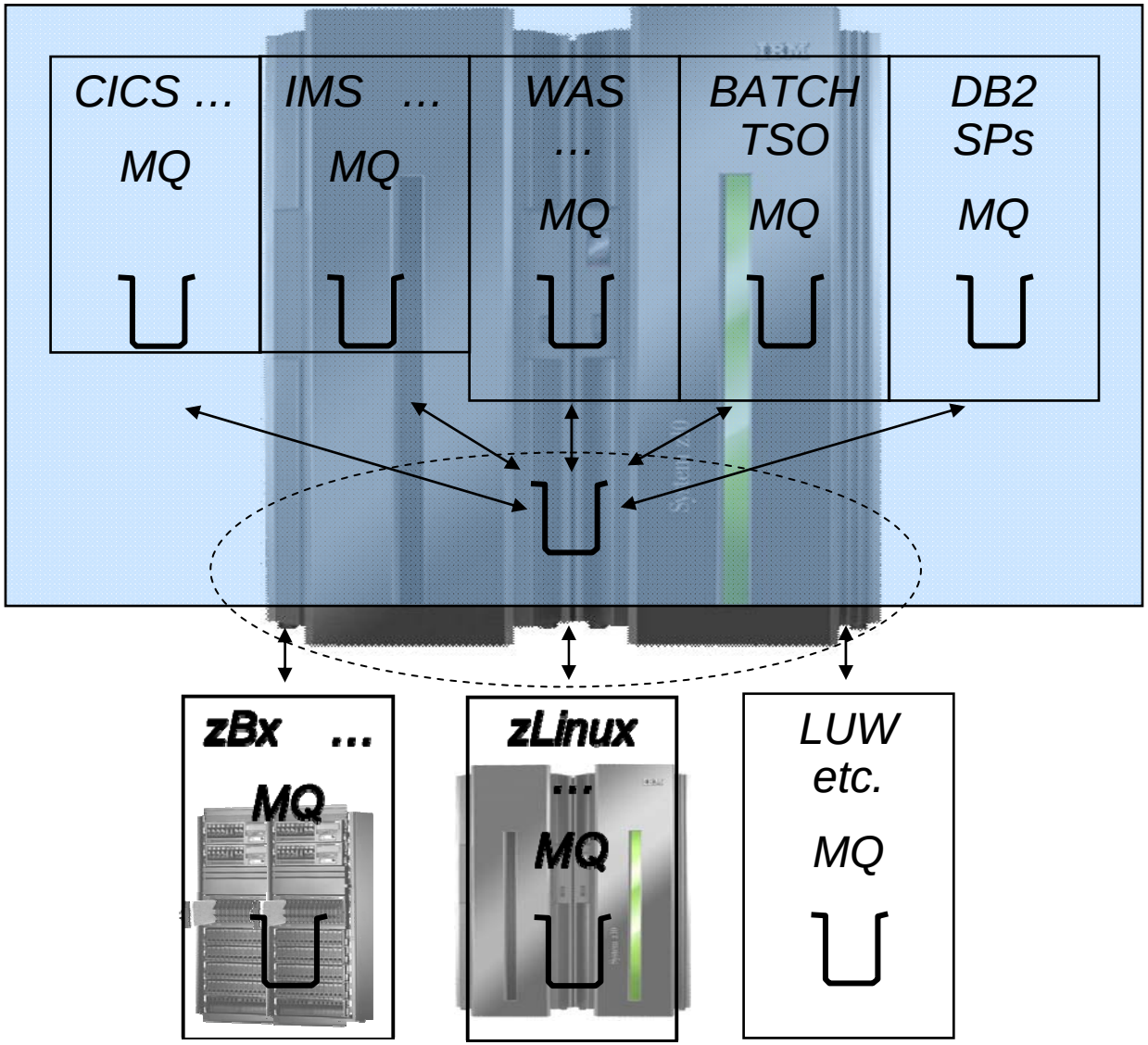
WebSphere MQ V7.1: Universal Messaging

WebSphere MQ Value: Connectivity to, from and within an Enterprise

- A Universal Message Bus for access to data wherever it exists to support your business
- Provides a comprehensive range of Messaging capabilities to support your Business requirements for data integration
  - Managed File Transfer
  - Messaging integration patterns
  - Reliability and availability QoS
  - SOA foundation
- Provides appropriate data access and data privacy controls to help meet audit and regulatory requirements
- WMQ Telemetry is one step in extending the reach of WMQ to a wider world of data relevant to your business
- Recent technology demonstration of MQ Web Messaging using HTML5 WebSockets continues this progress



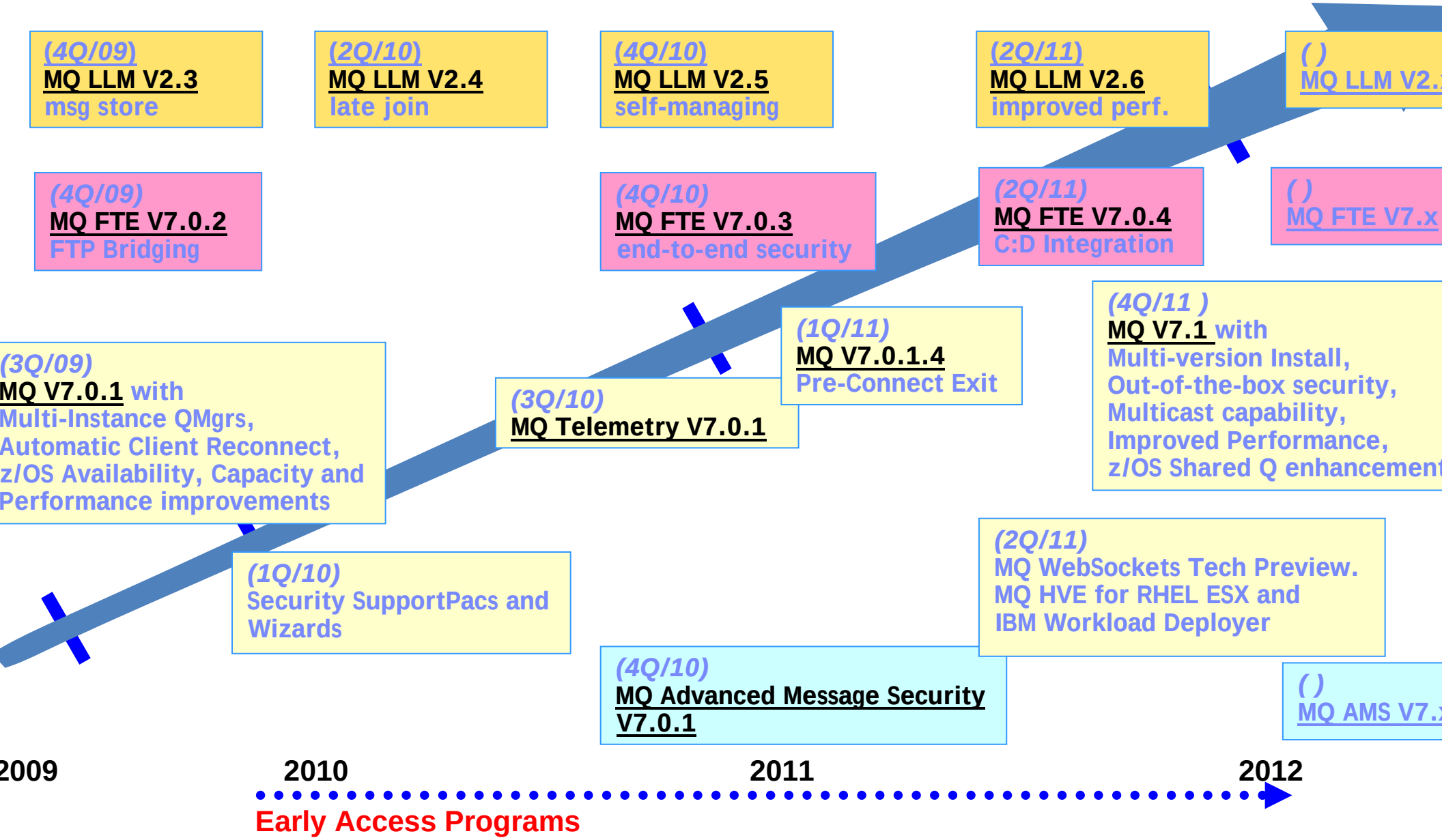
Connectivity to, from and within zEnterprise



Sysplex Shared Queue Message Availability:
Goal is to provide as near as possible continuous message data access under ALL failure scenarios (These scenarios include Application/Transaction failures, Application Execution Env. failures Qmgr failures, CF failures, DASD failures, Network failures, CEC failures)

Sysplex Shared Queue Message Capacity:
Goal is to provide Terabytes of affordable message capacity such that MQ is capable of meeting all business requirements for reliable message storage when processing applications are unable to run for whatever reason

WMQ Family Roadmap – continual delivery of customer value



WebSphere MQ V7.1

WebSphere MQ V7.1: Feature Summary

WebSphere MQ V7.1
Announced: 4 October 2011
Availability: 11 November 2011

Simplification

New Feature	*Benefits*	
Multi-Version Install capability on Distributed platforms	Makes it easier to deploy and upgrade systems and stage version to version migration	Unix and Windows support for multiple versions of MQ V7.x (AND one copy of MQ V7.0.1) down to fixpack levels. Relocatable installation support. Applications can connect to any Qmgr
Enhanced Security	Simplified Configuration Enhanced Authorisation and Auditing	IP address Authorisation capability Additional crypto algorithms More granular authorisation for non-local queues Application Activity Reports
Cloud Support	Simplifies and support Cloud deployments	Additional HVE images
Enhanced Clustering	Improves ease-of-use	Authorisation on Cluster Q rather than XMIT Q on Dist. Platforms Bind-on-Group Support
Multicast capability	New messaging QoS provides low latency with high fan-out capability	MQ Pub/Sub Topic space can now map to multicast Group Addresses Provides direct interoperability with MQ LLM
Improved scalability and availability on z/OS	Further exploitation of z196 Customer control over CF storage use CF Connectivity Loss improvements	Code contention reduced to improve multi-processor linear scaling Use of MQ Datasets rather than DB2 significantly improves “large” message capability Structure rebuild capability for CF Connectivity Loss scenarios
Improved Performance on Dist platforms	Improved multiprocessor exploitation	Various code improvements

WebSphere MQ V7.1: Feature Summary

NOTES

- This page shows the highlights of the new release in one chart. The rest of this presentation goes into the details.
- A one-word summary of this summary is “simplification”: making it easier to own, run and work with MQ.

- One part of the MQ V7.0.1 rationale was to deliver new function via the service stream, without requiring a full new release and migration cycle. Lessons learned from that have fed into V7.1, which has been designed to be more capable and more flexible when adding function through this channel.
- These new functions can be optionally enabled. The default is that new function requires adminstrative action to enable it, so that there is no unasked-for change in behaviour when you install a fixpack

Multi-Version Installation

- MQ on Unix and Windows can install multiple levels on a system
  - Relocatable to user-chosen directories
  - Can have multiple copies even at the same fixpack level
- Simplifies migration
  - Can move applications as needed, not all at once
  - No need for parallel hardware
- Easier for ISVs to imbed MQ in solutions
  - Can install in “private” locations without worrying about other copies
  - Reduces support concerns
- Permits a single copy of V7.0.1 to remain on system
  - So existing systems can be migrated
  - Must be 7.0.1.6 or later

Multi-Version Installation

NOTES

- With this release, you can install more than one copy of MQ on Unix and Windows platforms.
  - It is not available on System i
  - z/OS already has this capability, in a different form
- This will simplify migration strategies, as you can continue to use one version of MQ and only gradually migrate applications to a new version, without needing parallel hardware.
- When installing MQ you can choose the directory into which it will be installed. There is no longer a requirement to use /opt/mqm (Linux, most Unix) or /usr/mqm (AIX).
- Third party applications can imbed MQ under their own private directory if they wish, and can choose which versions of MQ they support, without worrying about the "visible" version of MQ that a user might be exploiting.
- You can leave an existing copy of MQ V7.0.1.6 (or later) on your systems, and this new feature will work alongside it. So you do not need to move to V7.1 before starting to exploit the multiple installation capabilities.

Multi-Version Installation: Concepts

- Main concept is an installation
  - Refers to the directory containing the binaries from a particular version of MQ
  - Can have a descriptive name

- One installation can be designated as primary
  - Required on Windows where some OS-specific elements have to be registered
  - Optional on Unix, creates symlinks to commands and libraries in /usr
    - Not created by default so your PATH will not always find MQ commands

- Queue Managers are owned by a specific installation
  - Governs the level of function available when the queue manager is running
  - Ownership can be changed to a newer installation for migration

Multi-Version Installation: Concepts

NOTES

- An installation is the collection of binaries that make up the MQ runtime, its commands a libraries. The precise details vary by platform, but conceptually you first define an installation, optionally giving it a name and description. You then run the install program, which will copy the code from media onto the disk in your chosen directories.
- Generally, there are no “default” paths created to an installation. However a primary installation does insert itself into default locations. You can only have a single primary installation on a system.
- On Windows, one installation is always denoted as the primary. This is necessary because some OS functions can only deal with a single location, for example how DLLs are registered for MSCS.
- On Unix, you do not require a primary installation unless you want to recreate links from /usr/bin and /usr/lib to the MQ commands and libraries in that installation.
- If you still have V7.0.1 on your system, that will always be the primary on all platforms.
- Each queue manager is considered as being owned by an installation. This governs the level of function available when running a queue manager. When you install a newer level of code, the queue manager can be moved to being owned by that newer installation, and hence new function can be used. Ownership is not changed automatically; you must execute the new setmqm command. Queue manager control commands such as strmqm must be issued from the directory associated with that owning installation

Multi-Version Installation: Application Impacts

- Existing applications “know” where the MQ libraries are
  - Embedded path or PATH/LIBPATH/LD_LIBRARY_PATH
  - Has always been a fixed location on Unix
- When MQ libraries move, apps will need to know where the new location is
  - /usr cannot be assumed
- New application libraries able to connect to any version of queue manager
  - Libraries such as libmqm, libmqic etc redesigned
  - Dynamically loading dependent libraries associated with the corresponding qmgr
  - If your app can find one V7.1 libmqm, it can connect to any qmgr, including future versions

- **MIGRATION NOTE:** Exits that invoke the MQI will need to be updated
  - Such as API Exits
  - Do not want exits to pull in different libraries than main application
  - Extended interface provides pointers instead for invoking MQI

