Introduction to MQ

John Waldron

IBM

November 2018

Session AJ

Agenda

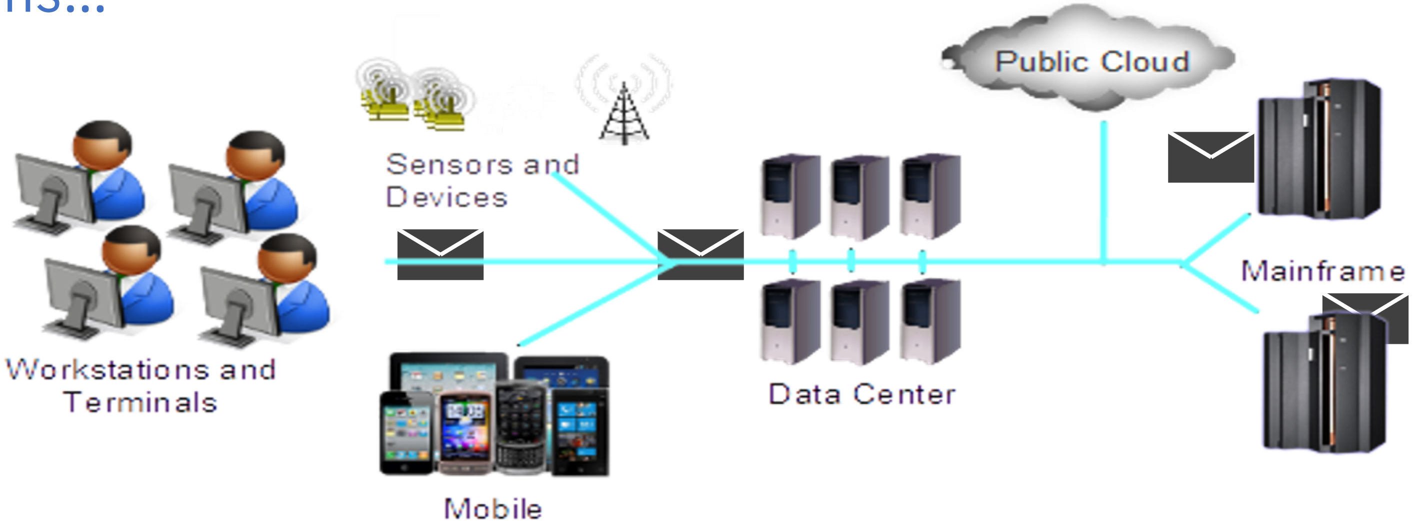
- Messaging
  - What is messaging and why use it?
  - What does MQ give you?

- Fundamentals of IBM MQ
  - Messaging models
  - Key components
  - Messaging applications
  - MQ Environments
  - Security
  - Reliability and availability
  - Administration
  - MQ Advanced

What is messaging?

It connects your applications…

From the simplest pairs of applications…

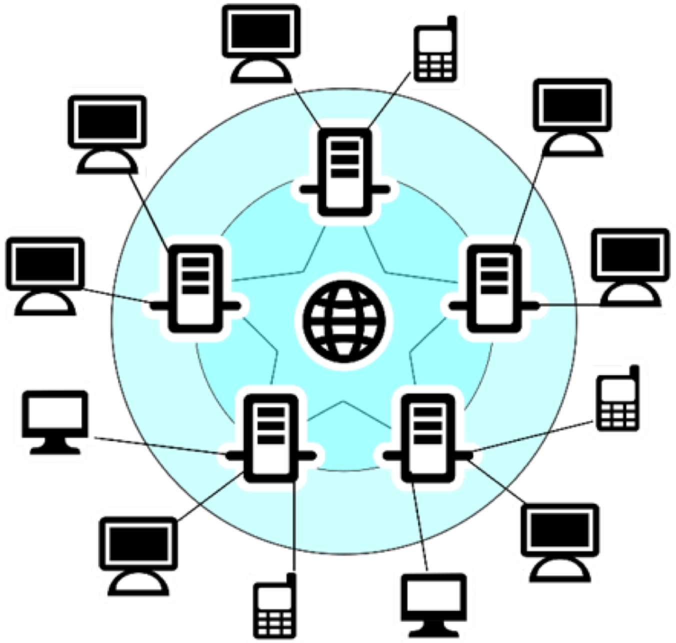


…to the most complex business processes.

…and breaks the tight coupling…

Why use it?

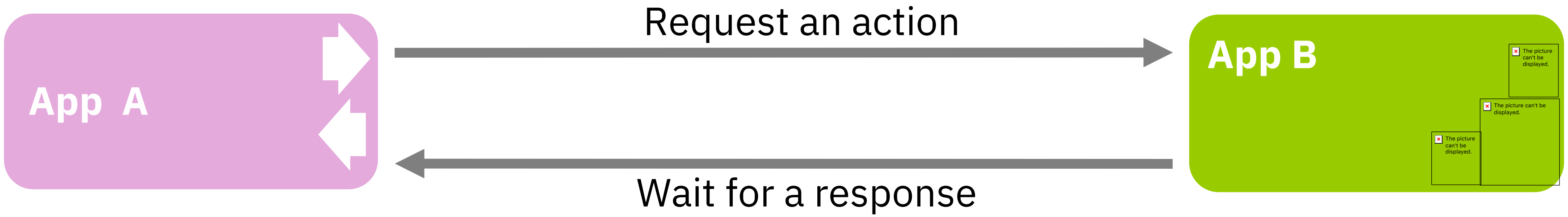
- Extended reach
- Reliability
- Scalability
- Flexibility



- Provides for simplification of application development
  - Ubiquity
  - Easy to change and scale
  - Focus on the business logic

- Important regardless of the initial scale of deployment

Direct communication between applications



App A

Request an action

App B

Wait for a response

The picture can't be displayed.
The picture can't be displayed.
The picture can't be displayed.

