Introduction to IBM MQ

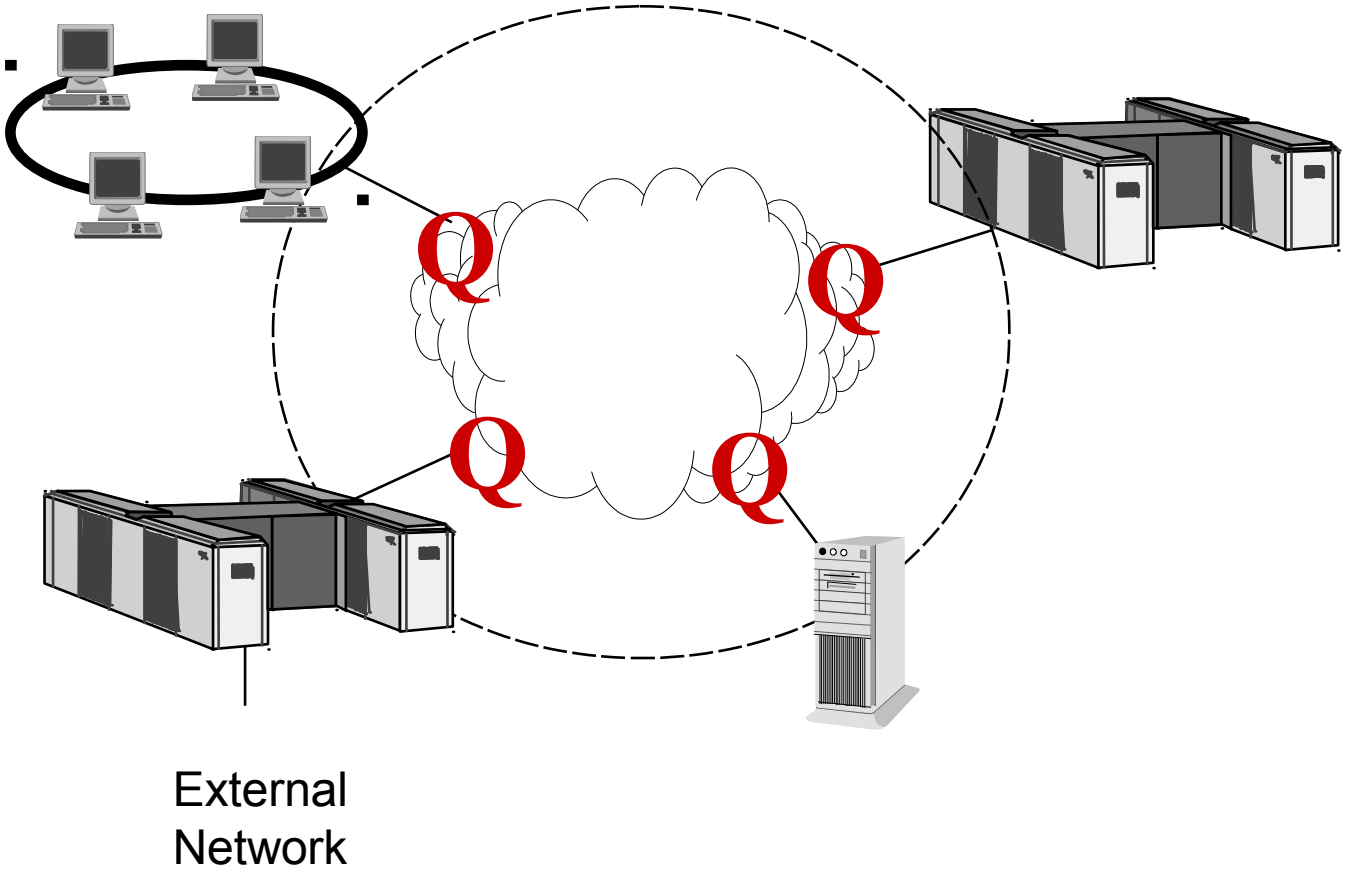
Roger Lacroix
roger.lacroix@capitalware.com
http://www.capitalware.com

Why use IBM MQ

Things to consider:

- Importance of parallel operations
- Variety of info flow patterns
- Dependence upon serialization
- Message traffic volumes
- Relationship of programs
- Programming skills level
- Knowledge of networking
- Mixtures of application types
- Mixed old/new applications

Queuing Technology



Reasons to Choose IBM MQ

- Take away the communications nightmares
- Component-built applications approach
- Allow control of load balancing
- Allow protocol independence
- Provide consistent programming interface across platforms
- More supported platforms than any other product