Multi-Version Installation: Administration Impacts

- **MIGRATION NOTE:** Commands not in default PATH for Unix
  - Unless you have a primary install, all the control commands need explicit path
  - “ksh: crtmqm: not found” will be typical error message
- The dspmq, dspmqver & dspmqinst commands work for all installations
  - Shows installation details; which queue managers exist and owning installations
  - Other administration operations require PATH to point at install-specific commands
    - Get an error if you try to administer a queue manager the command does not own
- Installation details held box-wide
  - /etc/opt/mqm/mqinst.ini or Windows registry
- Default data paths unchanged
  - Still have /var/mqm/mqs.ini on Unix
  - On Windows, many other registry items are changed or removed
    - Now have qm.ini text files with the same content

Administration Examples

```
$ /usr/mqm/bin/dspmqver -i
Name:        WebSphere MQ
Version:     7.1.0.0
Level:       p000-L110915
BuildType:   IKAP - (Production)
Platform:    WebSphere MQ for AIX
Mode:        64-bit
O/S:         AIX 6.1
InstName:    Installation1
InstPath:    /usr/mqm
InstDesc: My default installation
DataPath:    /var/mqm
Primary:     Yes
MaxCmdLevel: 710

Name:        WebSphere MQ
Version:     7.1.0.0
InstName:    Installation2
InstPath:    /usr/mqm2/usr/mqm
InstDesc:    A second installation
Primary:     No
```

```
$ dspmq -o installation
QMNAME(V71A)
        INSTNAME(Installation1)
        INSTPATH(/usr/mqm)
        INSTVER(7.1.0.0)
QMNAME(V71B)
        INSTNAME(Installation1)
        INSTPATH(/usr/mqm)
        INSTVER(7.1.0.0)
QMNAME(INST2QM)
        INSTNAME(Installation2)
        INSTPATH(/usr/mqm2/usr/mqm)
        INSTVER(7.1.0.0)
```

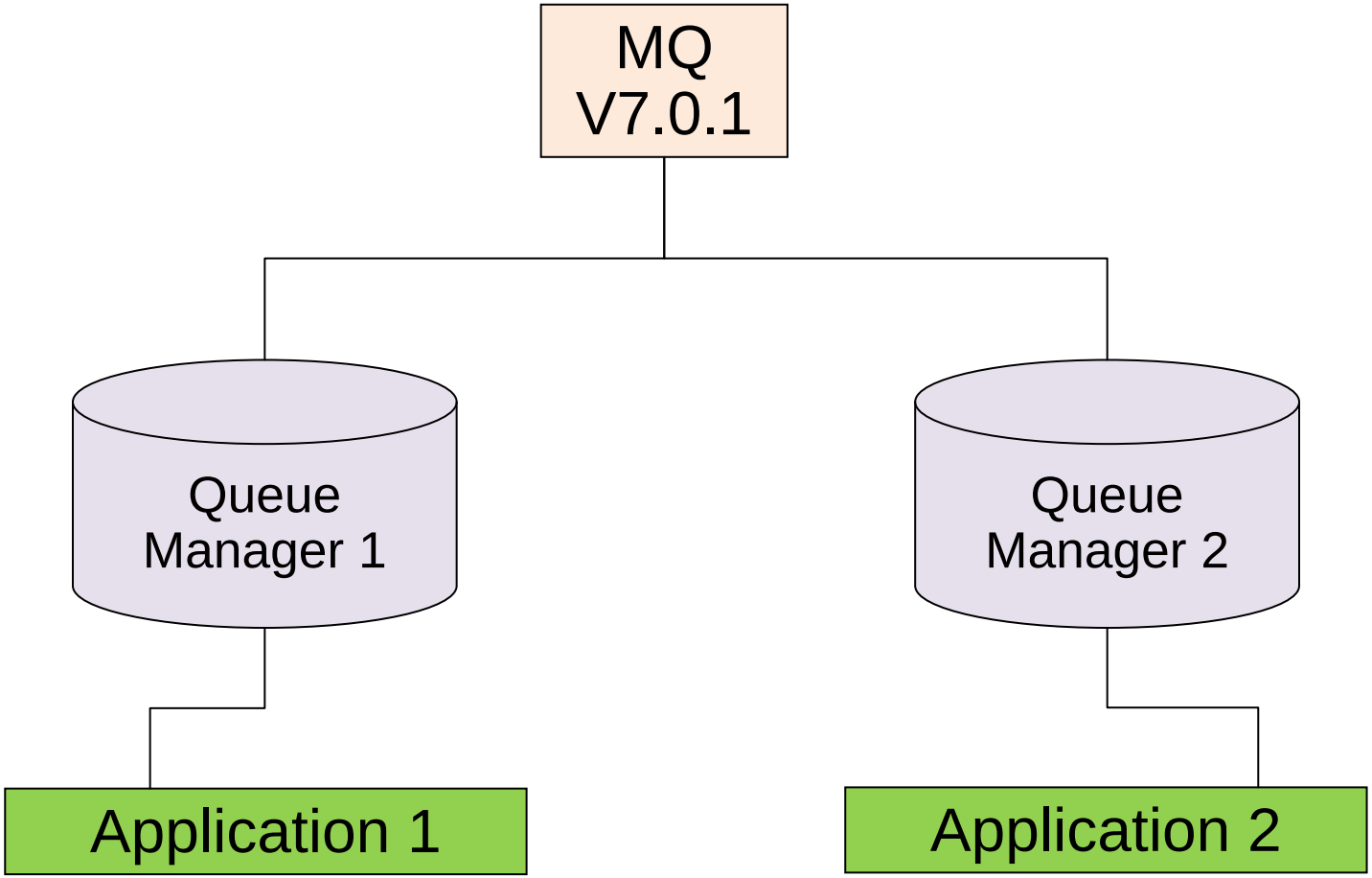
```
$ /usr/mqm/bin/endmqm INST2QM
AMQ5691: Queue manager 'INST2QM' is
associated with a different installation.
```

Administration Examples

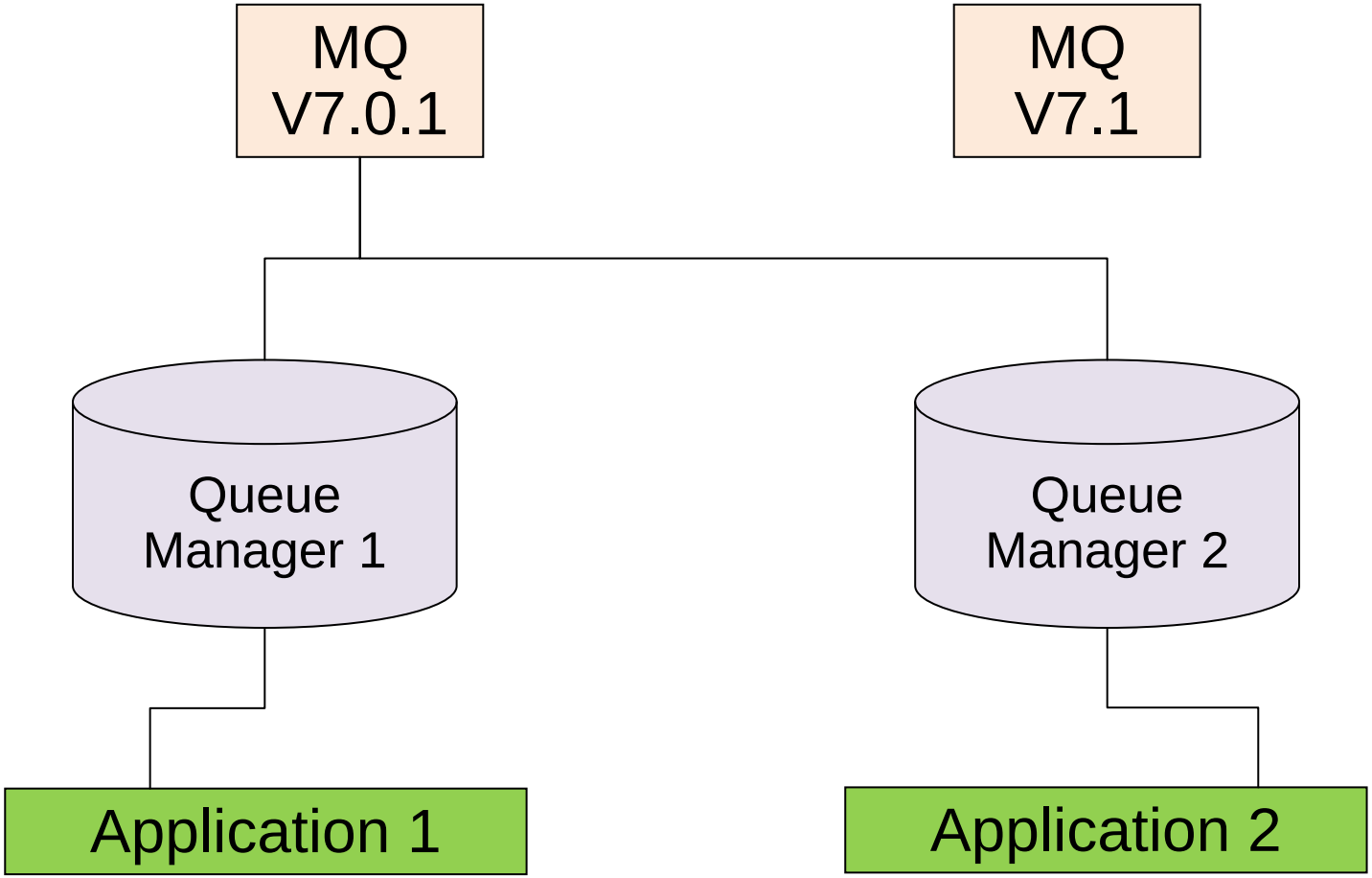
NOTES

- Here we see some of the commands that assist with installation management.
- The dspmq and dspmqver commands know enough about the system to show all of the installations, their directories, and the associated queue managers
- However queue manager control commands must be issued from the correct directory; they will return an error if you try to run them against a queue manager that is not associated with that installation.
- Tools that inspect the Windows registry or update it explicitly may need to be changed as most of the information that was previously in there now moves to flat-text ini files like the Unix platforms.

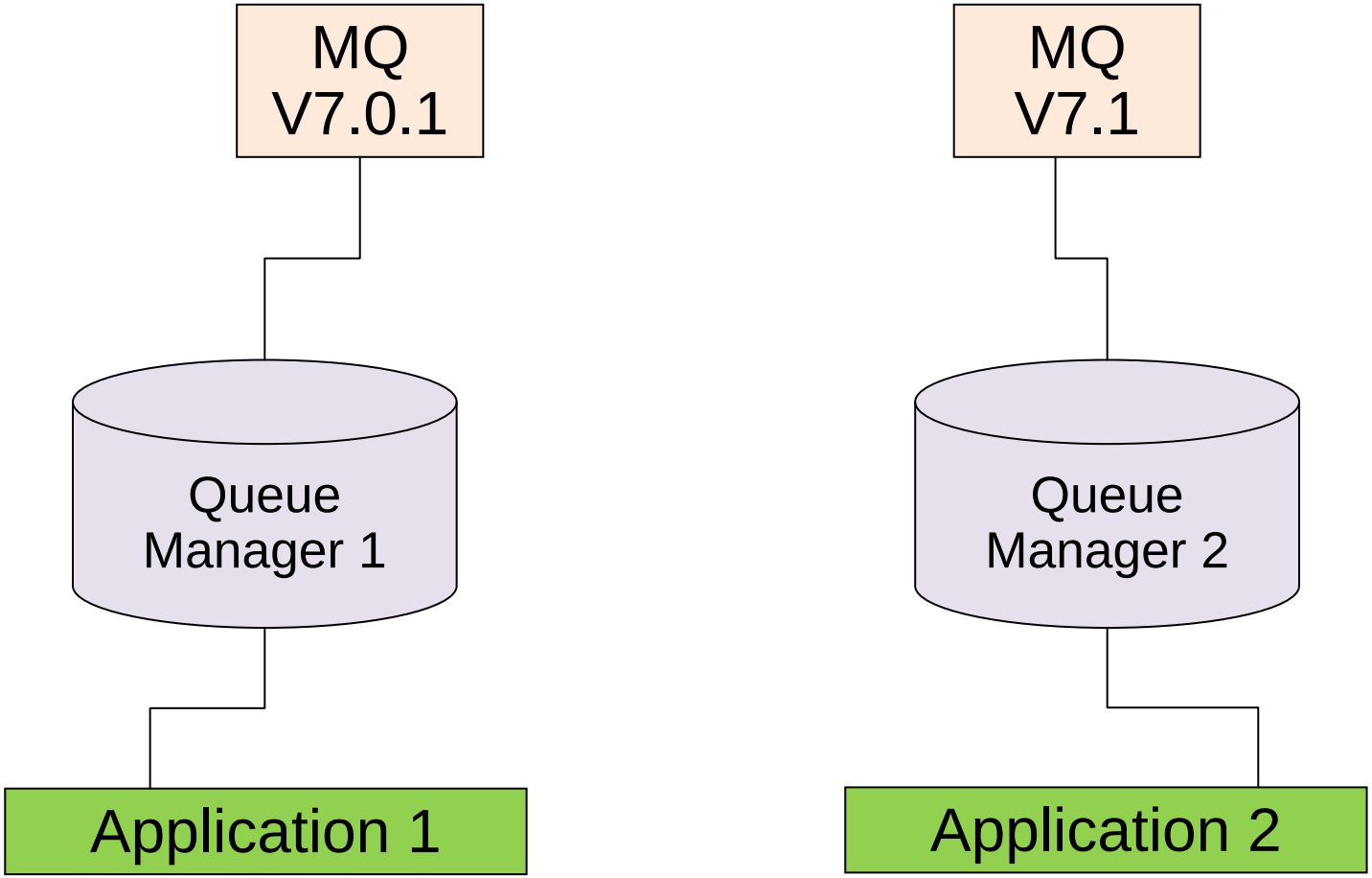
Application Migration: (1) Identify Applications to Migrate