- Issues with this ‘synchronous’ approach
  - Both applications A and B **must always** be available for A to continue
  - A cannot do anything whilst B is processing A’s request
  - What if B fails whilst A is waiting for it to complete?
  - What if B needs to handle a high workload of different priority requests?

Fragility of tight coupling

As systems become more tightly coupled, their reliance on each other increases

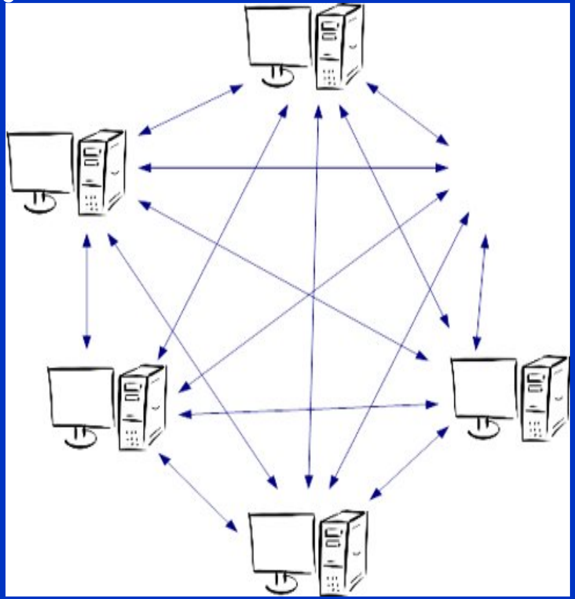
The cost of a failure of a process increases

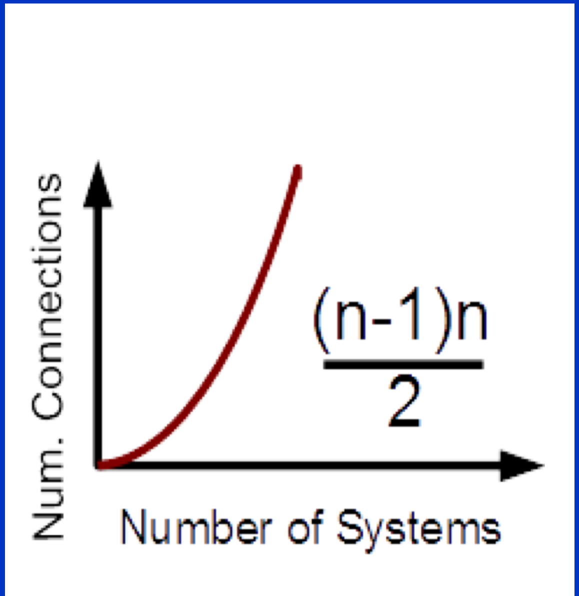
Maximum number of connections goes up with the square of the number of systems

Scaling systems independently to respond to requirements becomes unmanageable

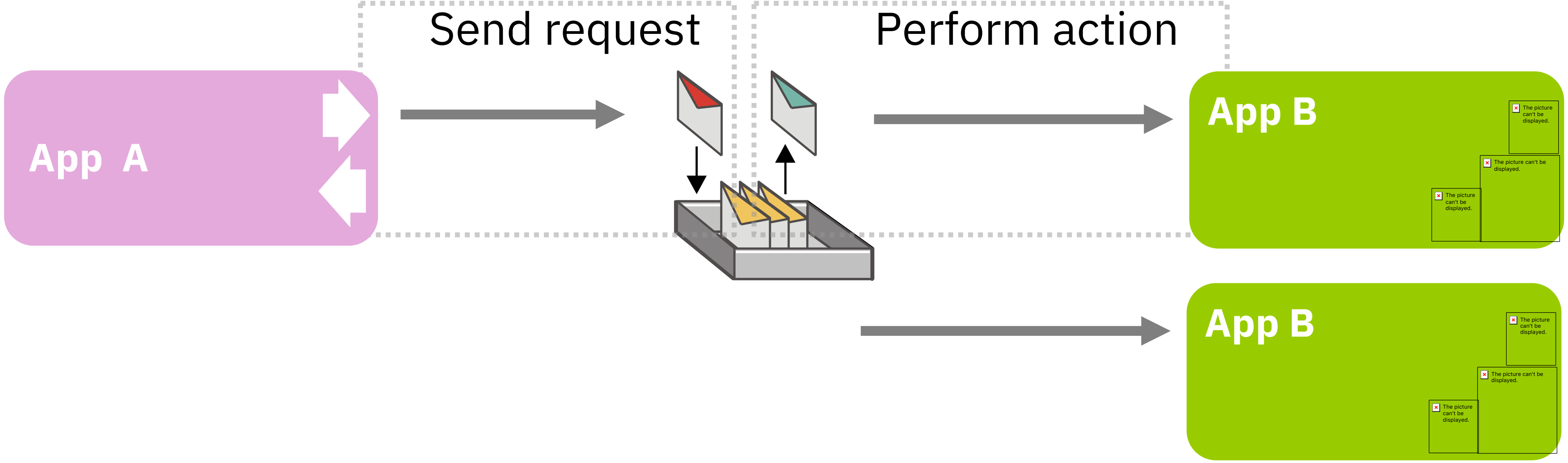
A process was originally designed for one, well-defined, purpose…it then needed to change to meet new requirements

Being able to respond rapidly to internal and external challenges by rapidly modifying existing services gives a competitive advantage