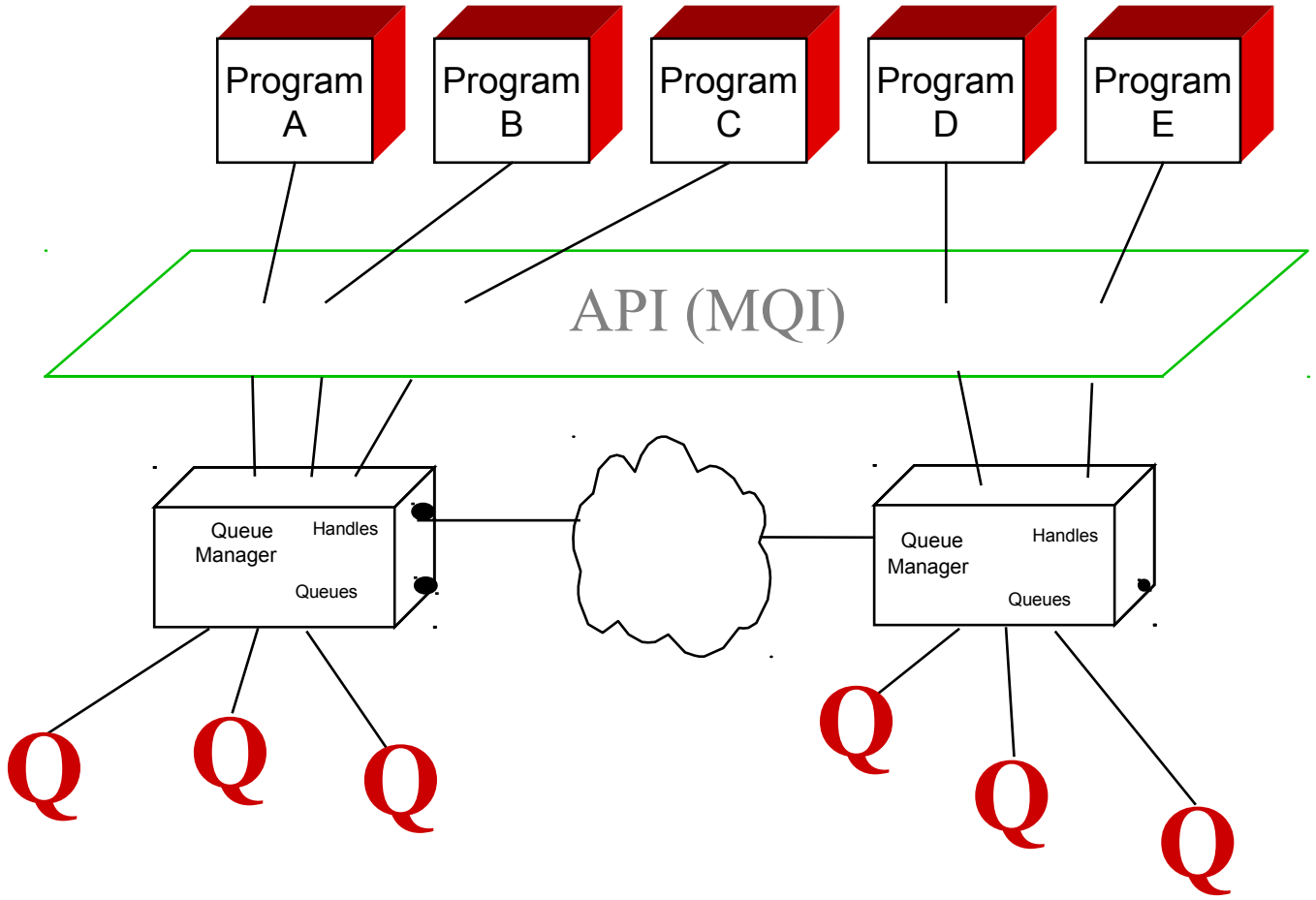
Business Perspective

- Time independent (asynchronous) processing
- Connectionless communications
- Assured message delivery
- Once and once only delivery
- Syncpoint control
- Resource manager
- Integrated with operating system
- Interfaces to other system managers
- Triggering, Message grouping, Clustering

Application Perspective

- Single, multi-platform Application Programming Interface (API)
- Faster application development
- Portable code