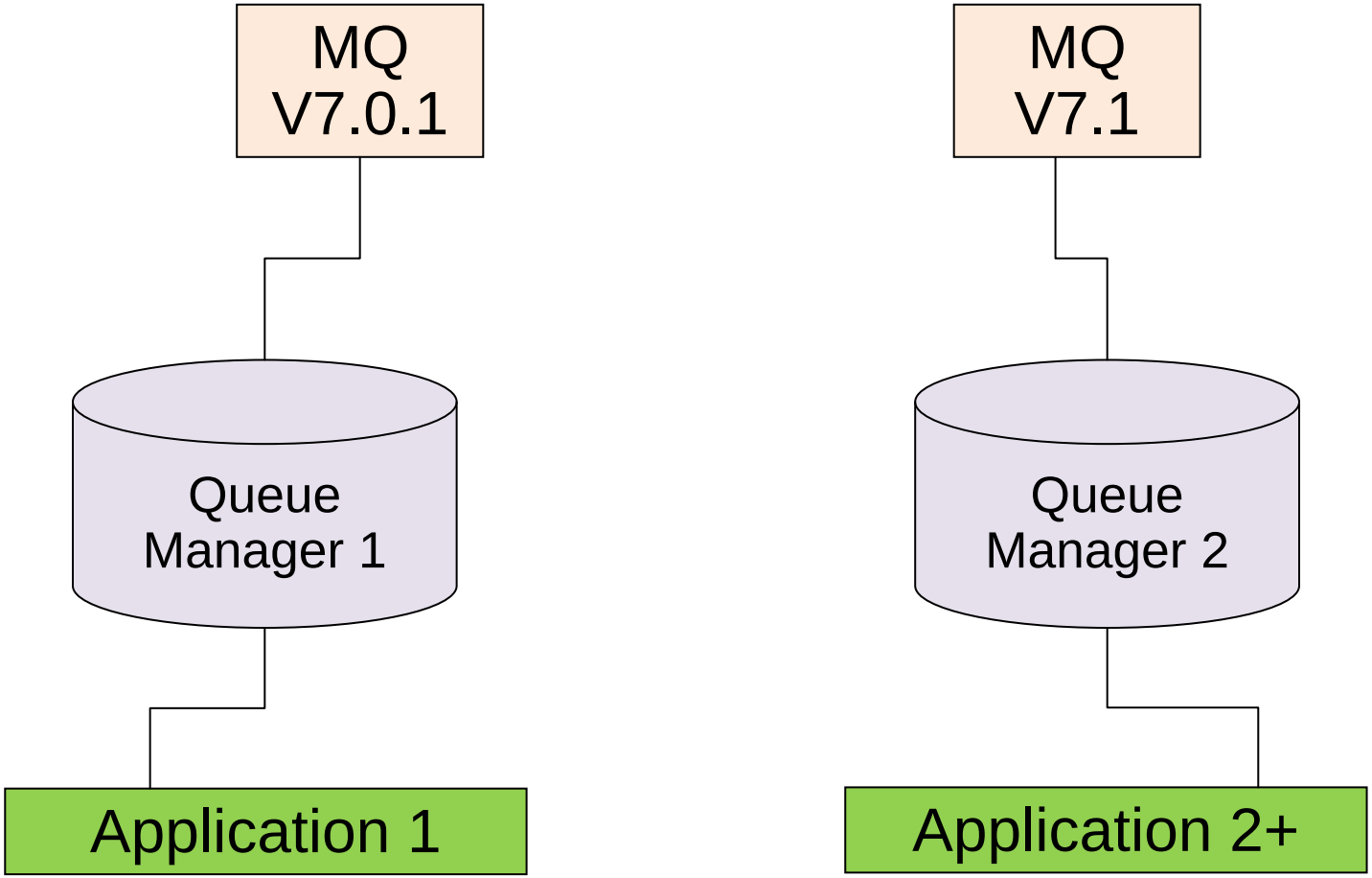
Application Migration: (2) Install V7.1 alongside V7.0.1



Application Migration: (3) Associate V7.1 code with a QMgr



Application Migration: (4) Modify App to Exploit New Features



Security: Channel Access Control

- Simplifying configuration for channel access
  - Clients and queue managers
- CHLAUTH definitions control who can use channels
  - Name mapping
  - Access blocking
- **MIGRATION NOTE**: Standard rules block clients on new queue managers
  - “Secure by default”
  - Migrated queue managers behave as before until you enable the rules
  - Queue manager attribute CHLAUTH(ENABLED|DISABLED) provides overall control

Security: Channel Access Control

NOTES

- Over the years there have been many requirements raised to make it simpler to block unwanted access to a queue manager. For example, only defined IP addresses should be allowed through.
- With V7.1 many of these rules can now be defined directly to the queue manager and channels.
- A standard set of rules are created when you create a new queue manager or migrate an existing one to run under V7.1. However, the rules only start to be used when you ENABLE them – a migrated queue manager has them DISABLED by default, so as to not disrupt any existing configuration and applications. The default rules block most client access; don't be surprised to get authorisation failures until you have reviewed the rules. The default rules were chosen to make new queue managers automatically more secure, simplifying the process of securing a system.

Security: Channel Access Control

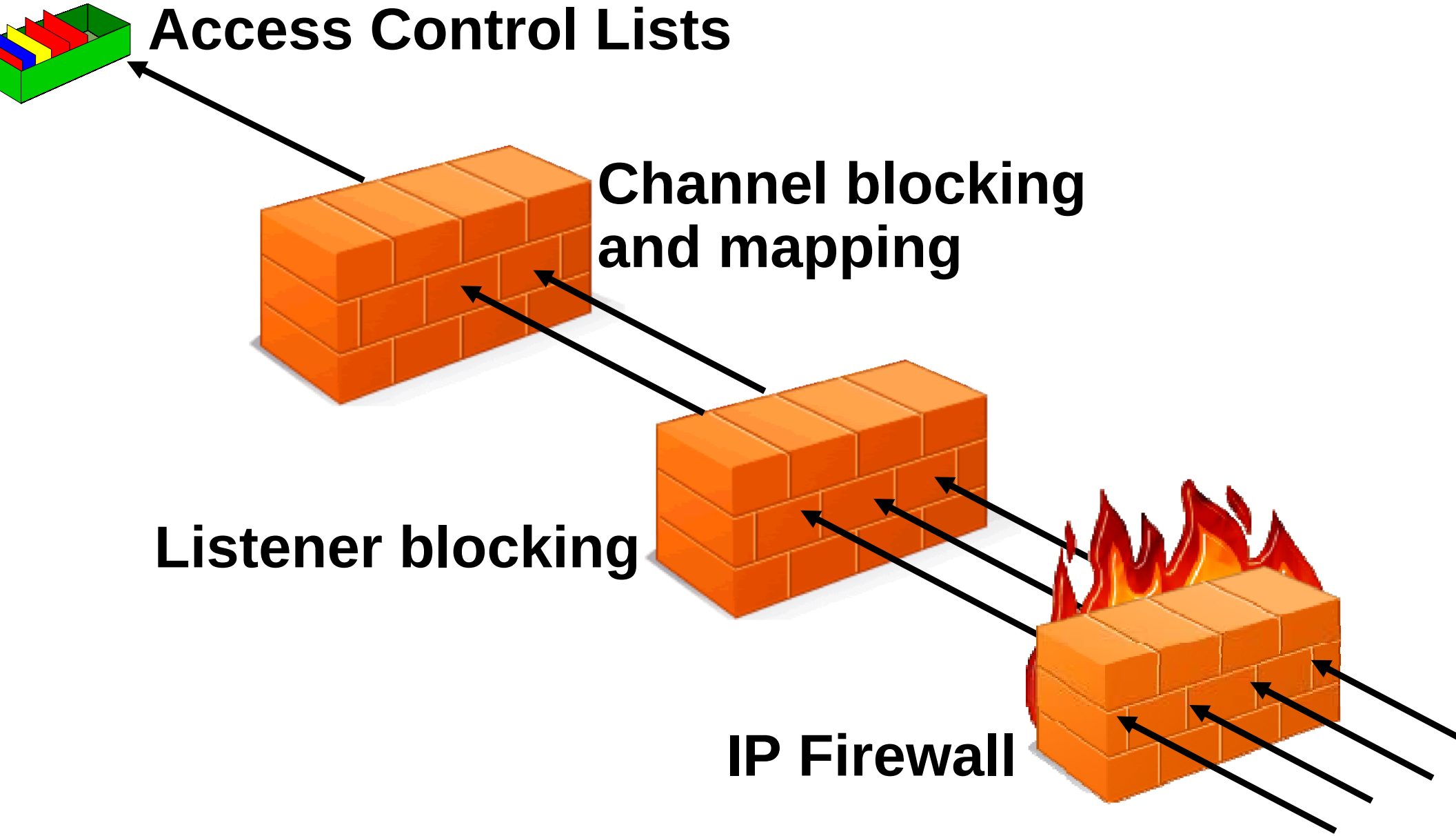
- Rules are based on
  - Partner IP address
  - Partner Queue Manager name
  - SSL Distinguished Name
  - Asserted identity (including *MQADMIN option)
  - Derived identity from DN mapping
- Easy to test rules that you define
  - DISPLAY CHLAUTH can “execute” rules
- Rules can be applied in WARNING mode
  - Not actually blocked, but errors generated
- MQ Explorer configuration via wizard

Security: Channel Access Control

NOTES

- Channel Auth records define the rules that are applied when a queue manager or client attempts to connect through a channel. A number of elements about the partner can be checked, and choices made about whether or not to allow the connection to proceed.
- A pseudo-userid (*MQADMIN) can be used in these rules, which covers the use of any id that would otherwise gain automatic administrative rights over a queue manager. For example, is the asserted identity in the mqm group or blank (so would inherit the channel's userid). Having a pseudo-userid makes it easier to have the same rules on all platforms, where the actual definition of who is an administrator might vary.
- The DISPLAY CHLAUTH command can be used with the MATCH(RUNCHECK) option to verify a simulated connection (pretending to have an address or id). This means you can test rules from the console without making a real connection.
- Rules can also be defined as "WARN", causing authorisation events to be generated, but not actually blocking the connection. This may assist in migrating to a secure environment, by not turning off connections immediately.
- To further simplify setting up these rules, the MQ Explorer has a Wizard to take you through the steps; this is shown later.

Channel Access Blocking Points



Channel Access Blocking Points

NOTES

- In this picture we illustrate that there are a number of points that an inbound connection must get through in order to actually make use of an MQ queue. We remind you that your IP firewall is included in this set of blocking points, should not be forgotten, and is not superseded by this feature in MQ.

Blocking at the Listener

- Single list of IP address patterns
- NOT A REPLACEMENT FOR AN IP FIREWALL
  - Temporary blocking
  - Blocking until IP firewall updated
  - Shouldn't be many entries in the list
- Blocked before any data read from the socket
  - i.e. before SSL Handshake
  - Before channel name or userid is known
- Avoiding DoS attack
  - Really the place of the IP firewall
  - Simplistic 'hold' of inbound connection to avoid reconnect busy loop
- Network Pingers if blocked don't raise an alert
  - Immediate close of socket with no data not considered a threat

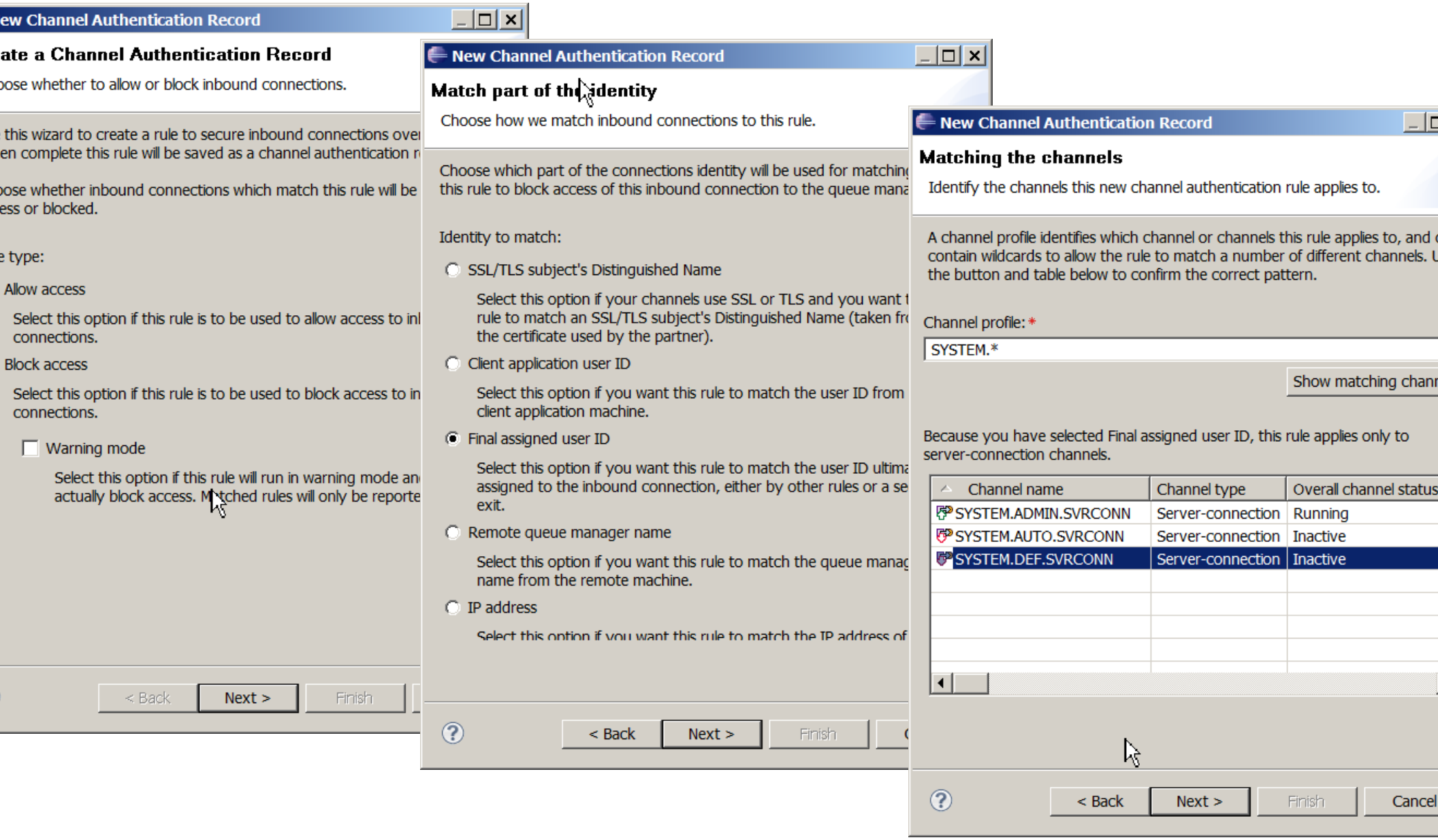
SET CHLAUTH(*) TYPE(BLOCKADDR) ADDRLIST('9.20.*', '192.168.2.10')