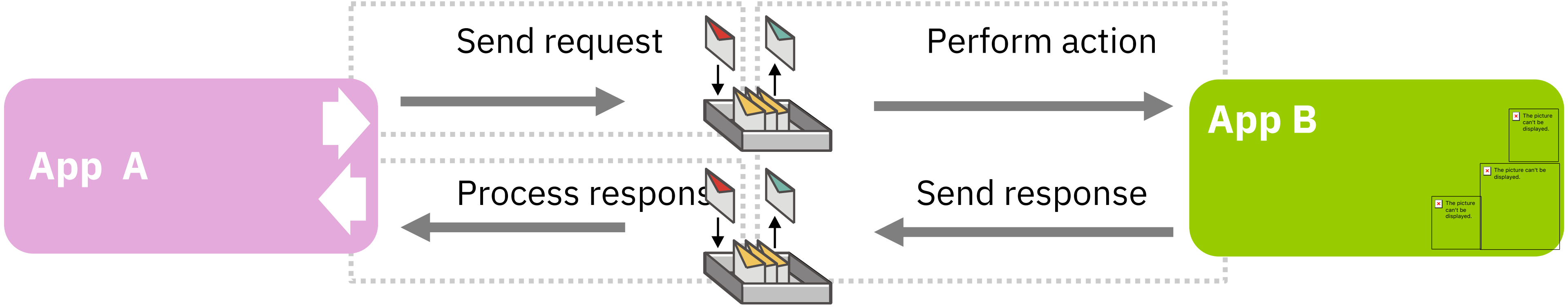
Adding flexibility with Messaging



A ‘queue’ is placed between the two applications

- Allows A to continue immediately, without waiting for B
- Allows B to efficiently process a queue of work when it is ready to do so.
  - ➢ Additional instance of B can be started to handle additional load during peak times
- Overcomes availability of B versus A – “store and forward” of messages

What if you NEED a response?

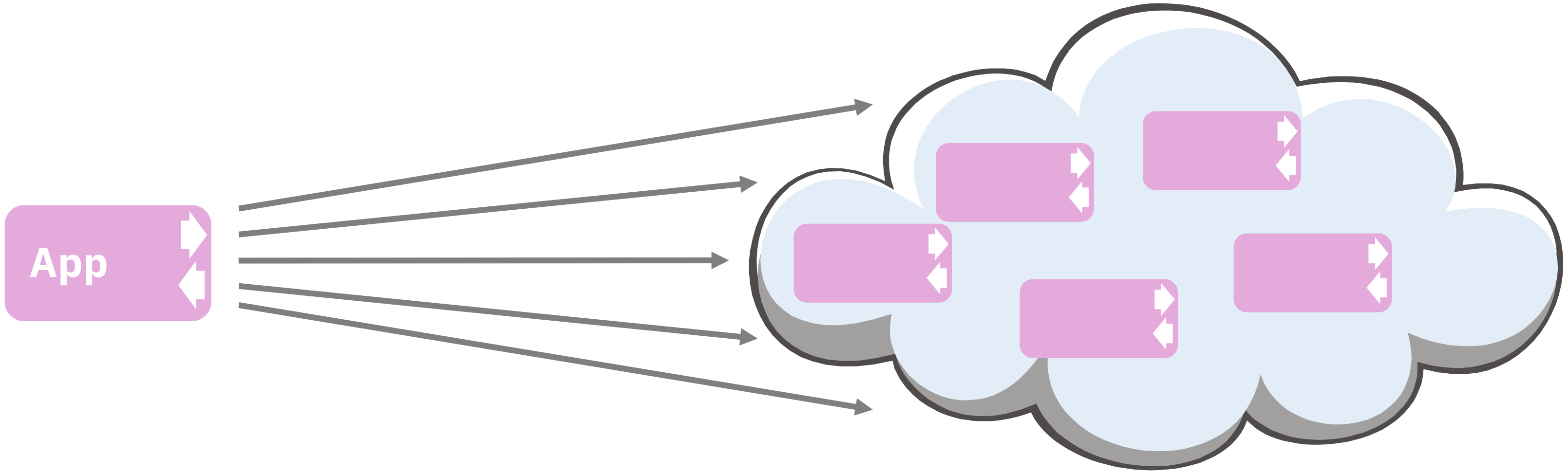

Send request
Perform action
App A
App B
The picture can't be displayed.
The picture can't be displayed.
The picture can't be displayed.
Process respons
Send response

Using messaging still adds value!

- Process the response whenever it becomes available
  - *No need for A to be idle whilst the request is performed*
- Application B processes its workload efficiently and can handle spikes in load
- Application, network and infrastructure failures can be handled independently

Messaging Models:

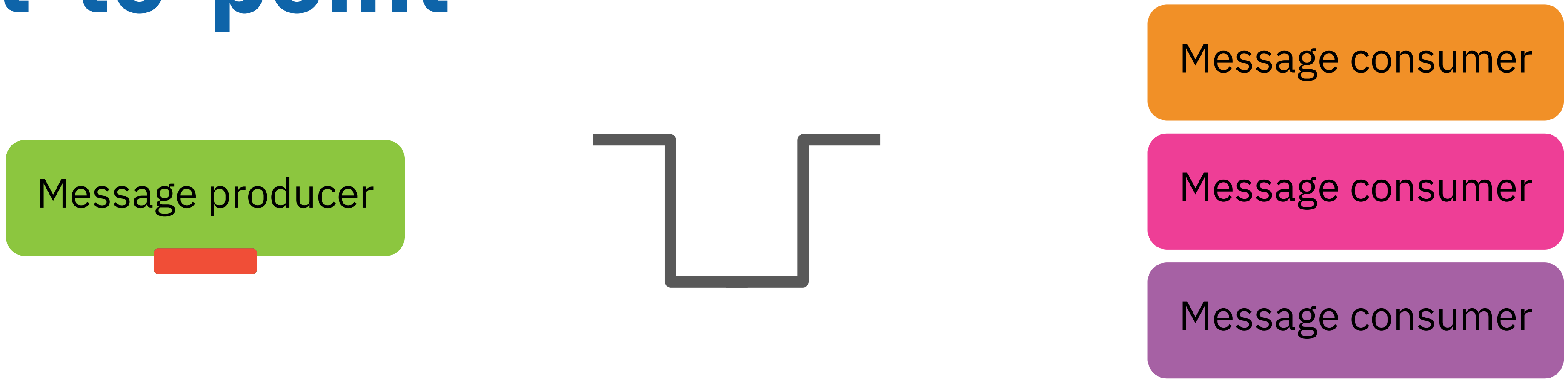
The Power of **events**



Not all information is distributed on a one-to-one basis

- Think about streams of information
  - Regularly updated information - such as stock prices or sensor data
  - Business events - such as ‘new customer’ or ‘purchase’
- **Publish / Subscribe** messaging is the solution!
  - The owner of the information simply ***publishes*** it on a ***topic***
  - Anybody who is interested simply ***subscribes*** to the ***topic***