MQ Environment Overview



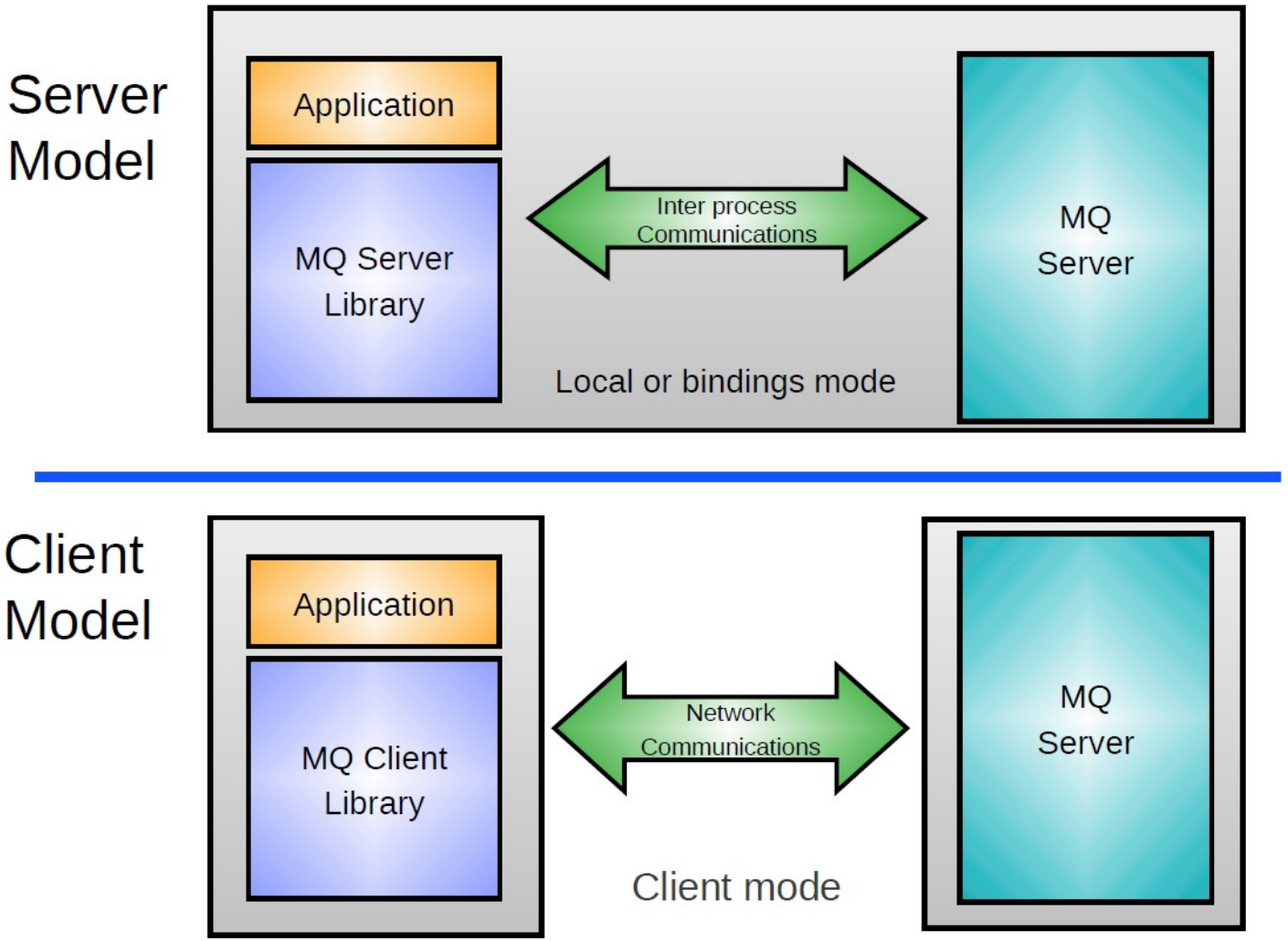
IBM MQ Clients

- An IBM MQ client is a component that can be installed on a separate machine from the Base product and Server
- IBM MQ applications can run on client
- Client uses a server queue manager via a network protocol (i.e. TCP/IP)

Why use IBM MQ Clients?

- Supported on roughly 45 platforms
- Reduces client hardware requirements
- IBM MQ Client license is free

Connecting: Bindings vs Client Mode



Terminology

Message:

A message is a collection of data sent by one program and intended for another program.

Examples of Messages

- Units of information transfer (one-way)
- A request for service or information
- A reply to a service or information request
- A report of status
- An announcement or broadcast

MQ Message Overview

Message = Header + User Properties + User Data

A Series of Message Attributes
Understood and augmented by the Queue Manager
- Message Id
- Correlation Id
- Routing information
- Reply routing information
- Message priority
- Message codepage/encoding
- Message format

....etc.

- User Properties require WMQ V7
  - Emulated for JMS in older versions of WMQ
- Arbitrary properties
  - For example, this is a “green” message

- Any sequence of bytes
- Private to the sending and receiving programs
- Not meaningful to the Queue Manager

Terminology

MQ Objects:

- Queue managers
- Queues
- Channels
- Processes
- Namelists
- Distribution Lists
- Topics
- Storage Classes

Terminology

Queue Manager:

A queue manager is the IBM MQ component that provides the messaging and queuing services to application programs through Message Queue Interface (MQI) program calls.

Queue Manager Characteristics

- May define Multiple Queue Managers
  - Development
  - Testing
  - Acceptance
  - Production
- Name
  - Up to 48 characters in length, case sensitive
  - Z/OS only 4 characters (Subsystem ID)

Terminology

Queue:

A queue is an IBM MQ object that can store messages. It has attributes that determine what processing occurs when an application accesses it through the MQI calls.

