

Introduction to Messaging and WebSphere MQ

Paul Roy

Middleware definitions

- Middleware is a general term for any piece of software that serves to 'glue together', mediate between, or enhance separate existing programs.
- Middleware is software that is used to move data from one program to another, shielding the developer from dependencies on communications protocols, operating systems and hardware platforms.

What is middleware ?

- Software that manages some parts of a connection between systems
 - Physical and logical communication
 - Data transmission
 - Error recovery
 - Messages sequencing
 - ...

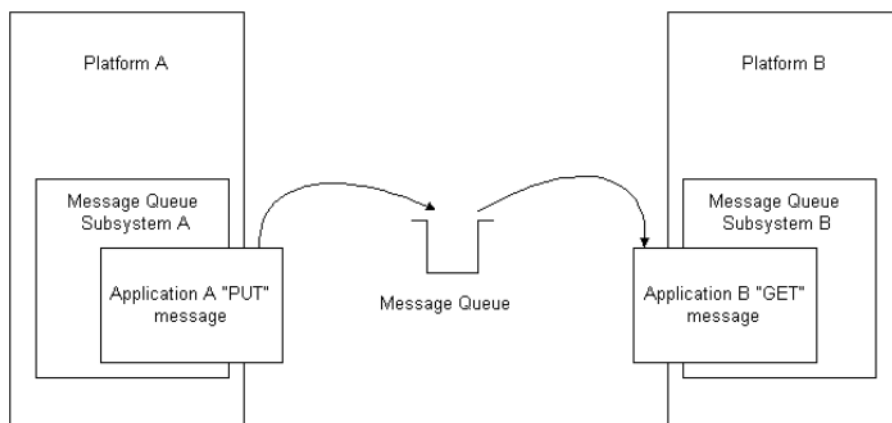
Websphere MQ

- Is a MOM (message oriented middleware)
- Is a framework for communications
 - Multiple levels
 - Multiple Operating systems
 - Multiple hardware
 - Multiple communications protocols
 - Multiple connectivity (distributed)

MOM Advantages

- Easy transaction over the internet
- Easy reuse and integrate any existing application
- Reusable components easily interfaced
- Simple programming concentrated on business logic.
 - format a message and put it on a queue.
 - Get the message from a queue and process the content.

Message Oriented Middleware



Websphere MQ platform(s)

- AIX
- Tandem
- Z/OS
- i5/OS
- OS/400
- DOS
- Microsoft
- Java
- OpenVMS Vax
- HP-UX
- Linux
- Sun Solaris
- Unix
- VM/ESA
- VSE/ESA
- MVS/ESA
- OS/2
- MacOS
- OpenVMS Alpha

Access

- API for all platforms
- Source is independant from target
- Same concept for each system
- Store and Forward (vs Send&Forget)
 - Guaranteed delivery
- Asynchronous communication

Components

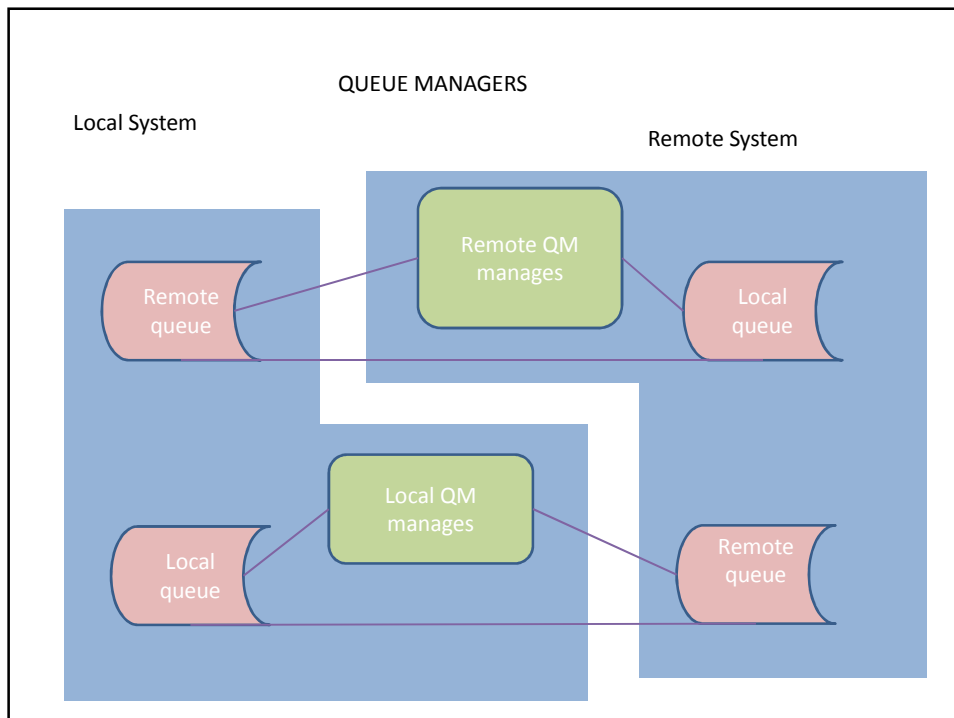
- 4 main components
 - Queue manager
 - Queues
 - Channel
 - Message