Blocking at the Listener

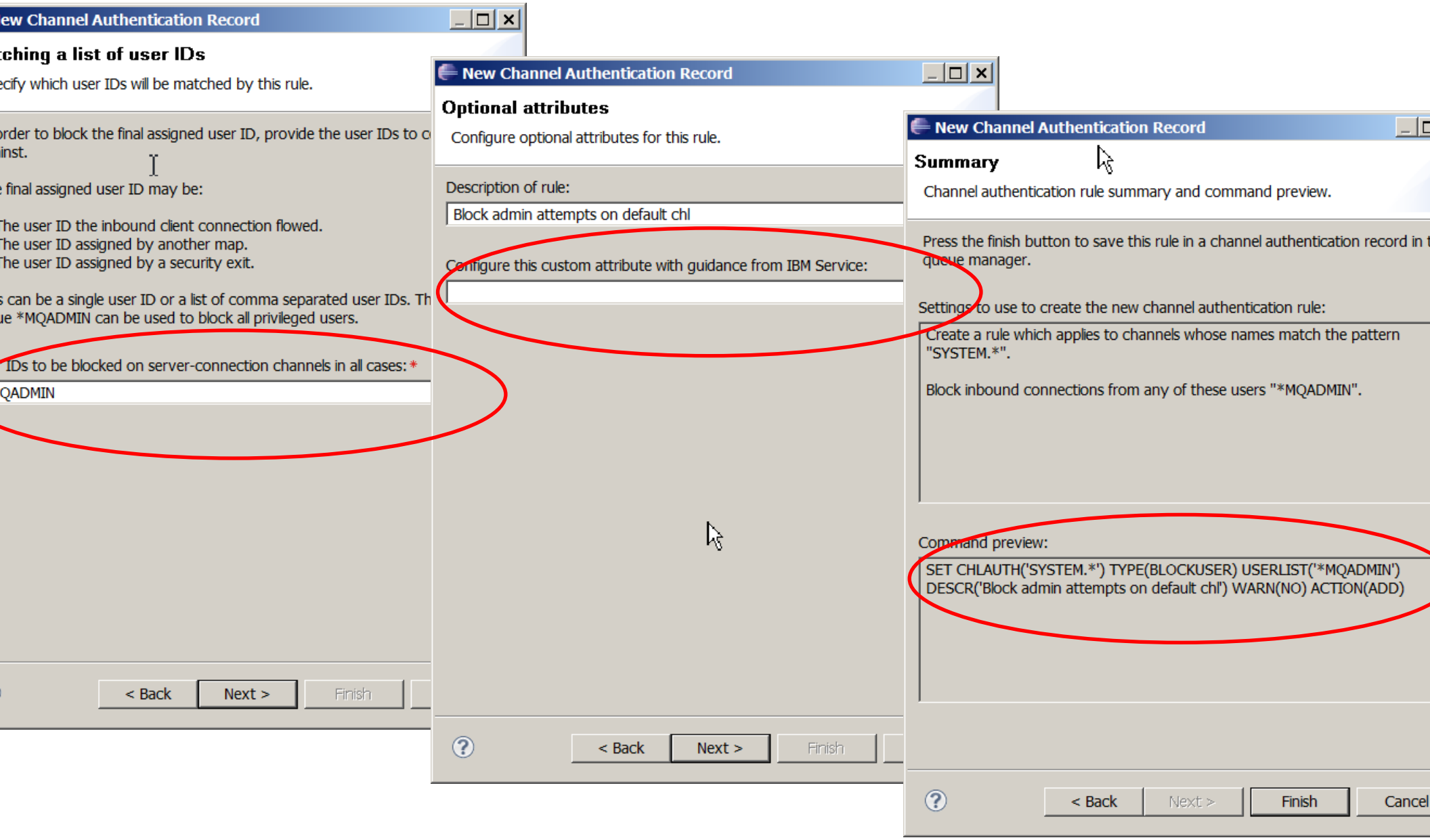
NOTES

- There is a list of IP addresses that the listener configuration will have access to. If any of these IP addresses attempts to start an inbound channel connection, the listener will bounce the connection prior to starting a channel instance to process any data, for example SSL Handshake, that might need to be read before knowing what the channel name is. If the queue manager is not running it will still have access to the configuration and it will still block the specified IP addresses.
- The default setting of this configuration information on the listener will be an empty list. We don't have enough knowledge to be able to sensibly populate this with anything.
- A Denial of Service (DoS) attack on the listener, whilst really the place of the firewall to deal with, would mean high CPU in the listener if it had to deal with a repeated connection from a client. This and the fact that we would like to quietly ignore network pingers if they don't send any data and only raise blocking events on them if they do send data, means that the listener will hold any sockets that come from blocked IP address open for a period of time prior to closing the socket, rather than immediately rejecting it. This will stall the repetitiveness of the attack and protect the listener process allowing it some time to process real requests, and additionally give the network pingers time to close the socket before we do allowing us to detect we don't need to emit an event for that case. By default this time is 30 seconds.

Channel Blocking and Mapping from the Explorer



Channel Blocking and Mapping from the Explorer



Channel Blocking and Mapping in the Explorer

NOTES

- We see here a walkthrough of the Explorer Wizard that creates Channel Authentication records.
- In one of the panels you can also see highlighted part of the design that will enable additional function to be delivered in the service stream.
- The final stage of the Wizard shows the actual command that has been created. You can copy this command and reuse it in scripts.
- Of course there are also PCF equivalents for these commands so you can issue these from your own programs too.

Channel Access Scenario (1)

SET CHLAUTH(*) TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)

"We must make sure our system is completely locked down"

Channel Access Scenario (2)

SET CHLAUTH(*) TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)

SET CHLAUTH(BPCHL.*) TYPE(SSLPEERMAP) SSLPEER('O=Bank of Shetland') MCAUSER(BANK123)

SET CHLAUTH(BPCHL.*) TYPE(SSLPEERMAP) SSLPEER('O=Bank of Orkney') MCAUSER(BANK456)

"Our Business Partners must all connect using SSL, so we will map their access from the certificate DNs"

Channel Access Scenario (3)

SET CHLAUTH(*) TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)

SET CHLAUTH(BPCHL.*) TYPE(SSLPEERMAP) SSLPEER('O=Bank of Shetland') MCAUSER(BANK123)

SET CHLAUTH(BPCHL.*) TYPE(SSLPEERMAP) SSLPEER('O=Bank of Orkney') MCAUSER(BANK456)

SET CHLAUTH(SYSTEM.ADMIN.SVRCONN) TYPE(ADDRESSMAP) ADDRESS('9.20.1-30.*') MCAUSER(ADMUSER)

"Our Administrators connect in using MQ Explorer, but don't use SSL. We will map their access by IP Address"

Channel Access Scenario (4)

SET CHLAUTH(*) TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)

SET CHLAUTH(BPCHL.*) TYPE(SSLPEERMAP) SSLPEER('O=Bank of Shetland') MCAUSER(BANK123)

SET CHLAUTH(BPCHL.*) TYPE(SSLPEERMAP) SSLPEER('O=Bank of Orkney') MCAUSER(BANK456)

SET CHLAUTH(SYSTEM.ADMIN.SVRCONN) TYPE(ADDRESSMAP) ADDRESS('9.20.1-30.*') MCAUSER(ADMUSER)

SET CHLAUTH(TO.CLUS.*) TYPE(QMGRMAP) QMNAME(CLUSQM*) MCAUSER(CLUSUSR) ADDRESS('9.30.*')

"Our internal cluster doesn't use SSL, but we must ensure only the correct queue managers can connect into the cluster"

Channel Access Scenario

NOTES

- Here is an example of how we expect this to be used.
- Our business requires that “We must make sure our system is completely locked down”. So we start off with a rule that blocks everyone. Therefore anyone that doesn’t match a more specific rule will not be allowed in.
- Our business requires that “Our Business Partners must all connect using SSL, so we will map their access from the certificate DNs”. So we have some rules that map specific DNs of our Business Partners to specific user IDs. Previously you might have done this by having separate channel definitions for each BP, now if you wish they can come into the same receiver definition.
- Our business requires that “Our Administrators connect in using MQ Explorer, but don’t use SSL. We will map their access by IP Address”. So we have a rule that gives them all a single administrative access user ID based on a range of IP addresses.
- Our business requires that “Our internal cluster doesn’t use SSL, but we must ensure only the correct queue managers can connect into the cluster”. So we have a rule that gives access to the correctly named queue managers but only if they come from a recognised IP address.

Security: SSL

- More crypto algorithms supported for SSL
  - Stronger algorithms are now available and recommended
  - MQ V7.0.1 added some SHA-2
  - MQ V7.1 adds more, with support for the NSA “Suite B” standard which includes Elliptic Curve cryptography
- Some older algorithms (eg SHA-1) should be considered deprecated
  - No plans to withdraw older algorithms immediately
  - But expect them to be removed in a future version of MQ
- Newer algorithms supported by gskit8 on Distributed platforms
  - Waiting for z/OS and iSeries SSL implementations before MQ can support them there
- The gskit toolkit is now provided inside the MQ installation
  - Will not clash with alternative levels from other MQ installations or other products

Security: SSL

NOTES

- MQ V7.1 extends the support for newer cryptographic algorithms, including from the SHA-2 family, and the NSA Suite B set.
- NIST has stated that SHA-1 should be considered deprecated. While MQ has not removed these algorithms, it may be done in future versions.
- Like the earlier FIPS-140 option within MQ, you can choose to enforce Suite B compliance on your channels.
- Note that these algorithms are currently only available where gskit is used as the SSL provider. For Java (which uses JSSE), z/OS and System i, MQ makes use of external toolkits and is dependent on those products to support the algorithms first.

Security: Authorisations for Non-Local (Clustered) Queues

- Distributed platforms now have authorisations for non-local queues
  - Including clustered queues
  - Making it consistent with z/OS
  - Also consistent with Topic authorisations
- So there is no longer a need to authorise access to the cluster transmit queue
- Grant authorisation to the remote queue manager instead
  - A new pseudo-object known to the OAM
- Also works for applications which explicitly open queue@qmgr
  - Common pattern when using ReplyTo information

```
setmqaut –m QM1 –t queue –n SYSTEM.CLUSTER.TRANSMIT.QUEUE -p mquser +put
                                    BECOMES
setmqaut –m QM1 –t rqmname –n QM2 –p mquser +put
```

Security: Authorisations for Remote Queues

NOTES

- Access to non-local queues can be authorised at a more granular level than previously. This new function matches something that was already available on z/OS and for Topics on all platforms.
- An object does not need to exist in order to associate ACLs with it
- A new pseudo-object, the “remote queue manager” is known by the OAM, and authorities are applied to it instead of the transmission queue that will be used to send the message to the remote queue. This means that the cluster transmission queue no longer has to be accesible to anyone wanting to use clustered queues, and the common pattern of using the ReplyToQueue/ReplyToQueueManager can also be managed in a more granular and explicit fashion.
- This does not remove the need for controlling access at the receiving queue manager as well (for example by using PUTAUT(CTX)) but it makes it easier to manage the sending side.
- The next page shows some before/after commands

- This function can be disabled by a switch in the ini file. It is not a queue manager attribute as we expect customers to migrate to the new model once and then not revert, so it does not need to be as dynamic as real attributes can be.

Security: Authorisations for Remote Queues (Examples)

NOTES

- Consider a configuration where QM1 and QM2 both host a clustered queue, Q1

- Examples showing compatibility/original behaviour

```
setmqaut –m QM1 –t queue –n QM2    –p mquser +put
setmqaut –m QM1 –t queue –n TO.QM2 –p mquser +put
setmqaut –m QM1 –t queue –n Q1     -p mquser +put
setmqaut –m QM1 –t queue –n SYSTEM.CLUSTER.TRANSMIT.QUEUE -p mquser +put
setmqaut –m QM1 –t queue –n SYSTEM.CLUSTER.TRANSMIT.QUEUE -p mquser +put
setmqaut –m QM1 –t qmgr            -p mquser +connect
```

- Examples using the new model

```
setmqaut –m QM1 –t rqmname –n QM2 –p mquser +put
setmqaut –m QM1 –t rqmname –n QM2 –p mquser +put
setmqaut –m QM1 –t queue   –n Q1  -p mquser +put
setmqaut –m QM1 –t queue   –n Q1  -p mquser +put
setmqaut –m QM1 –t rqmname –n QM2 –p mquser +put
setmqaut –m QM1 –t qmgr           -p mquser +connect
```

Application Activity Reports

- New set of events to report on MQI operations by applications
  - One PCF event may contain multiple MQI operations
- Configurable in granularity
  - Amount of data
  - Which applications
- Enables scenarios such as
  - Application audit trail
  - Message duplication
  - Resource usage: which queues or topics are actually being used
  - Problem Determination: most recent MQI calls by applications
  - Application Coding Standards: does everyone use the MQI in the recommended way
  - And more …
- On all Distributed platforms

Application Activity Reports

NOTES

- New for the Distributed platforms is the ability to report on all the MQI operations from an application. This is similar to an MQI trace, but differs in several respects.
  - It is running “inside” the queue manager so has access to more than just the MQI parameters passed by the application. There is some additional information reported, such as the real queues used by the application, not just the name passed to MQOPEN.
  - The output follows the same style as many other reports, in that it is a set of PCF events, where each event holds details about multiple MQI calls
- Applications and their relationships and resources can be analysed without inspecting the application source code.
- Like other events, it is possible to redefine the event queue to be a topic alias so multiple consumers can work with these messages.
- An ini file defines the required granularity – you can have reports of all message data for all applications, but that might be excessive. Changes to the ini file can be made dynamically without restarting the queue manager; just cycle the queue manager attribute that determines whether or not these reports are to be collected.
  - You can also permit applications to disable their own data collection
- Now these reports are generated, many interesting requirements can be met by analysing or using the data. A sample program (source included) formats these events and you can use this as the basis of more sophisticated tools.