Queue Definition Types

- Local
- Remote
- Alias
- Model
- Dynamic:
  - Permanent
  - Temporary
- Cluster
- System:
  - Transmission
  - Event
  - Dead letter
  - Initiation
  - Command server
  - Cluster
  - Pub/Sub Broker Queues

Terminology

Channel:

A channel is a communication link providing a path on the same or different platforms. The message channel is used for the transmission of messages from one queue manager to another, and shields the application programs from the complexities of the underlying networking protocols.

Channel Characteristics

- Sender - Receiver - push-type model
- Requester - Server - pull-type model
- Requester - Sender - call-back model
- Server - Receiver - push-type model
- Cluster Server – Cluster Receiver - push-type model

Note: Channel pairs must match, names of channels in pairs must be identical

Terminology

Process:

A process is an IBM MQ object that defines an application to the IBM MQ Queue Manager.

Process Characteristics

- Process definition used to identify applications to be started by a trigger monitor
- The process definition includes application ID and type, plus some application specific data

Terminology

Namelist:

A Namelist is an IBM MQ object that contains a list of other IBM MQ objects.

Namelist Characteristics

- Example of use is for trigger monitors where a Namelist could contain a list of queues to monitor
- Can be maintained independently of applications that use it

Terminology

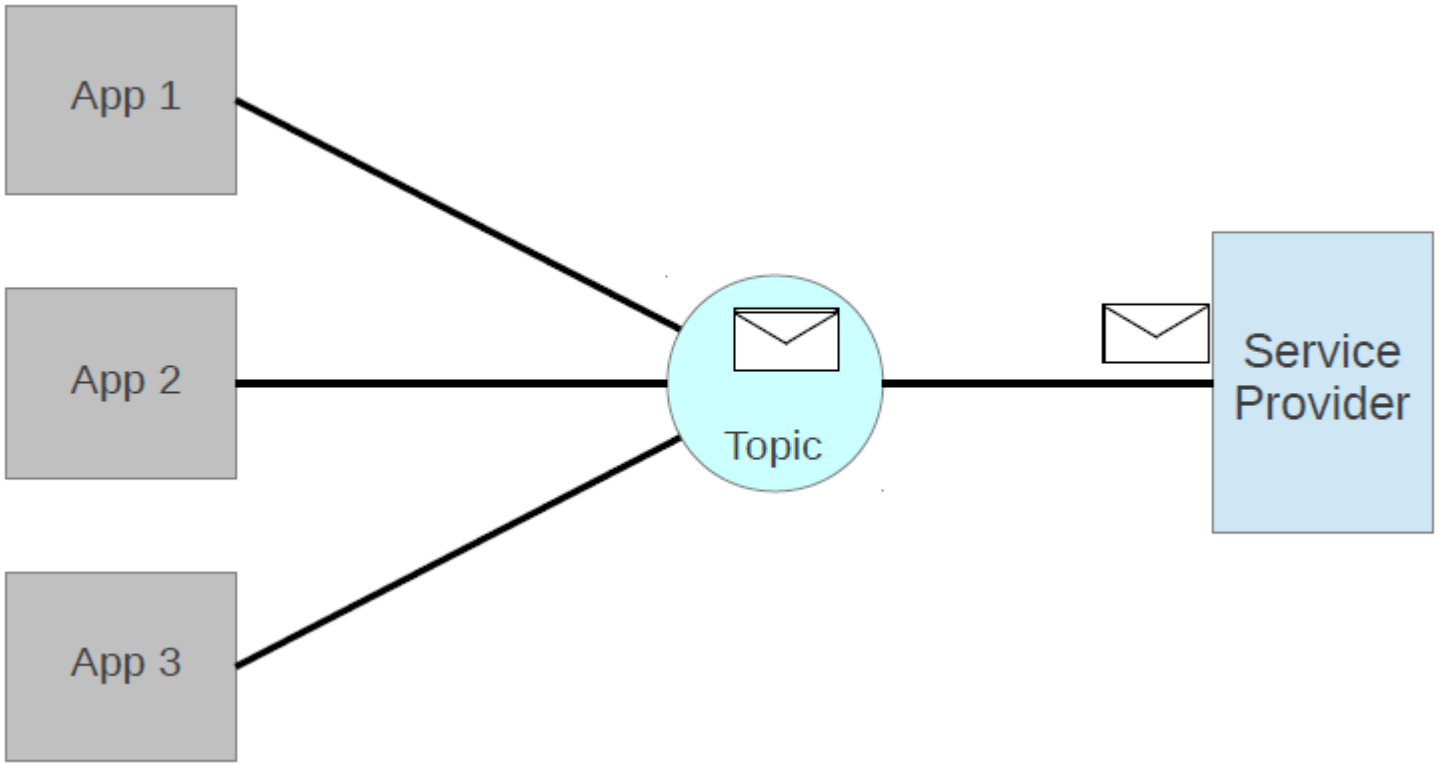
Topic:

A topic is an IBM MQ object that contains a theme that applications publish messages to.