Point-to-point



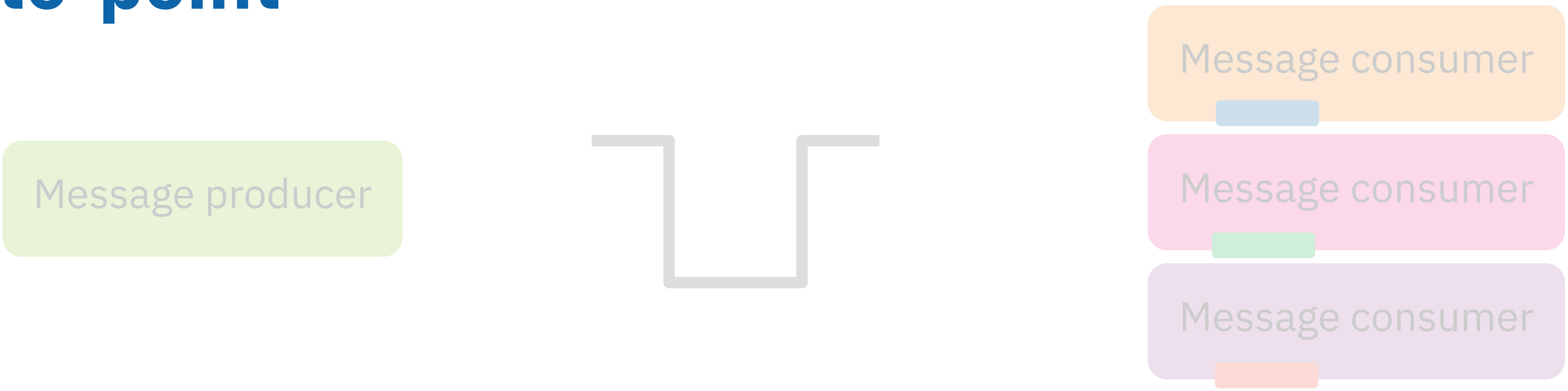
Message producer

Message consumer

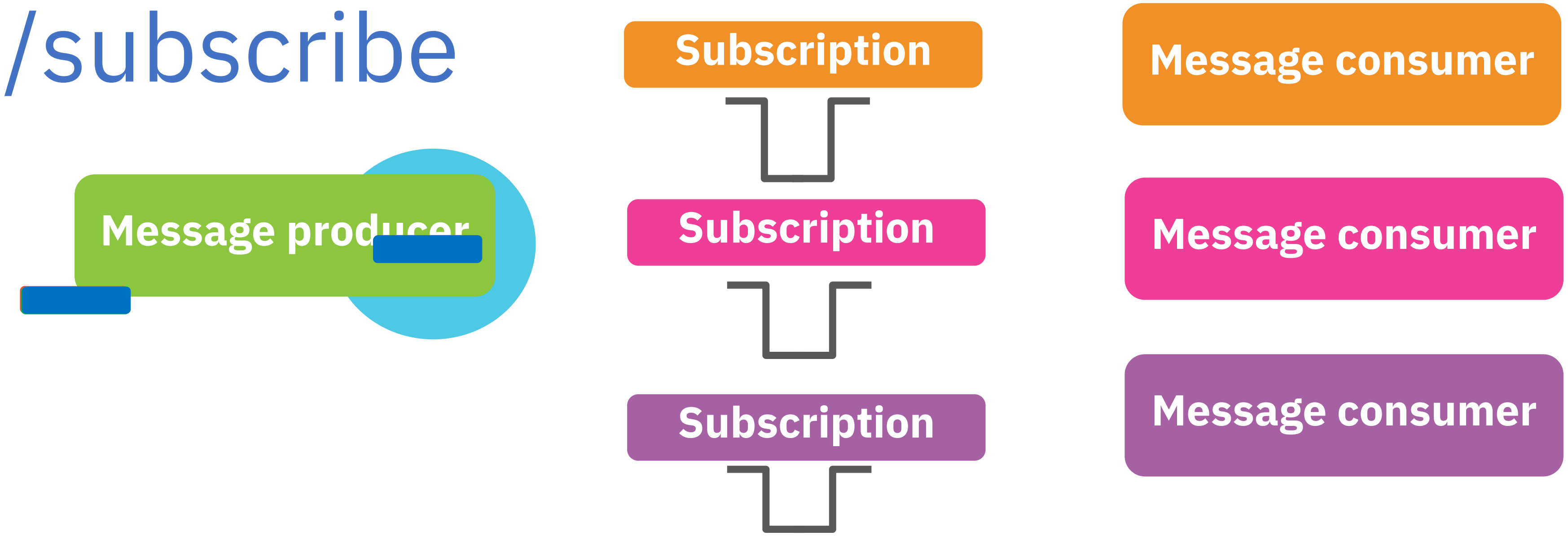
Message consumer

Message consumer

Point-to-point

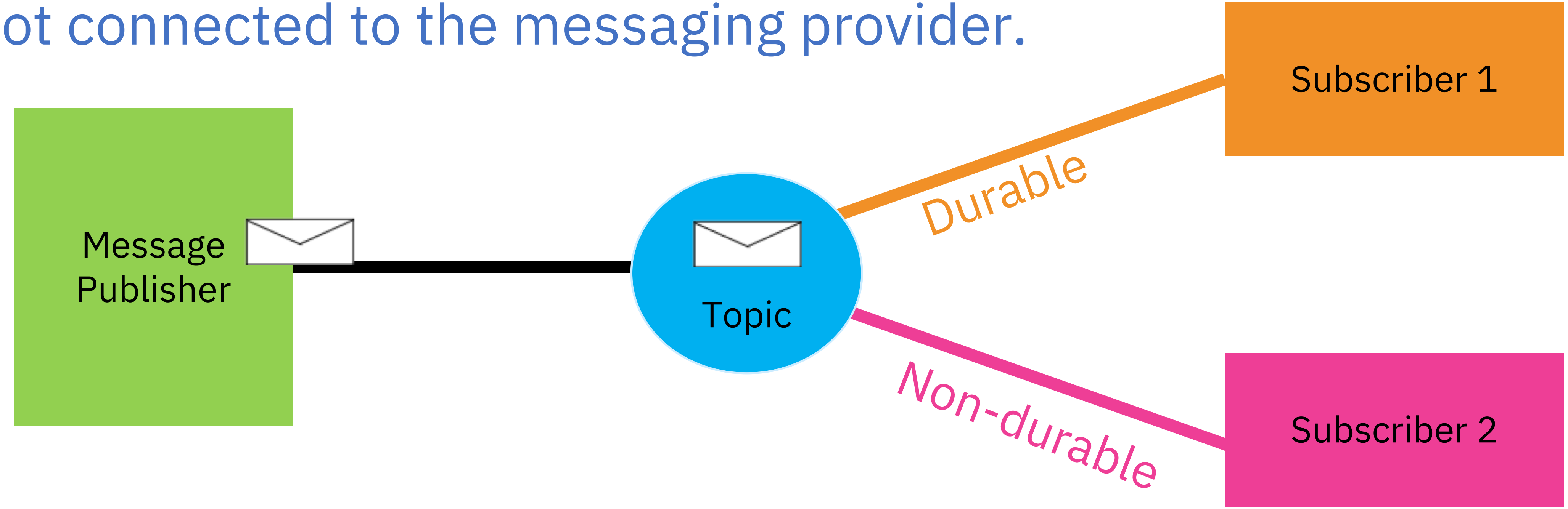


Publish/subscribe



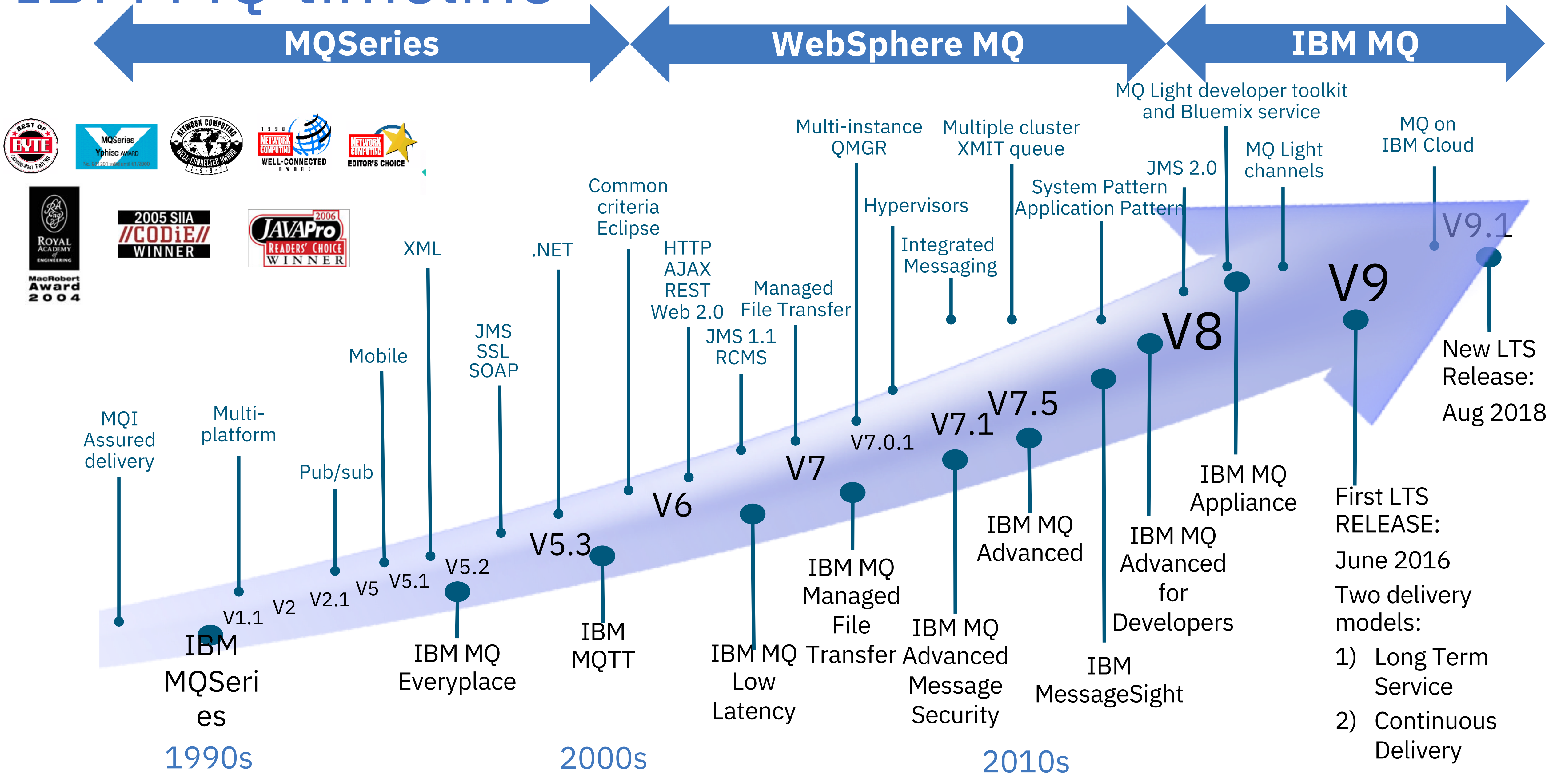
Durable **publish/subscribe** in action

Durable subscriptions result in published messages being retained when the subscriber is not connected to the messaging provider.



IBM MQ

IBM MQ timeline



MQSeries

WebSphere MQ

IBM MQ

MQI Assured delivery

Multi-platform

IBM MQSeries

V1.1

V2

Pub/sub

V2.1

V5

Mobile

V5.1

XML

V5.2

IBM MQ Everyplace

JMS SSL SOAP

.NET

V5.3

IBM MQTT

Common criteria Eclipse

HTTP AJAX REST Web 2.0

V6

JMS 1.1 RCMS

IBM MQ Low Latency

Managed File Transfer

V7

IBM MQ Managed File Transfer

Multi-instance QMGR

V7.0.1

Hypervisors

Integrated Messaging

V7.1

IBM MQ Advanced Message Security

Multiple cluster XMIT queue

V7.5

IBM MQ Advanced

System Pattern Application Pattern

IBM MessageSight

JMS 2.0

V8

IBM MQ Advanced for Developers

MQ Light developer toolkit and Bluemix service

IBM MQ Appliance

MQ Light channels

V9

First LTS RELEASE:

June 2016

Two delivery models:

1) Long Term Service

2) Continuous Delivery

MQ on IBM Cloud

V9.1

New LTS Release:

Aug 2018

1990s

2000s

2010s

What MQ adds to messaging

Enterprise Messaging

Reliability

Assured message delivery “Once and once only”