Queue manager

- Is the 'subsystem' responsible for queues
- Each queue belongs to one queue manager
- A local queue is managed by a local queue manager
- A remote queue is managed by a remote queue manager
- A local application puts messages on a remote queue (is a local queue for the remote queue manager) and the remote application gets messages from a local queue (viewed as remote from the local application).

Message

- The data transmitted
 - Is meaning full only for the application.
- 2 parts of the message
 - Message Data :The content of the message is transparent for MQ.
 - Message descriptor
 - From and to queues, priority, security, timestamp, Code page, etc...

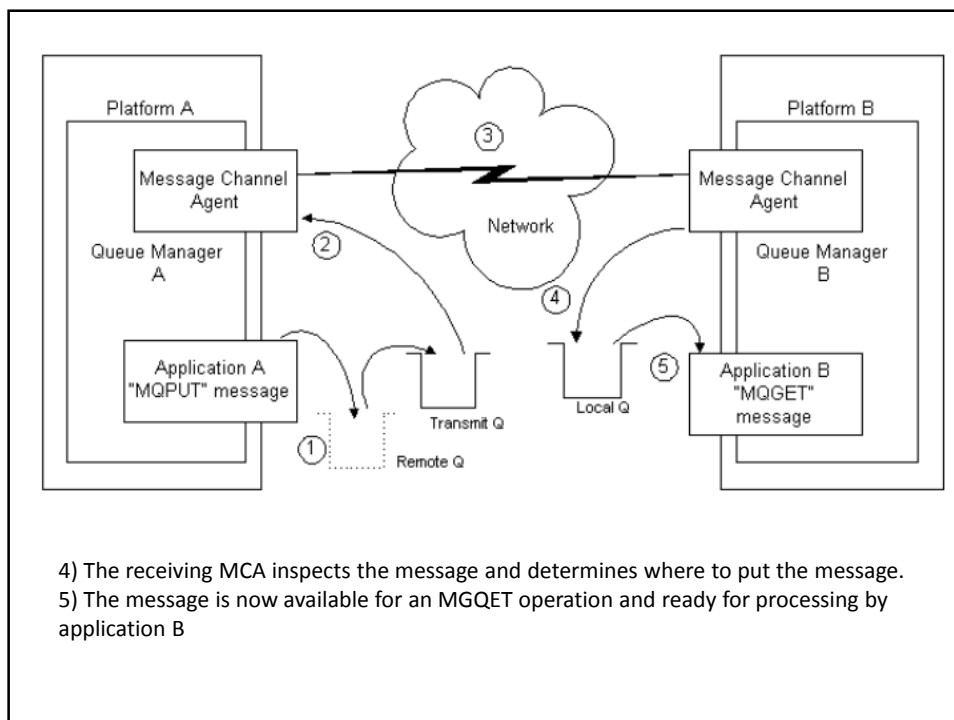
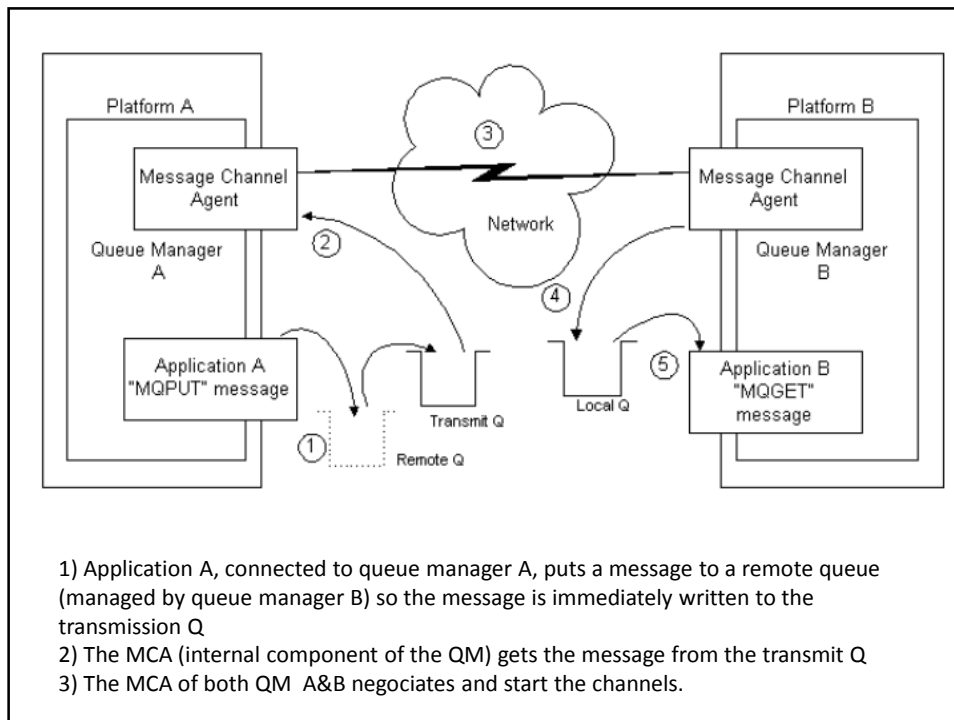