Topic Characteristics

- A topic string is a case sensitive
- '/' The topic level separator – provides structure to topic trees
- '#' The wildcard character
- '+' The single-level wildcard character
- The Topic can be defined in a number of ways:
  - Predefined by the MQSC command
  - Predefined by the PCF interface (as used by the IBM MQ Explorer)
  - Subscribing or Publishing to the Topic object

Topic Overview



Terminology

Distribution List:

A Distribution List is used by an IBM MQ application to access a group IBM MQ queues. It is generated by the application.

Distribution List Characteristics

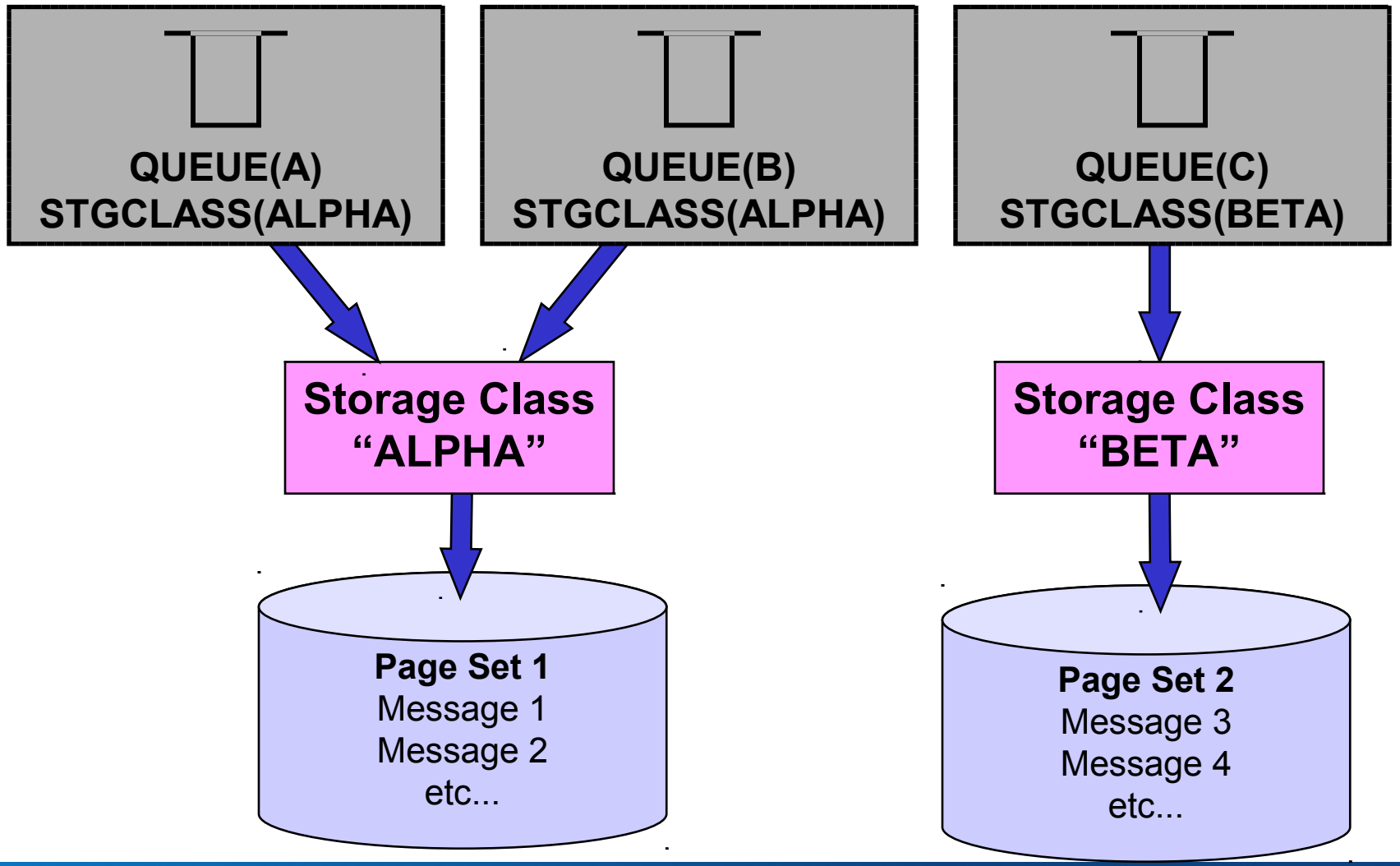
- Allows one MQPUT to send a message to many destinations.
- If the same transmission queue is to be used for multiple destinations, only one copy of the message is placed on the queue.