Resiliency and high availability of the infrastructure

Continued support and interoperability of systems for over twenty years

Scalability

High performance solution

Incremental growth of applications and infrastructure

Ubiquity

Breadth of support for platforms and environments

Multiple application environments and APIs to suit many styles

Security

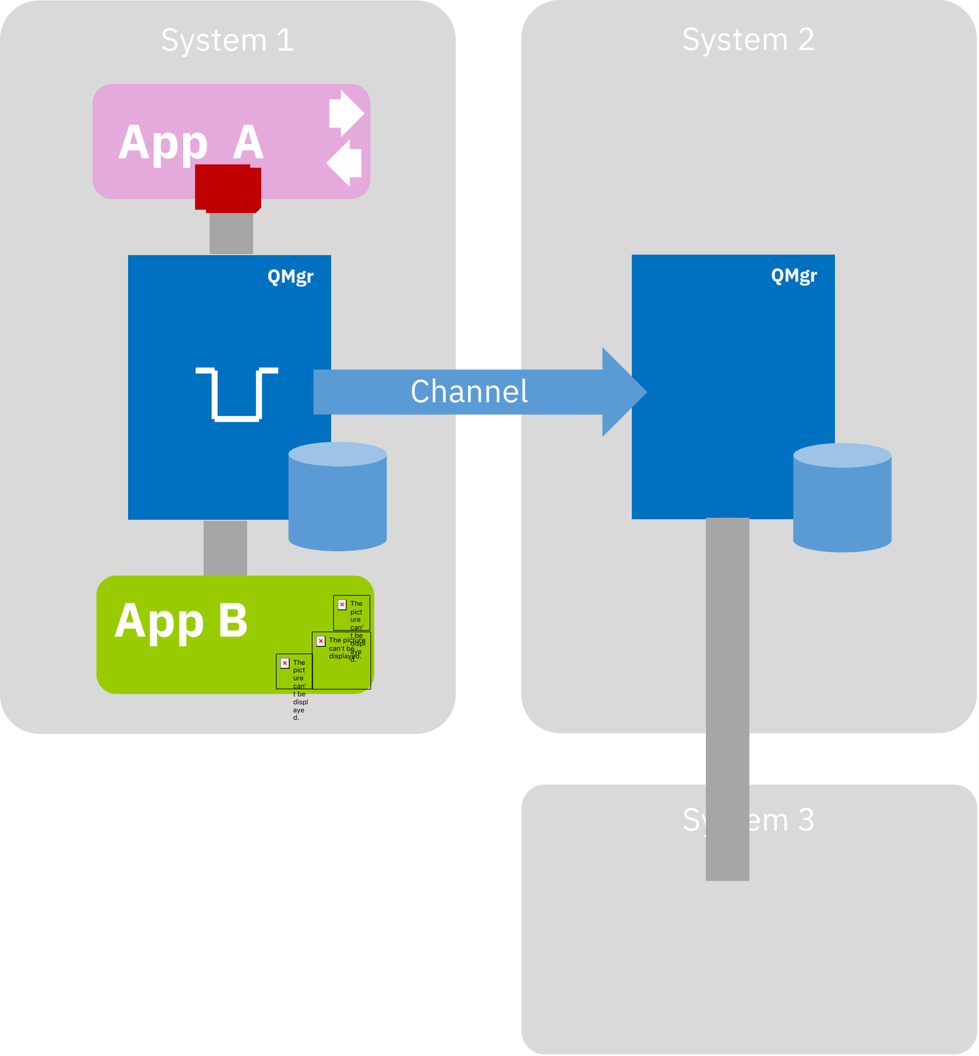
Data encryption and integrity

End use authentication and authorisation

Audit trails for configuration and data flows

Anatomy of an MQ system

- **Applications**
  - Applications use MQ clients to connect to an MQ **queue manager**
  - Applications can connect to queue managers either on the same system (*BINDINGS mode*) or remotely over a network (*CLIENT mode*)
- **Queue Managers**
  - A queue manager is a runtime that hosts messaging resources such as **queues** and their **messages**
  - A queue manager manages the flow and storage of messages
  - Each queue manager runs on a single system
  - Multiple queue managers can be connected together using **channels** and messages routed between them
- **Queues**
  - Queues are a named resource where messages sent to by applications, stored by the queue manager and retrieved by applications
- **Messages**
  - Are just chunks of data
  - Applications build messages to send and receive
- **Channels**
  - Channels define a way for one queue manager to connect to another queue manager
  - Channels can be manually configured or dynamically created as and when needed using **MQ Clusters**



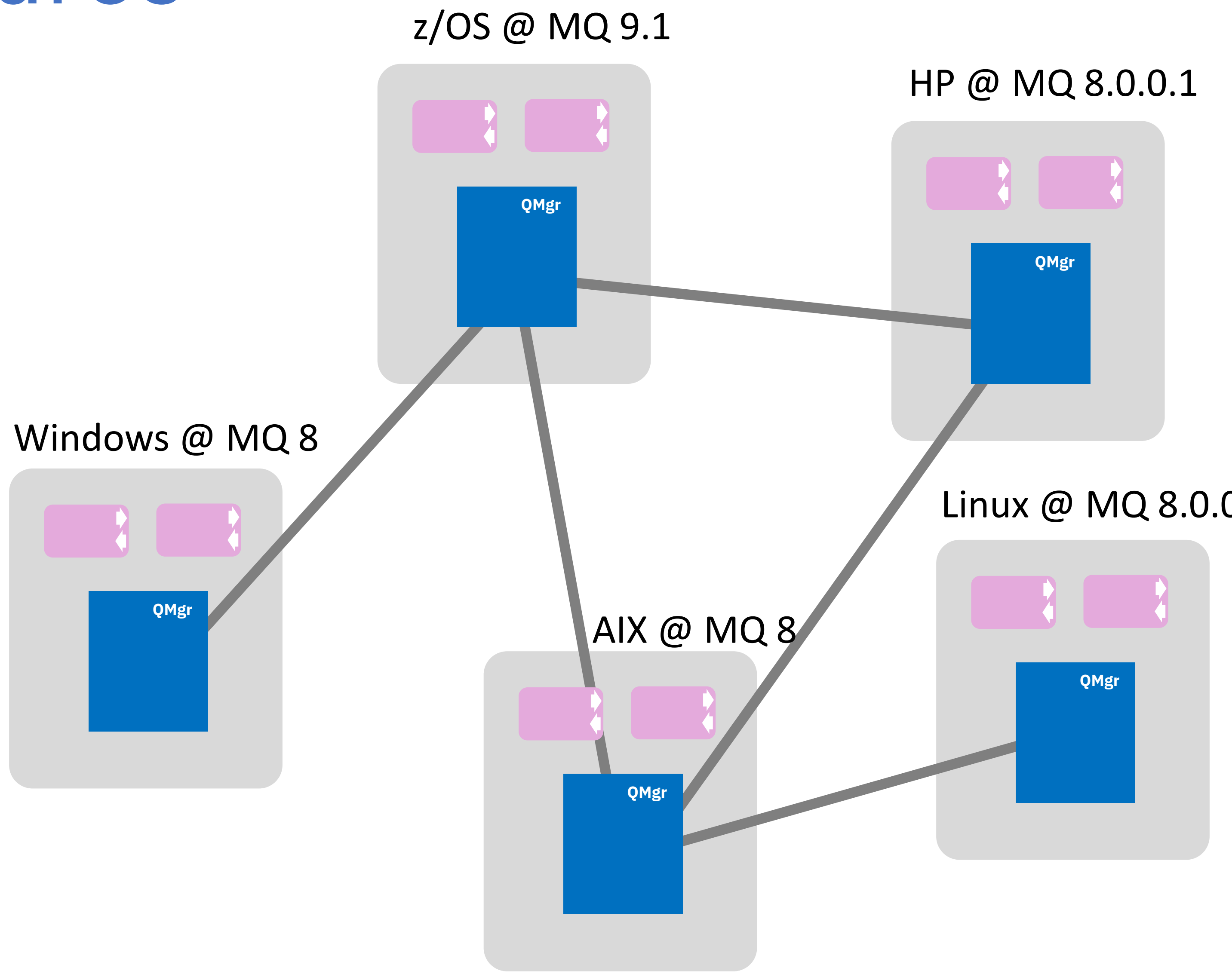
Distributed Architectures

- Used for connectivity of heterogeneous systems

- “Store and Forward” system to account for network outages

- This is the ‘original’ deployment pattern for MQ

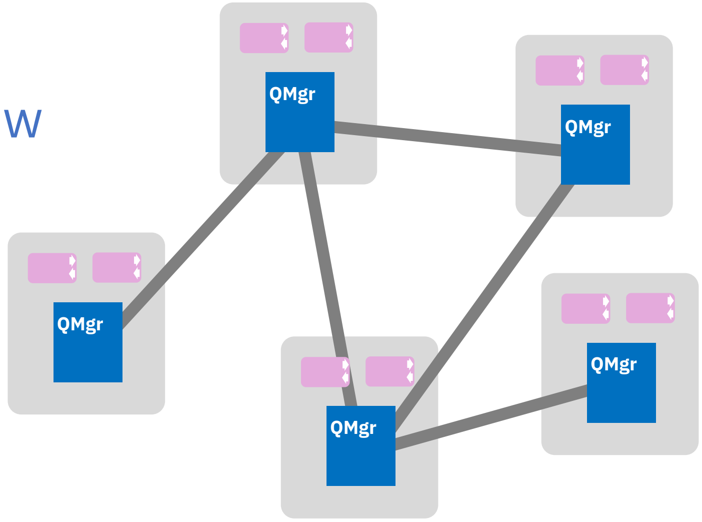
- Queue managers will interoperate with other queue managers and clients at any other version of MQ