Extract from Report

```
MonitoringType: MQI Activity Trace
QueueManager: 'V71'
Host Name: 'rockall.hursley.ibm.com'
CommandLevel: 710
ApplicationName: 'WebSphere MQ Client for Java'
ApplicationPid: 18612354
UserId: 'mquser'
ConnName: '9.20.95.106'
Channel Type: MQCHT_SVRCONN
Platform: MQPL_UNIX
==========================================================================
Time     Operation  CompCode MQRC  HObj (ObjName)

10:04:09 MQXF_INQ   MQCC_OK  0000  2
10:04:09 MQXF_CLOSE MQCC_OK  0000  2
10:04:09 MQXF_OPEN  MQCC_OK  0000  4 ()
10:04:09 MQXF_INQ   MQCC_OK  0000  4
10:04:09 MQXF_CLOSE MQCC_OK  0000  4
10:04:09 MQXF_OPEN  MQCC_OK  0000  4 (SYSTEM.DEFAULT.LOCAL.QUEUE)
10:04:09 MQXF_INQ   MQCC_OK  0000  4
```

Clustering

- "Bind on group"
  - All messages within a logical group are routed to the same queue manager
  - Workload balancing is done for each group
  - Simpler for applications that use message groups
  - Previously would have had to close and reopen the queue

- New option in the MQI and DEFBIND attribute for queues

- Once a group has started its path to a selected queue manager, messages in that group will not be reallocated in the event of a failure

MQ Clients

- A client is now available on System i enabling connectivity from C and RPG programs without needing a local queue manager
  - Platform already had a Java client
- MQI libraries like libmqm connect to local and remote queue managers
  - Smart switching for clients, as well as handling multi-version systems
- API Exits available in C clients
  - Same interface as available for local binding applications

MQ Clients

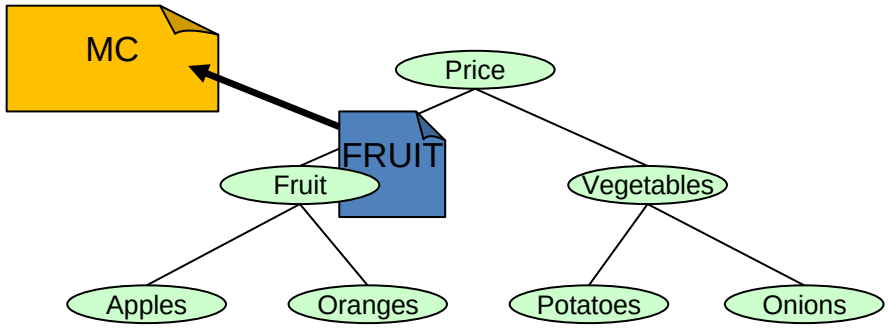
NOTES

- The System i platform gets a C-based client, like other Distributed platforms. This complements the Java client that is already available
- Part of the “smart” API switching libraries that are needed to support multi-version installations can now also handle the differences between local and client connections. This makes it simpler to develop new applications as you do not need to compile/link them differently for the different environments.
- API Exits are also now available for the C client libraries, matching the interfaces on the server side. There are some minor differences, such as how XA verbs are handled, but these should not affect existing exits.

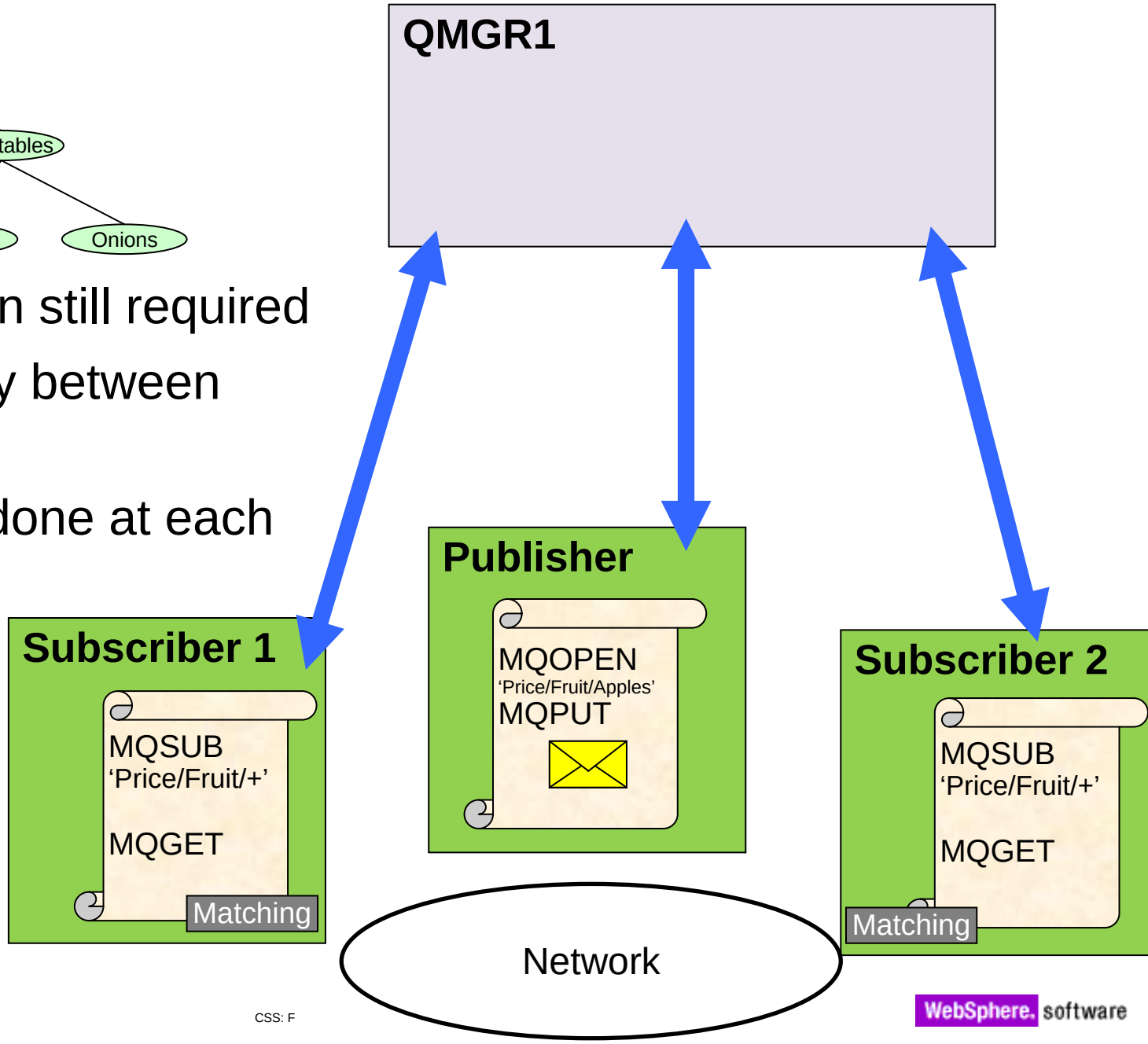
MQ Clients – Multicast

- Publish/Subscribe is enhanced to support multicast communication
  - Uses technology from the MQ Low Latency Messaging product
  - So it is interoperable with LLM
- Provides new Quality of Service
  - Low latency with high fan-out
  - Provides higher speeds for non-persistent messages
  - Provides higher availability as queue manager can be removed without affecting flow
  - Provides “fairness” as all recipients of a message get it at the same time
  - Higher scalability as additional subscribers cause no additional traffic
- Mapping MQ topic space to multicast group addresses
  - Can have mix of multicast and queue-based subscribers
  - Topic objects have associated COMMINFO objects to define addresses and other attributes
- Supports direct communication from publisher to subscriber, bypassing qmgr
- Queue manager maintains status and statistics for monitoring

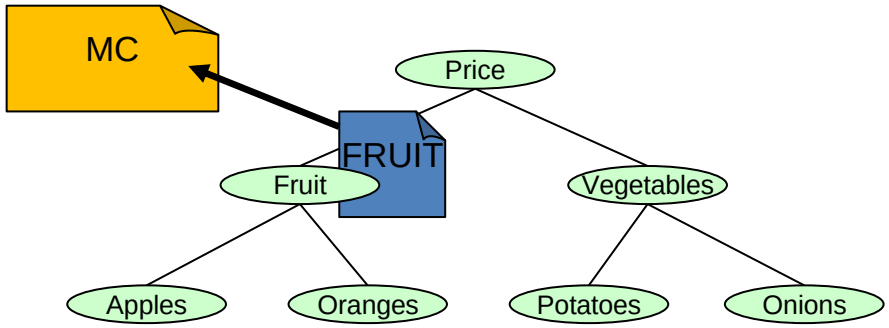
Multicast Publish/Subscribe



- Normal MQ connection still required
- Messages flow directly between clients
- Subscriber matching done at each client



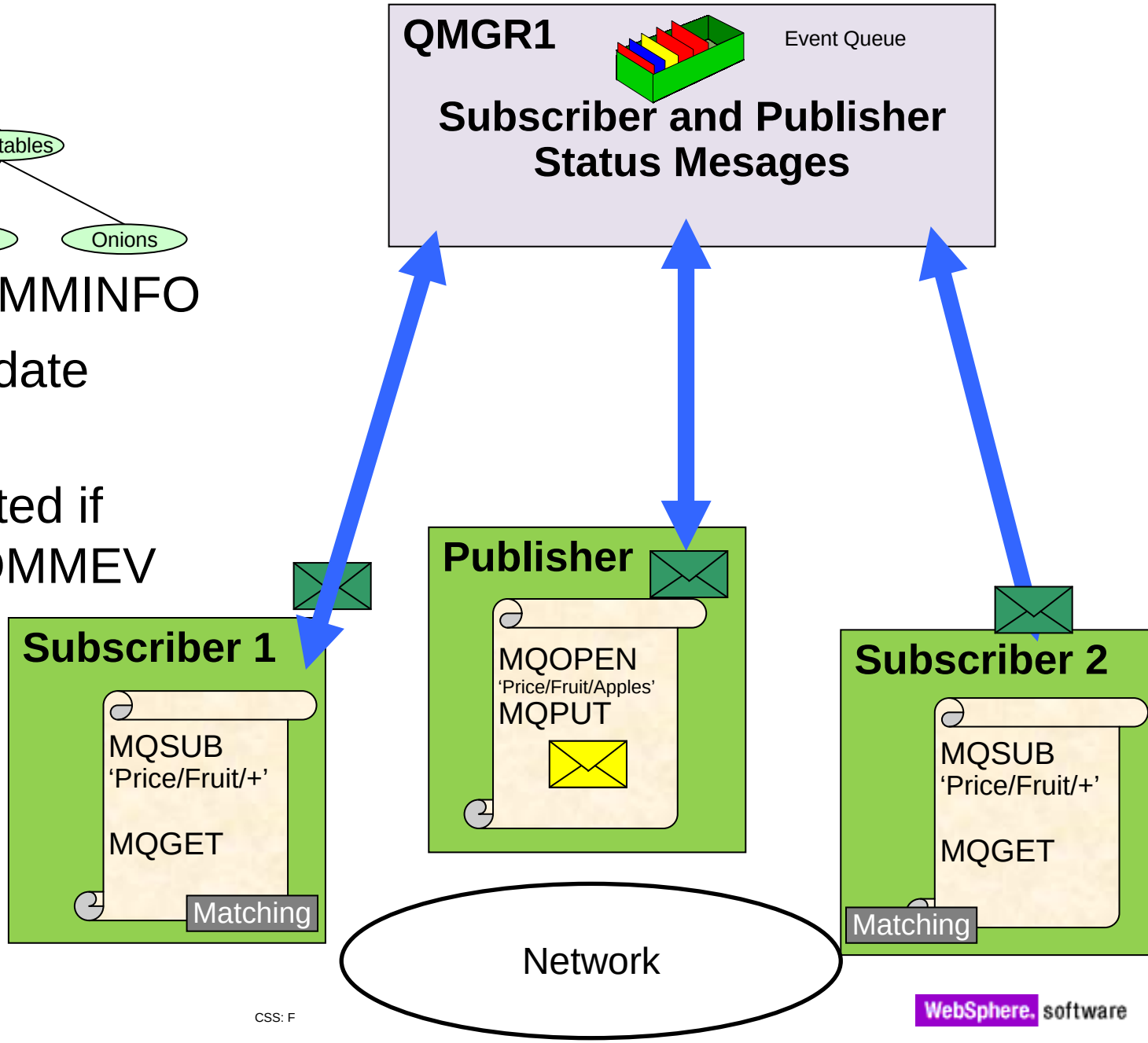
Multicast Monitoring



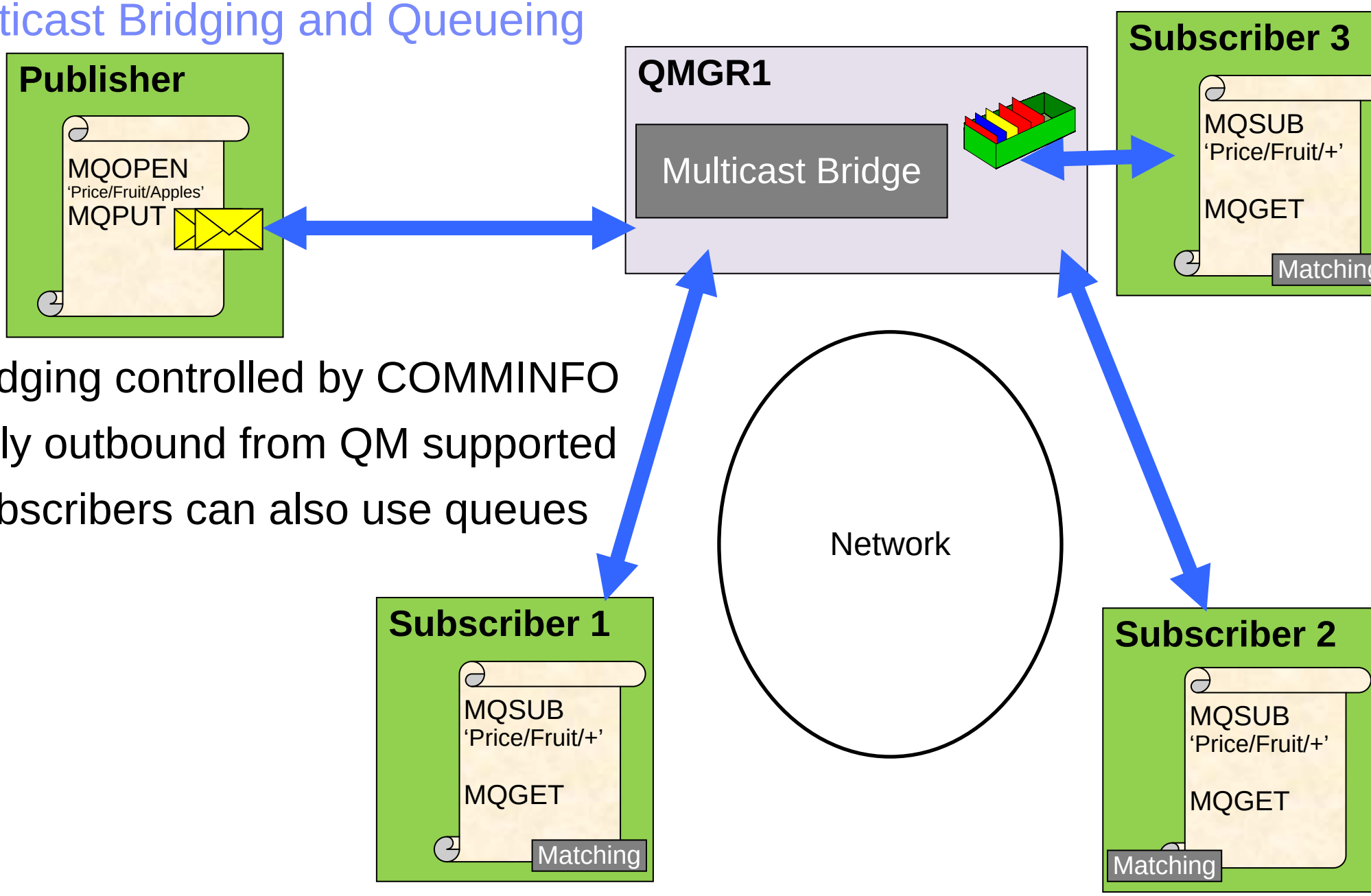
Monitor interval in COMMINFO

Monitor messages update TPSTATUS

Event messages created if required based on COMMEV



Multicast Bridging and Queueing



Bridging controlled by COMMINFO

Only outbound from QM supported

Subscribers can also use queues

Multicast MQI Support

NOTES

- Supported
  - Normal MQI
  - MQGET & MQCB
  - Message properties
  - MQSUB with selection by message property
  - Managed subscriptions
  - Data conversion (MQGMO_CONVERT)
- Limitations
  - C client only
  - Pub/Sub only
  - No transactionality or persistence
  - No durable subscriptions
  - No message grouping or segmentation
- Uses LLM technology but does not include all LLM features