QUEUES

- Repository for messages
- Same concept as a dataqueue or a message queue i5/OS objects (*DTAQ *MSGQ)
- Access through API
- The sending application is independant of the receiver.
- Characteristics
 - Put/Get enabled
 - Queue depth
 - Queue size

CHANNELS

- Link between communication layer and the Queue Manager
 - TCP/IP or SNA
 - 2 types of channels (message and client)
 - Message channel is used for server-server (QM/QM) communication
 - Unidirectional
 - 2 channels: sending channel and a receiving channel
 - Client channel connects a client to a server
 - Bi directional.
 - Queues only exist on the server.



Client connections

- If the application does not physically reside on the same machine as the queue manager, it can connect to a 'remote' queue manager using a client channel . There is no difference in the program.

MQ scripting language

- MQ objects can be created on IBM i using i5/OS command or a command file can be executed (RUNMQSC)

MQ Security

- Built in security : Object Authority Manager
 - Access control to MQ objects
 - Groups/users grpprf (QMADM)
 - Connect to QM
 - Use Q (RW)
- Encryption of channels
 - SSL

MQI – Message Queue Interface

- The api is implemented on different platform and the call syntax is different in each programming language, but use the same verbs.
- MQCONN – MQDISC
- MQOPEN – MQCLODE
- MQGET – MQPUT

15/OS Configuration

- WRKMQM
- Press F6 or run CRTMQM

Create Message Queue Manager (CRTMQM)

```

Message Queue Manager name . . . > MQCOMMON
Text 'description' . . . . . > 'Common Queue Manager'
Trigger interval . . . . . 999999999 0-999999999
Undelivered message queue . . . *NONE
Default transmission queue . . . *NONE
Maximum handle limit . . . . . 256 0-999999999
Maximum uncommitted messages . . 10000 1-999999999
Default Queue Manager . . . . . *NO *YES, *NO
Queue Manager Library . . . . . *AUTO Name, *AUTO
ASP Number . . . . . *SYSTEM 1-32, *SYSTEM
Journal receiver threshold . . . *DFT 100000-1000000000,
*DFT..

```

15/OS Configuration

- Library QMMQCOMMON created
- Start queue manager (option 14)

```

Work with queue managers LUPROI02
22/09/14 18:29:17
Type options, press Enter.
 2=Change  4=Delete  5=Display  8=Status  14=Start  15=End
18=Work with Queues  19=Work with Processes  20=Work with Channels ...

Opt  Name                               Status  Default
14  MQCOMMON                             ACTIVE  NO

```

I5/OS Configuration

- option 18 work with queues
- Create queue (Press F6)

```
Create MQ Queue (CRTMQMQ)
Type choices, press Enter.
Queue name . . . . . TRANSACTIONS
Queue type . . . . . *LCL          *ALS, *LCL, *MDL, *RMT
Message Queue Manager name . . . > 'MQCOMMON'
```