Terminology

Storage Class:

A Storage Class is an IBM MQ object that is used to map one or more queues to a OS/390 page set.

Queue/STGCLASS Relationship



Queue Manager Clusters

- This is a logical grouping of Queue Managers (QMGRs). The QMGRs can be physically remote.
- A QMGR can belong to more than one cluster. (Overlapping clusters)
- QMGRs can advertise Qs to the clusters. Channel definitions, XMITQ definitions, and QR definitions are not required.
- More than one QMGR can advertise the same Q name. (work load balancing / failover)

Message Queue Interface (Procedural)

- 13 verbs (original) + 12 introduced in WMQ v7.0
- 7 are used most commonly
- Important to understand use of call parameters

Common MQI Calls

- MQCONN — Connect to Queue Manager
- MQOPEN — Open a queue
- MQPUT — Put message to queue
- MQGET — Get message from queue
- MQCLOSE — Close a queue
- MQDISC — Disconnect from Queue Manager
- MQPUT1 — Put one message on a queue

Specialized MQI Calls

- MQBEGIN Signals start of Unit of Work
- MQCMIT Commits Unit of Work
- MQBACK Rollback Unit of Work
- MQINQ Inquire on MQ object
- MQSET Set queue attributes
- MQCONNX Connect with special options

New MQI Calls in WMQ v7.0

- MQCB - Defines a callback function
- MQCTL - Start/stop message delivery
- MQSUB - Registers a subscription
- MQSUBRQ - Request services from a subscription
- MQSTAT - Obtain information about previous Async puts

New MQI Calls in WMQ v7.0

- MQCRTMH - Create a message handle
- MQDLTMH - Delete a message handle

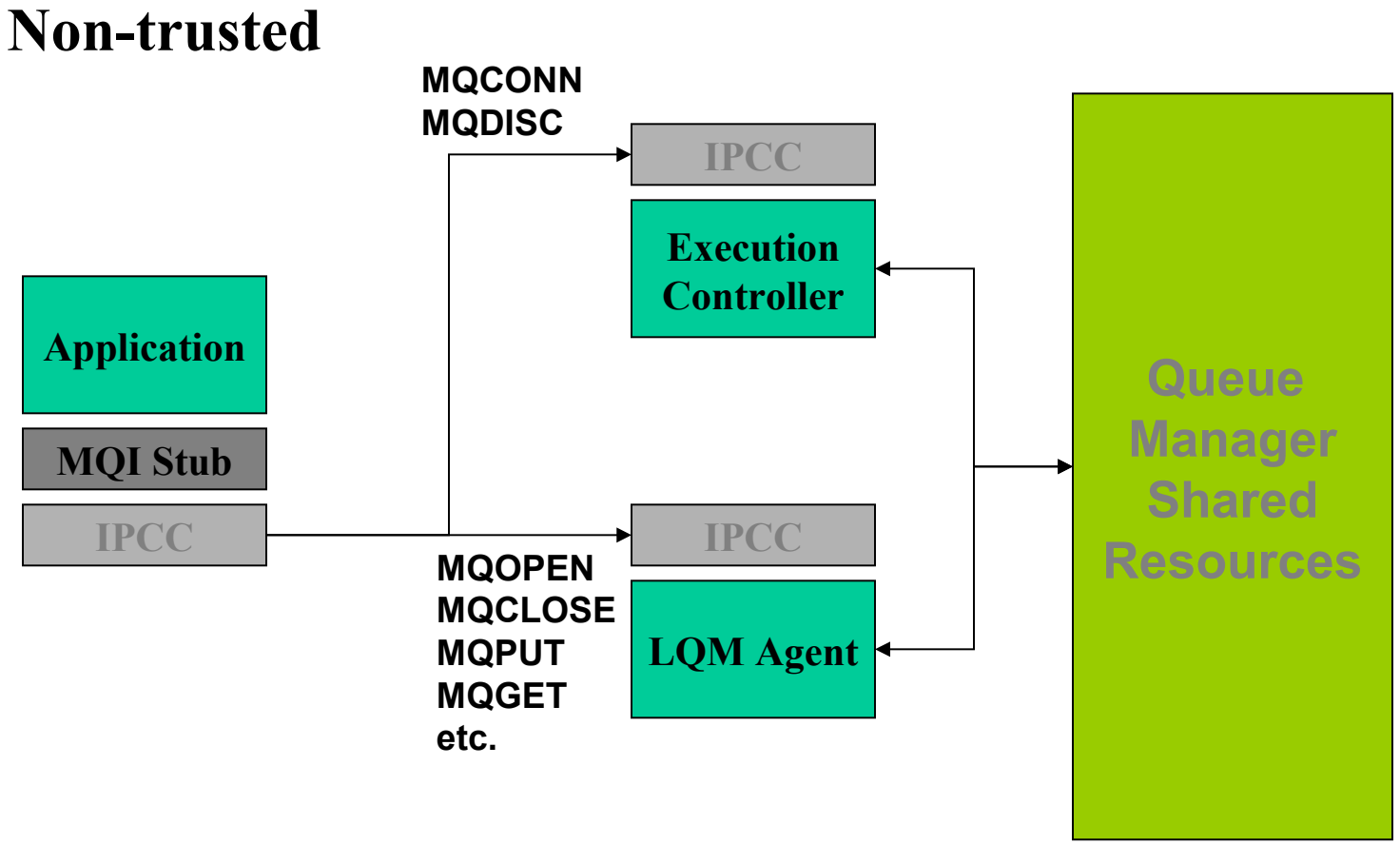
- MQSETMP - Set a message property
- MQINQMP - Inquire on a message property
- MQDLTMP - Delete a message property