Channels

- See the MQ version of connecting partner
  - Level of clients and queue managers available in channel status
  - For example a V7.0.0.1 client shows as RVERSION(07000001)
  - Can distinguish Java, C, .Net client programs
  - Helps administrator determine whether partner needs upgrading
- Distributed platforms now use DISCINT to disconnect idle clients
  - ClientIdle qm.ini parameter ignored
  - Consistent with z/OS
- Alternative channel batch control based on byte counts
  - BATCHLIM attribute
  - Useful when a transmission queue holds mix of large and small messages
  - Can make batch time (latency) more consistent
  - Batch is ended when first of either bytes or messages transferred reach configured limit

Channels

NOTES

- Some small but useful enhancements to channel controls
- The RVERSION and RPRODUCT values on channel status can tell an administrator what is connecting to a queue manager. The information has been sent between systems since V7.0, and is now exposed to users. Any client or queue manager that is at V6.0 or older will not send this data, so the lack of detail will indicate old systems.
- Both z/OS and Distributed platforms have ways of forcing a client to be disconnected when it has done no work for a while; with V7.1 those mechanisms are made consistent with use of the DISCINT channel attribute
- Traditionally, channels commit a batch after having sent 50 messages or when they reached an empty transmission queue (with modifications possible by BATCHINT). The amount of data that might be sent with 50 messages could vary wildly from, for example, 50 * 1K to 50 * 100MB depending on the pattern of messages sent over the channel. This means that there is no way to tell the channel to commit the batch sooner when some of these large messages appear and the problem will appear as a slow channel due to the need to re-transmit a very large quantity of data if there is a network outage. Adding a control based on bytes makes the transmission time more consistent. There is no precedence between BATCHLIM and BATCHSZ; whichever value is reached the first will cause the batch to be ended

z/OS Performance and Availability

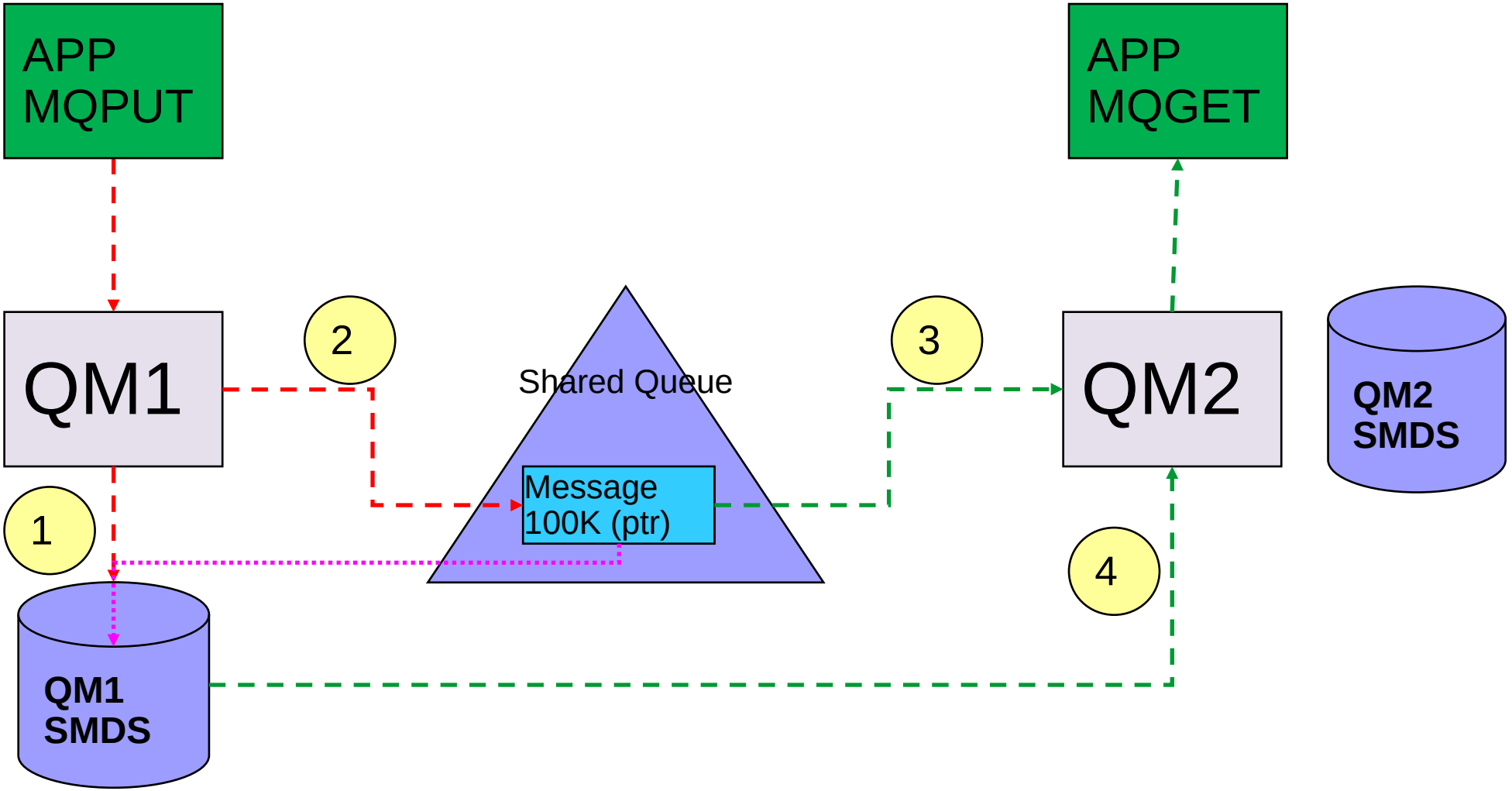
- Performance
  - z196 Scaling improvements for both non-shared and shared queues
    - Have successfully processed more than ONE MILLION non-shared messages/sec through a single queue manager
    - Have also successfully processed 150K shared msgs/sec with 3 queue managers
  - Improved performance by using SMDS for large messages on shared queues

- Availability
  - Structure rebuild when connectivity to CF is lost improves availability of Shared Queues
  - GroupUR function from MQ V7.0.1 for Distributed QSG connections available for CICS usag
    - CICS 4.2 can use this to enhance the MQ Group Attach originally provided in CICS 4.1

Large Shared Queue Messages: SMDS

- Using DB2 BLOBs to store large (>63KB) messages is expensive
  - Both CPU and pathlength
- Shared Message DataSets (SMDS) removes DB2 for large message storage
  - DB2 still needed for storing shared definitions
  - CF still holds small messages and pointers for offloaded messages
- Shared VSAM datasets increase shared queues capacity and performance
  - All queue managers in the QSG can access the datasets
- CF Structure message reference still controls locking, ordering, deletion etc.
  - So every message still has a “pointer” in the CF
- Rules control offload message size and % Structure-full offload trigger
  - Set per CF structure
  - Offloading messages at 63K gives 1.25M messages on a 100GB structure
  - Offloading all messages at 1KB gives 45M messages on same structure
- All QSG members must be at new level to access queues with this capability

Large Shared Queue Messages: SMDS

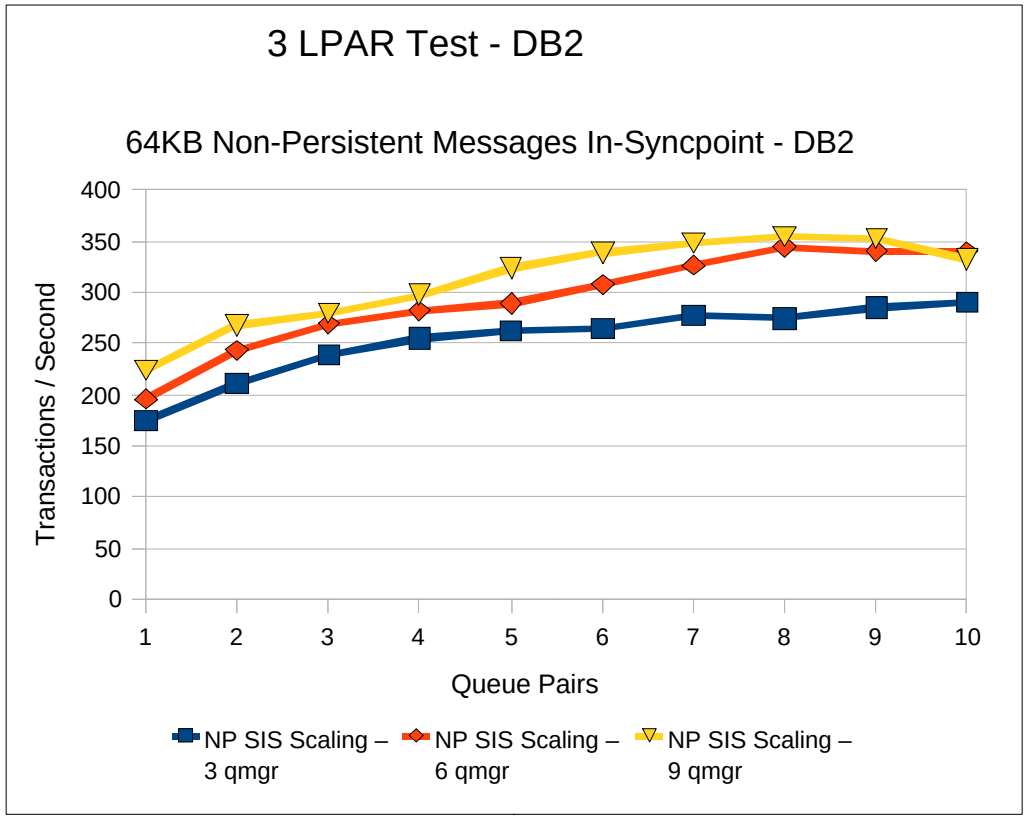


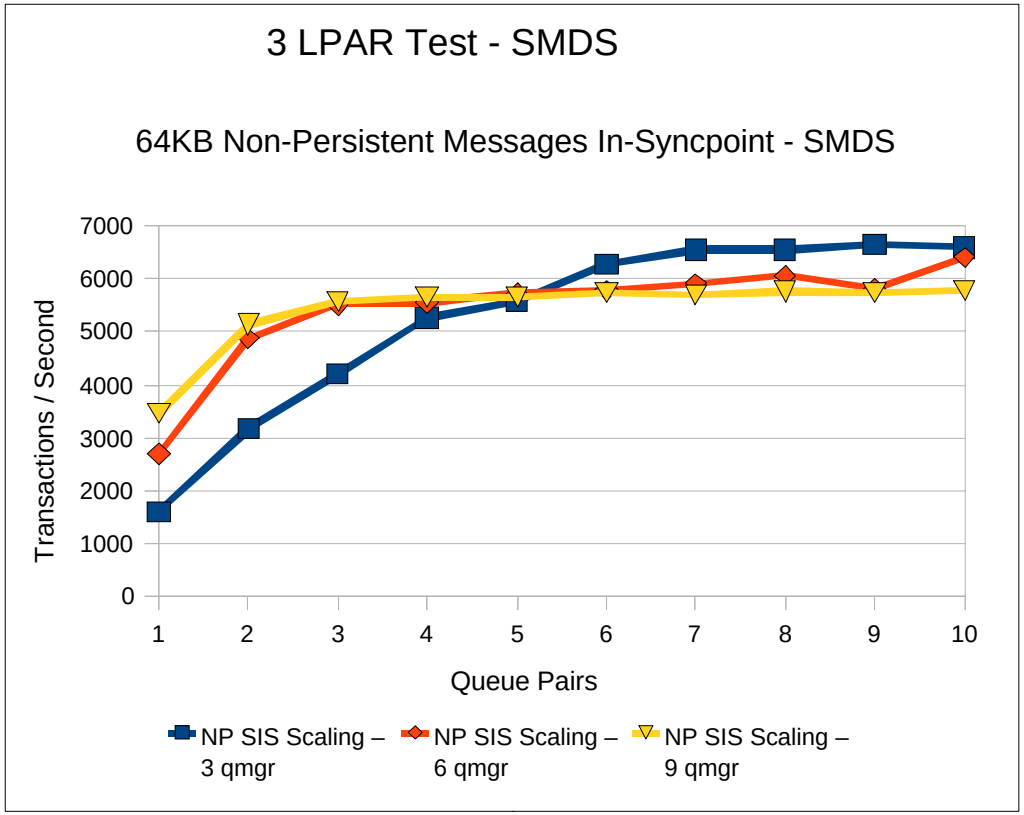
Shared Message Data Set Concepts

NOTES

- Offloaded message data for shared messages is stored in data sets.
- Each application structure has an associated group of shared message data sets, with one data set per queue manager.
  - Named using DSGROUP parameter on CFSTRUCT definition
- Each queue manager owns a data set for each structure, opened for read/write access, which it uses to write new large messages.
- Each queue manager opens the data sets for the other queue managers for read-only access, so it can read their message data
- When a message with offloaded data needs to be deleted, it is passed back to the queue manager which originally wrote it, so that the queue manager can free the data set space when it deletes the message.
- Messages too large for CF entry (> 63K bytes) are always offloaded
- Other messages may be selectively offloaded using offload rules
  - Each structure has three offload rules, specified on the CFSTRUCT definition
  - Each rule specifies message size in Kbytes and structure usage threshold
  - Data for new messages exceeding the specified size is offloaded (as for a large message) when structure usage exceeds the specified threshold
  - Default rules are provided which should be sufficient for most cases and can be set to dummy values if not required

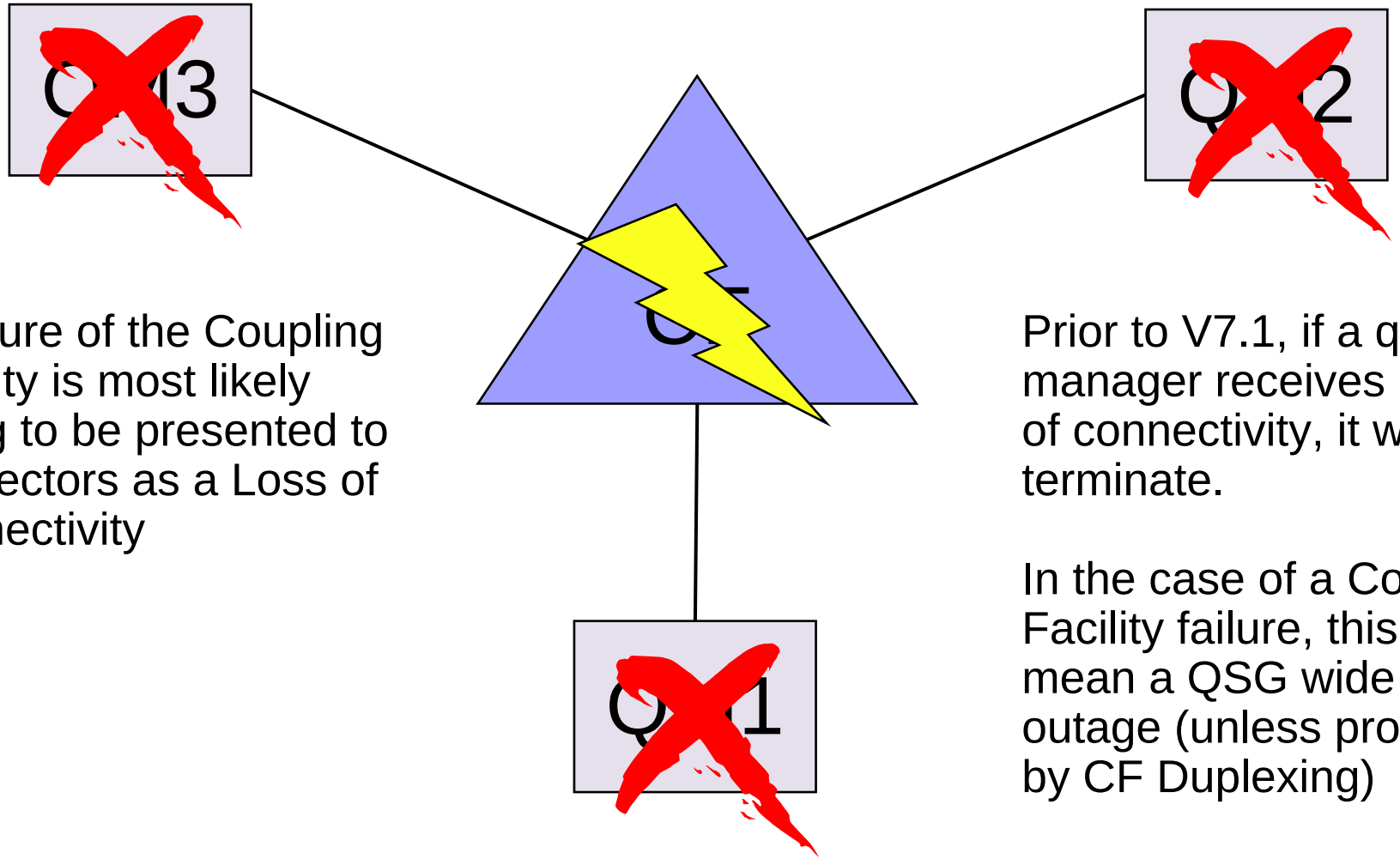
SMDS Performance Improvement





- Early Test Results on z196
- Tests show comparable CPU savings making SMDS a more usable feature for managing your CF storage
- SMDS per CF structure provides better scaling than DB2 BLOB storage

Loss of CF Connectivity: Before V7.1

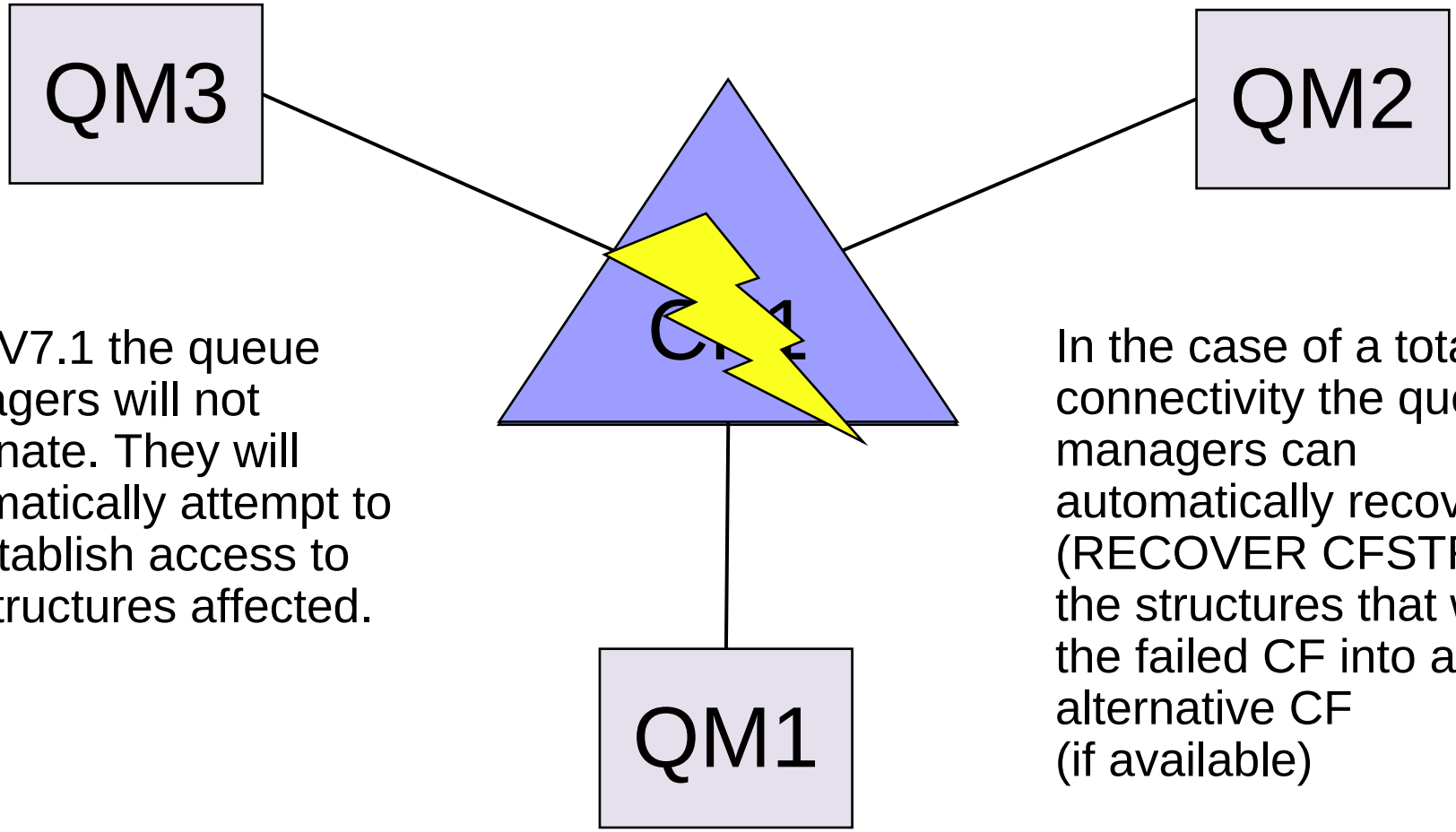


A failure of the Coupling Facility is most likely going to be presented to connectors as a Loss of Connectivity

Prior to V7.1, if a queue manager receives a loss of connectivity, it will terminate.

In the case of a Coupling Facility failure, this would mean a QSG wide outage (unless protected by CF Duplexing)

Total Loss of CF Connectivity: V7.1



With V7.1 the queue managers will not terminate. They will automatically attempt to re-establish access to the structures affected.

In the case of a total loss of connectivity the queue managers can automatically recover (RECOVER CFSTRUCT) the structures that were on the failed CF into an alternative CF (if available)

Loss of CF Connectivity: Admin Structure

NOTES

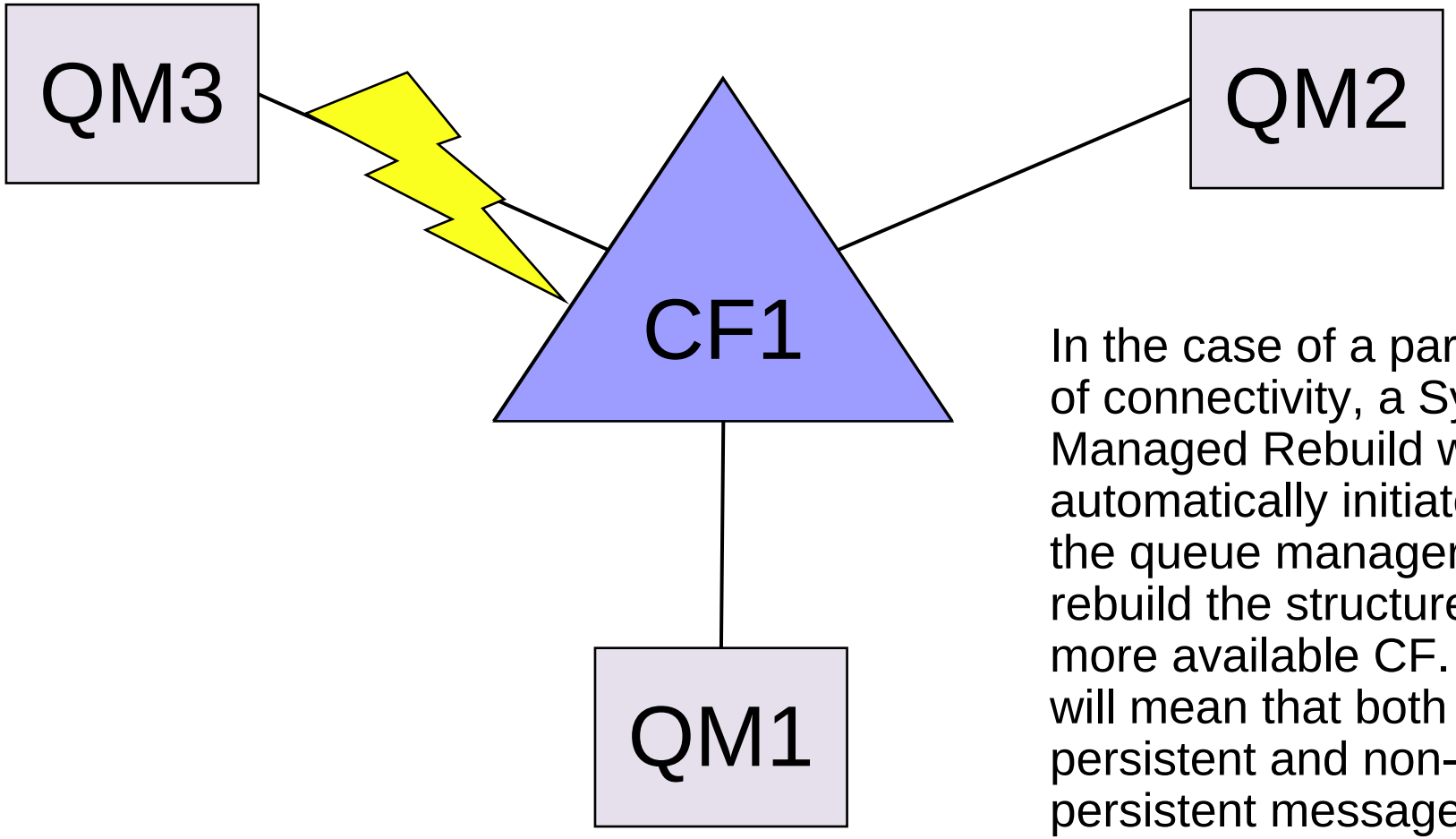
- Queue managers will tolerate loss of connectivity to the admin structure without terminating if
  - The QMGR CFCONLOS attribute is set to TOLERATE
  - All the queue managers in the QSG are at V7.1
- All queue managers in the QSG will disconnect from the admin structure, then attempt to reconnect and rebuild their own admin structure data.
- If a queue manager cannot reconnect to the admin structure, for example because there is no CF available with better connectivity, some shared queue operations will remain unavailable until the queue manager can successfully reconnect to the admin structure and rebuild its admin structure data.
- The queue manager will automatically reconnect to the admin structure when a suitable CF becomes available on the system.
- Failure to connect to the admin structure during queue manager startup is not tolerated, regardless of the value of CFCONLOS.

Loss of CF Connectivity: Application Structure

NOTES

- Queue managers will tolerate loss of connectivity to application structures if
  - They are at CFLEVEL(5)
  - The CFCONLOS attribute is set to TOLERATE
- All queue managers that lose connectivity to an application structure will disconnect from the structure
- The next action depends on whether it is a partial or total loss of connectivity
  - Loss of connectivity is partial if there is at least one system in the sysplex that still has connectivity to the CF that the structure is allocated in.
  - Loss of connectivity is total if all systems in the sysplex have lost connectivity to the CF that the structure is allocated in.
- In the case of total loss of connectivity
  - The structure will (probably) need to be recovered using the RECOVER CFSTRUCT command.
  - Non-persistent messages will be lost.

Partial Loss of CF Connectivity: V7.1



In the case of a partial loss of connectivity, a System Managed Rebuild will be automatically initiated by the queue managers to rebuild the structures into a more available CF. This will mean that both persistent and non-persistent messages will be retained.