- MQMHBUF - Converts buffer into message handle
- MQBUFMH - Converts a message handle into a buffer

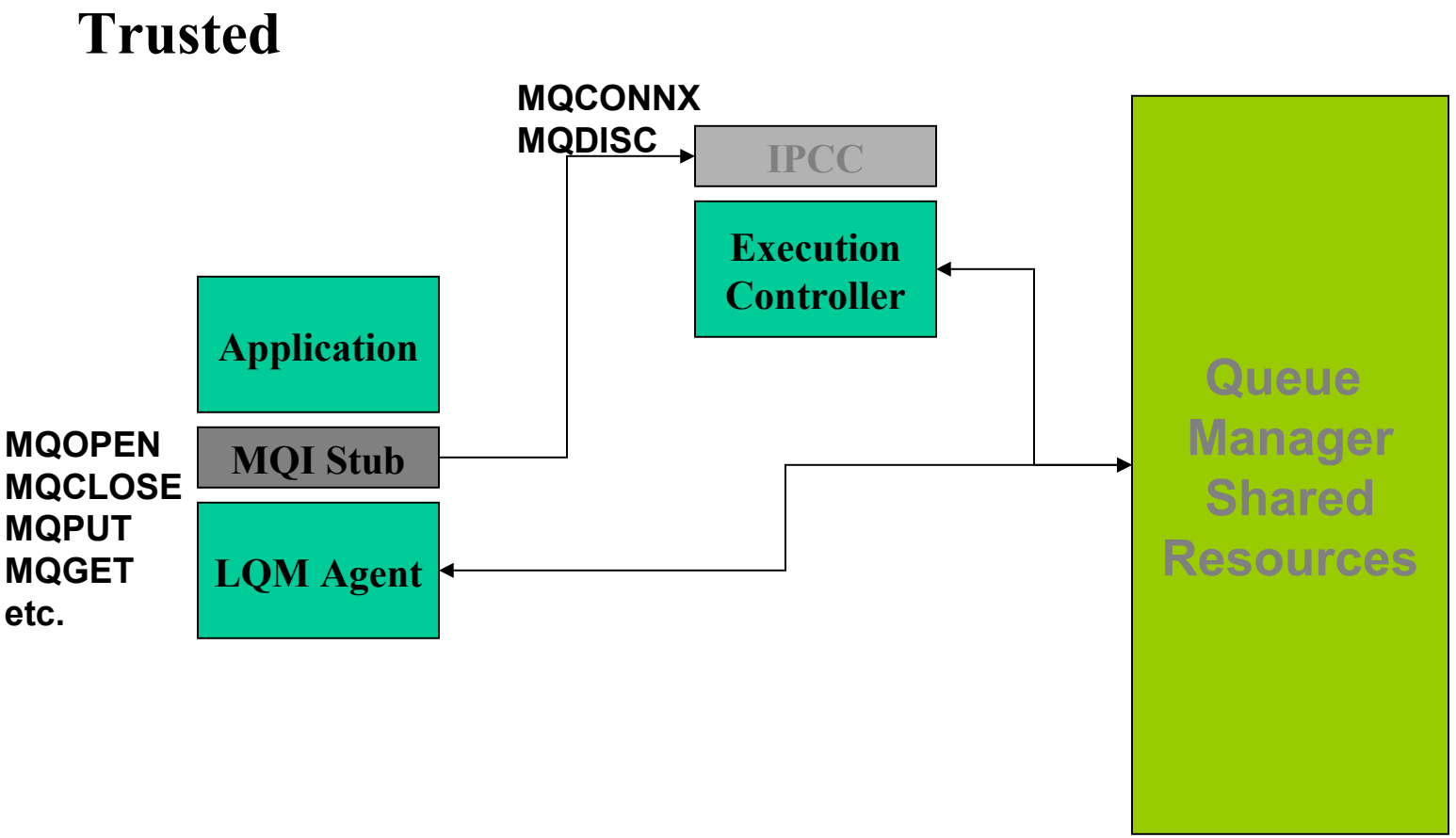
Summary of MQ API Verbs

Connection	Resource Use	Messages	Object attributes	Transactions	Message Properties
MQCONN	MQOPEN	MQPUT	MQINQ	MQBEGIN	MQCRTMH
MQCONNX	MQSUB	MQPUT1	MQSET	MQCMIT	MQDLTMH
MQCTL	MQSUBRQ	MQGET		MQBACK	MQSETMP
MQDISC	MQCLOSE	MQCB			MQINQMP
					MQDLTMP
					MQMHBUF/MQBUFMH