Partial Loss of CF Connectivity: V7.1

NOTES

- In the case of partial loss of connectivity, queue managers that lost connectivity to the structure will attempt to initiate a system-managed rebuild in order to move the structure to another CF with better connectivity
- If the rebuild is successful, both persistent and non-persistent messages will be copied to the other CF
- Queue managers that didn't lose connectivity to the structure may experience a slight delay during system-managed rebuild processing, but shared queues will remain available
- If an application structure cannot be reallocated in another CF with better connectivity, queues on the structure will remain unavailable until connectivity is restored to the CF that the structure is currently allocated in
- Queue managers will automatically reconnect to the structure when it becomes available

Scalability & Performance – Distributed platforms

- Performance measured and improved for a range of scenarios
  - Hardware capabilities have evolved over years to have more CPUs, more memory etc
  - MQ topologies have evolved to have more clients and larger/fewer queue managers
- "Fastest MQ ever": better performance than V6 and V7
- Multicast faster than traditional non-persistent
  - Over 5x for one-many publications
- Performance reports to be released on availability

Scalability & Performance – Distributed platforms

- Performance measured and improved for a range of scenarios
  - Hardware capabilities have evolved over years to have more CPUs, more memory etc
  - MQ topologies have evolved to have more clients and larger/fewer queue managers
- "Fastest MQ ever": better performance than V6 and V7
- Multicast faster than traditional non-persistent
  - Over 5x for one-many publications
- Performance reports to be released on availability

Scalability & Performance – MQ Explorer

- Design changes to MQ Explorer reduce its footprint and improve performance
- Now does not include full Eclipse development workbench
  - But Explorer can be easily added to other Eclipse installations and products

- Many Explorer installs are supported within the overall multi-version support
  - But each Explorer only fully manages queue managers associated with its own installation
  - Use client connections for other installation queue managers on same machine

	V7.0.1	**V7.1**
Time to install MS0T	203 seconds	92 seconds
Install Footprint (Windows excluding KRE)	309 MB	96 MB
Startup Time	6 seconds	4 seconds
Connect to 100 queue managers	At least 53 seconds	7 seconds
Enable and disable Sets for 100 queue managers	35 seconds	1 second

Management of Distributed platforms

- New integrated command (dmpmqcfg) to extract configuration
  - Fulfills the role that MS03 (saveqmgr) has done over many years
    - Backup your configuration, change control, rebuild systems etc
  - MAKEDEF already available on z/OS
  - Different syntax than MS03, but similar function

- MQSC commands equivalent to setmqaut/dspmqaut
  - So you don't need to drop out of the command interface to modify security
  - Can simplify scripting of configuration changes
  - No current plans to remove *mqaut commands

- Multi-instance Queue Managers on Windows
  - The need for domain controllers ("domainlets") has been removed
  - New option when creating queue managers to define ownership

Management of Distributed platforms

- New integrated command (dmpmqcfg) to extract configuration
  - Fulfills the role that MS03 (saveqmgr) has done over many years
    - Backup your configuration, change control, rebuild systems etc
  - MAKEDEF already available on z/OS
  - Different syntax than MS03, but similar function

- MQSC commands equivalent to setmqaut/dspmqaut
  - So you don't need to drop out of the command interface to modify security
  - Can simplify scripting of configuration changes
  - No current plans to remove *mqaut commands

- Multi-instance Queue Managers on Windows
  - The need for domain controllers ("domainlets") has been removed
  - New option when creating queue managers to define ownership

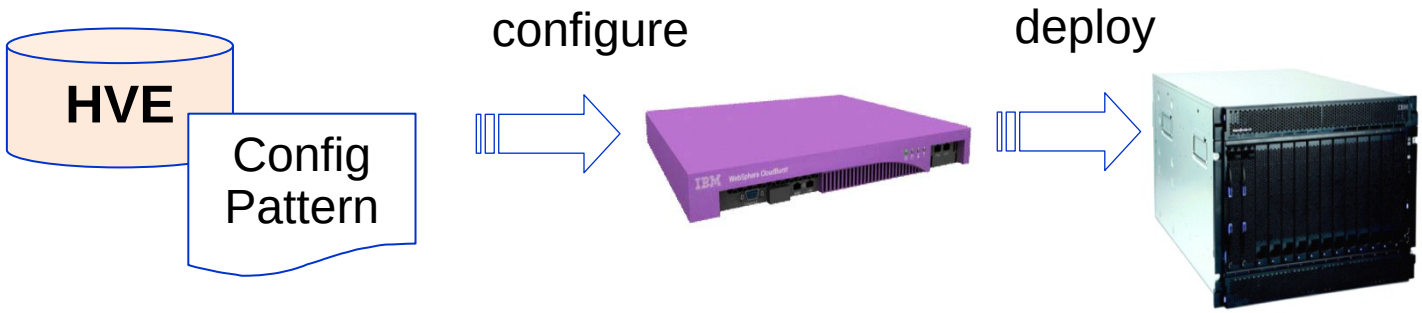
Management of Distributed platforms

NOTES

- Probably the most commonly-used SupportPac has been MS03 (saveqmgr). In MQ V7.1, the concept is now built into the product and formally supported. The new dmpmqcfg command has a slightly different syntax, but fulfills the same role of extracting a queue manager's configuration and displaying it in MQSC syntax.
- Dmpmqcfg can connect to local queue managers or use client connections to remote systems.
- New MQSC commands are available that are equivalent to the set/dsp/dmpmqaut commands. These may be more convienent when you are already inside runmqsc, rather than dropping out to the command line, and certainly more convenient when scripting configuration changes. The new dmpmqcfg for example can put authorisations in a single MQSC script for replay, rather than having to run separate commands.
- On Windows, the requirement for multi-instance queue managers to be domain controllers (even if limited in scope eg "domainlets") has been removed. When a queue manager is created, you can now name a group that both machines share knowledge of, and that group gets appropriate ownership of resources such as the files and directories that make up the queue manager.

MQ Cloud Support: HyperVisor Editions

- HVE is pre-packaged image of MQ with an operating system
  - For easy configuration deployment into virtualised environments
- First release included MQ V7.0.1.4 and Red Hat Enterprise Linux x86 64-bit OS
- Also now available with an AIX flavour
- Pre-defined patterns for IBM WebSphere Workload Deployer

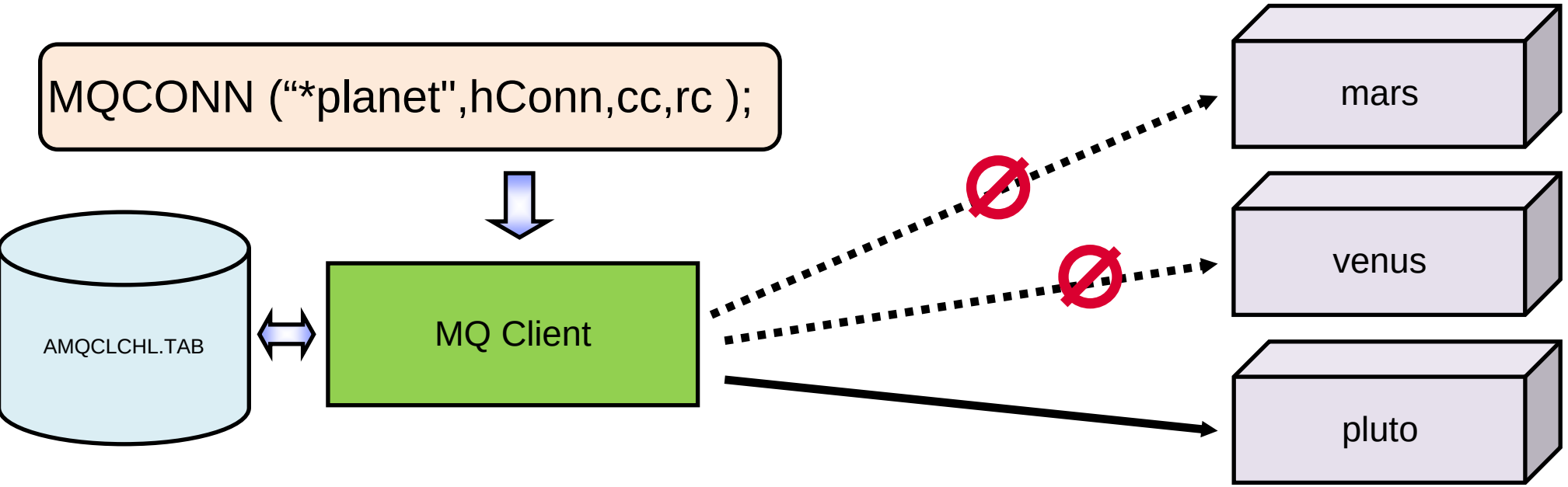


MQ Cloud Support: Pre-Connect Exit

- Supports movement by some to “Utility Compute”, Private Cloud configs, etc.
  - Rapid provision of applications allied with need to further decouple Client/Server connectivity
  - Server applications might move location – new addresses or queue managers
- MQ Client connects to a “service” rather than specific Queue Manager
- Can transparently change location of MQ server-side applications
  - No client code changes needed
  - No configuration files need to be updated at the client machine
  - JMS/XMS applications already do this via JNDI lookup
- Exit run during MQCONN queries a repository to discover real location
  - MQ V7.1 incorporates the LDAP implementation from SupportPac MA98

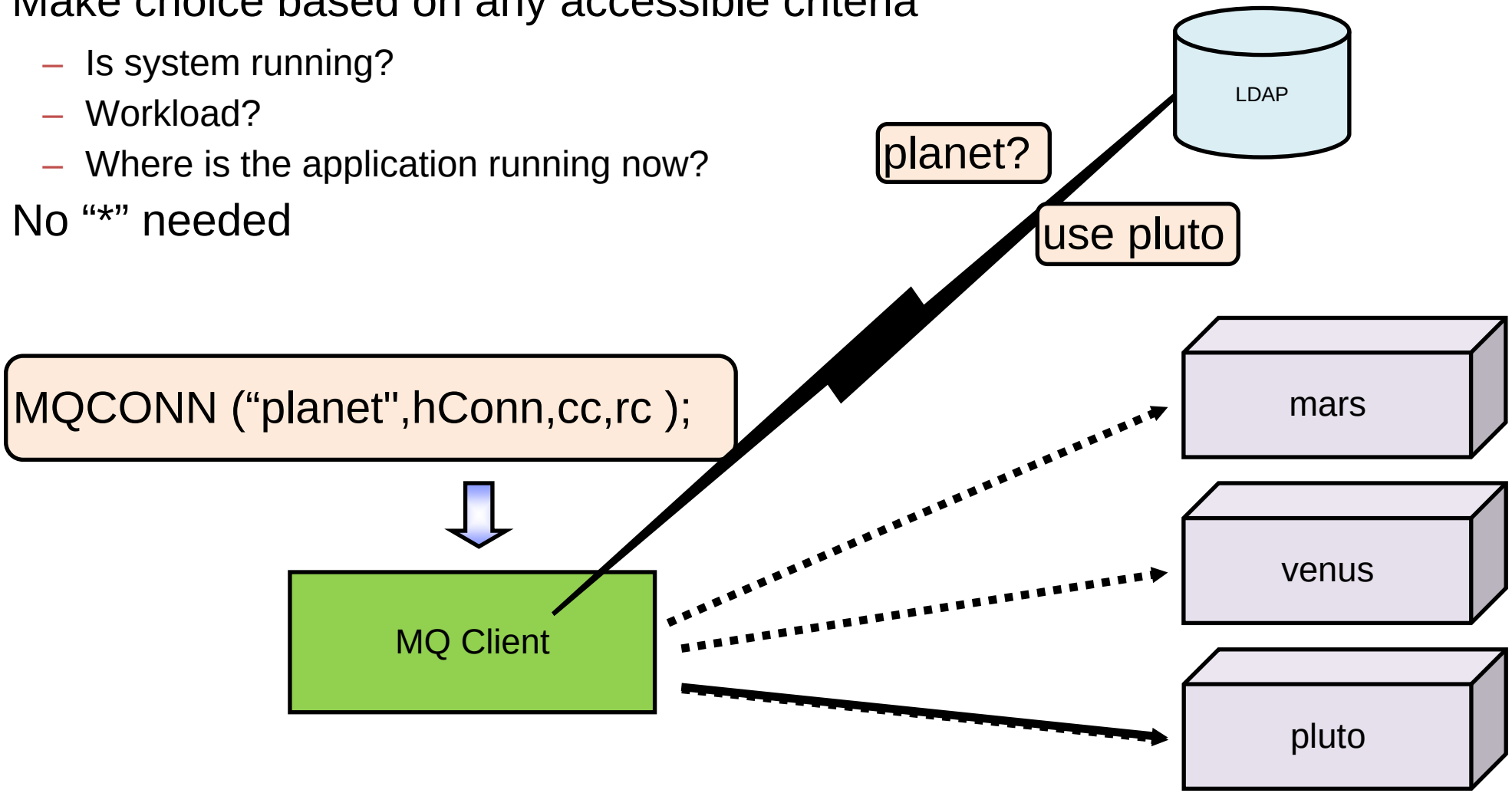
Dynamic Client Connection Using Channel Definition Tables

- How it used to be done …
- The CCDT is used to select a queue manager from a list
  - Based on a pseudo-queue manager name prefixed with “*”
  - CCDT is a locally-accessible file
- CCDT must be distributed to all client systems



Dynamic Client Connection Using Pre-Connect Exit

- Look up in a directory such as LDAP
- Make choice based on any accessible criteria
  - Is system running?
  - Workload?
  - Where is the application running now?
- No “*” needed



WebSphere MQ V7.1: Feature Summary

New Feature	*Benefits*	*Details*
Multi-Version Install capability on Distributed platforms	Makes it easier to deploy and upgrade systems and stage version to version migration	Unix and Windows support for multiple versions of MQ V7.x (AND one copy of MQ V7.0.1) down to fixpack levels. Relocatable installation support. Applications can connect to any Qmgr
Enhanced Security	Simplified Configuration Enhanced Authorisation and Auditing	IP address Authorisation capability Additional crypto algorithms More granular authorisation for non-local queues Application Activity Reports
Cloud Support	Simplifies and support Cloud deployments	Additional HVE images
Enhanced Clustering	Improves ease-of-use	Authorisation on Cluster Q rather than XMIT Q on Dist. Platforms Bind-on-Group Support
Multicast capability	New messaging QoS provides low latency with high fan-out capability	MQ Pub/Sub Topic space can now map to multicast Group Addresses Provides direct interoperability with MQ LLM
Improved scalability and availability on z/OS	Further exploitation of z196 Customer control over CF storage use CF Connectivity Loss improvements	Code contention reduced to improve multi-processor linear scaling Use of MQ Datasets rather than DB2 significantly improves “large” message capability Structure rebuild capability for CF Connectivity Loss scenarios
Improved Performance on Dist platforms	Improved multiprocessor exploitation	Various code improvements

Why WebSphere MQ ?

Over 17 years of proven experience	Leader in Messaging technology innovation
Connect virtually anything	Broad coverage of platforms, technologies, languages Draw skills from a larger pool – use who you have today Over 9,300 certified developers for IBM Messaging alone
Most widely deployed Messaging Backbone	Over 10,000 customers using IBM Messaging Backbone Over 90% of the Fortune 50 and 9 of the Fortune 10 Over 80% of the Global 25 and 7 of the Global 10
Entrusted with Tens of billions of messages each day	Government client sends 675 million messages per day* Banking client handles over 213 million messages per day on z/OS alone*
Relied upon as the mission-critical Backbone	Financial Markets client handles $1 trillion worth of traffic per day on one MQ network* Banking client sends $7-$35 trillion worth of traffic per day on just one MQ-based SWIFT gateway*
Continuously Investing and Innovating	Over 120 patents and filings within messaging space New WebSphere MQ family products Regular enhancements, updates and new releases

- ***Results reported from actual MQ implementations***

Universal Messaging with WebSphere MQ