Application/MQ Relationship



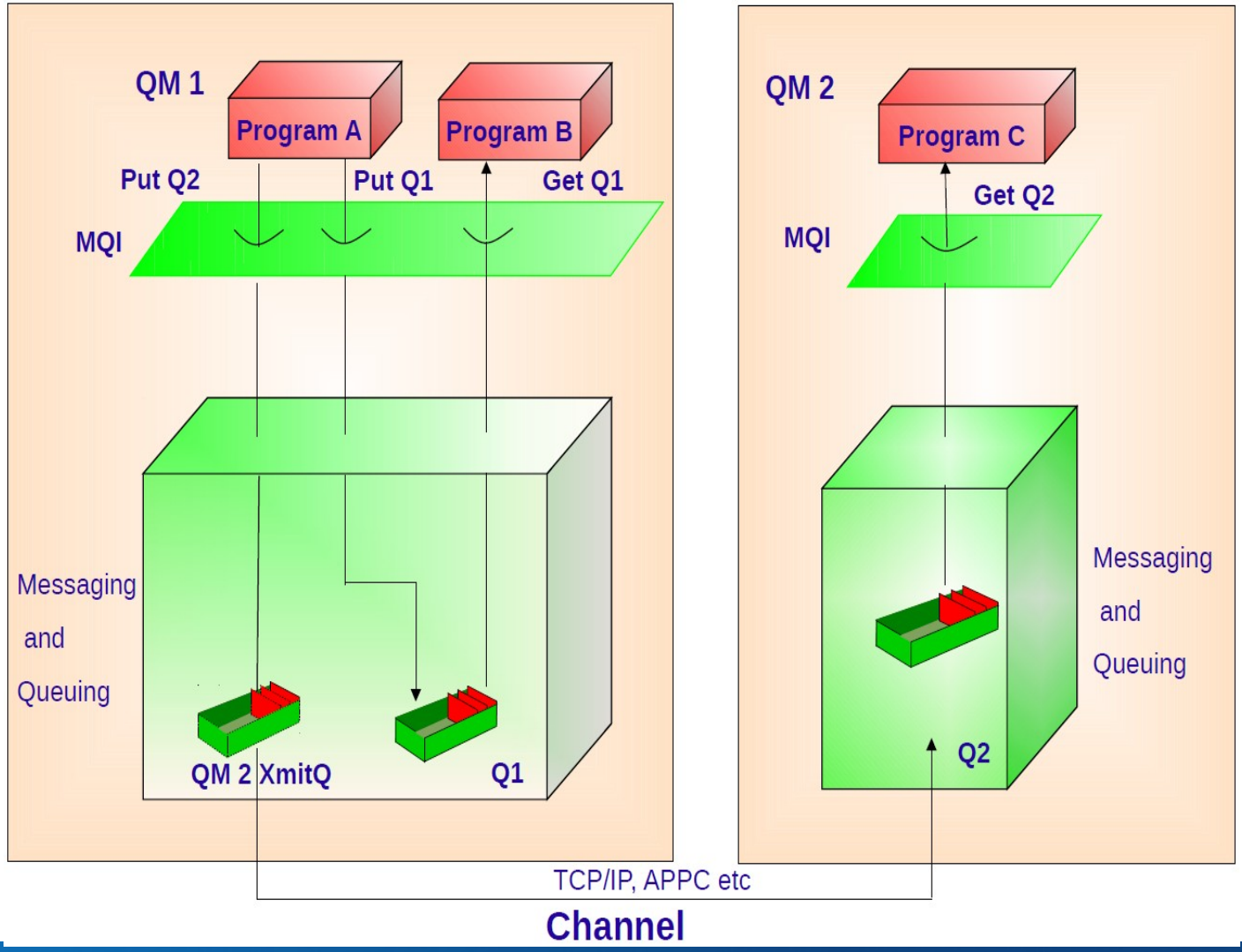
Application/MQ Relationship



PCF Messages

- Specialized Message Structure, similar to that used in MQ Event messages.
- Header and multiple parameters
- Has similar functions to RUNMQSC commands
- Permits Remote Management

The Big Picture



Questions & Answers