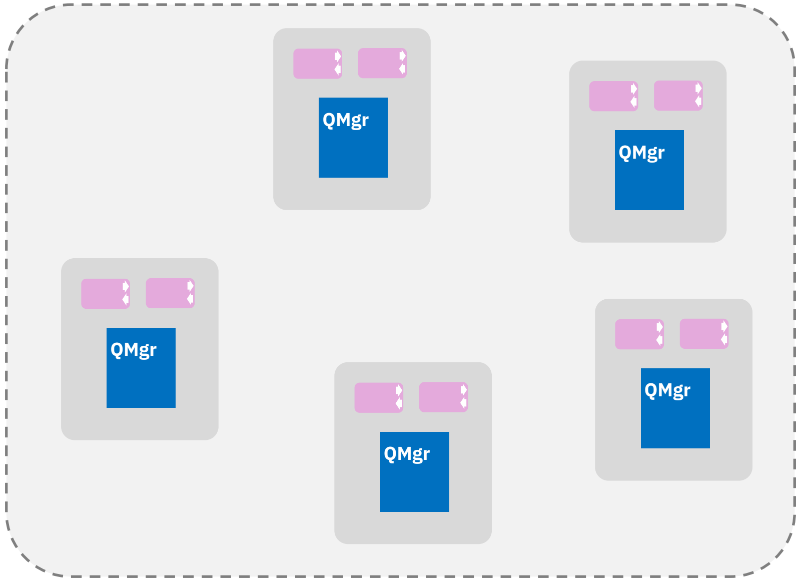
Connecting queue managers together

- **Channels** connect queue managers together, allowing messages to flow between them

- Two options:
  - **Manual configuration of channels**
    - Each channel relationship must be defined on both ends
    - Additional resource also need to be defined (***transmission queues*** and ***remote queues***)



  - **MQ clusters**
    - Once queue managers join a cluster (a pair of special channels must be defined) they can route messages to any other clustered resource in the cluster without requiring further, per queue manager, configuration.
    - As queue manager networks grow, clusters become a benefit
    - Clusters also enable workload balancing and availability routing of messages



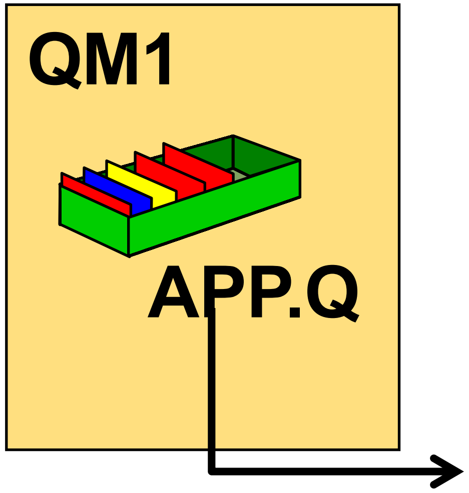
MQ API calls

MQ APIs – How do I connect my apps to my queue manager?

- **MQI**
- **JMS**
- **MQ Light API**
- **MQTT**
- REST API Messaging (point to point only!)

MQ APIs - MQI (MQ Interface)

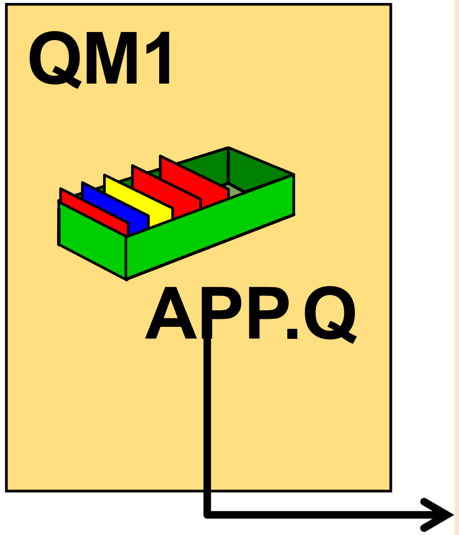
- C, COBOL, Java
- MQ’s proprietary API offering full access to MQ’s capabilities



```
MQCONN(QM)
…
MQOPEN(Q)
…
MQPUT(Q)
…
MQGET(Q)
…
MQCLOSE(Q)
…
MQDISC(QM)
```

MQ APIs - Java Message Service (JMS/XMS)

- JMS is part of the JEE specification.
  - Fully supported in application servers such as WSAS, Liberty, WebLogic and more
- Simplifies programming for Java developers
- No MQ coding knowledge needed!
- XMS syntactically the same as JMS V1.1 but for C, C++ and C#



```
// Lookup the MQ specific objects in JNDI
Context jndiContext         = new InitialContext();
ConnectionFactory cf        = (ConnectionFactory) jndiContext.lookup("jms/QM1");
Destination dest            = (Destination) jndiContext.lookup("jms/APP.Q");

// Establish a connection with the queue manager & create JMS objects
JMSContext context          = cf.createContext();
JMSConsumer consumer        = context.createConsumer(dest);

// Get a message
Message msg = consumer.receive();
```

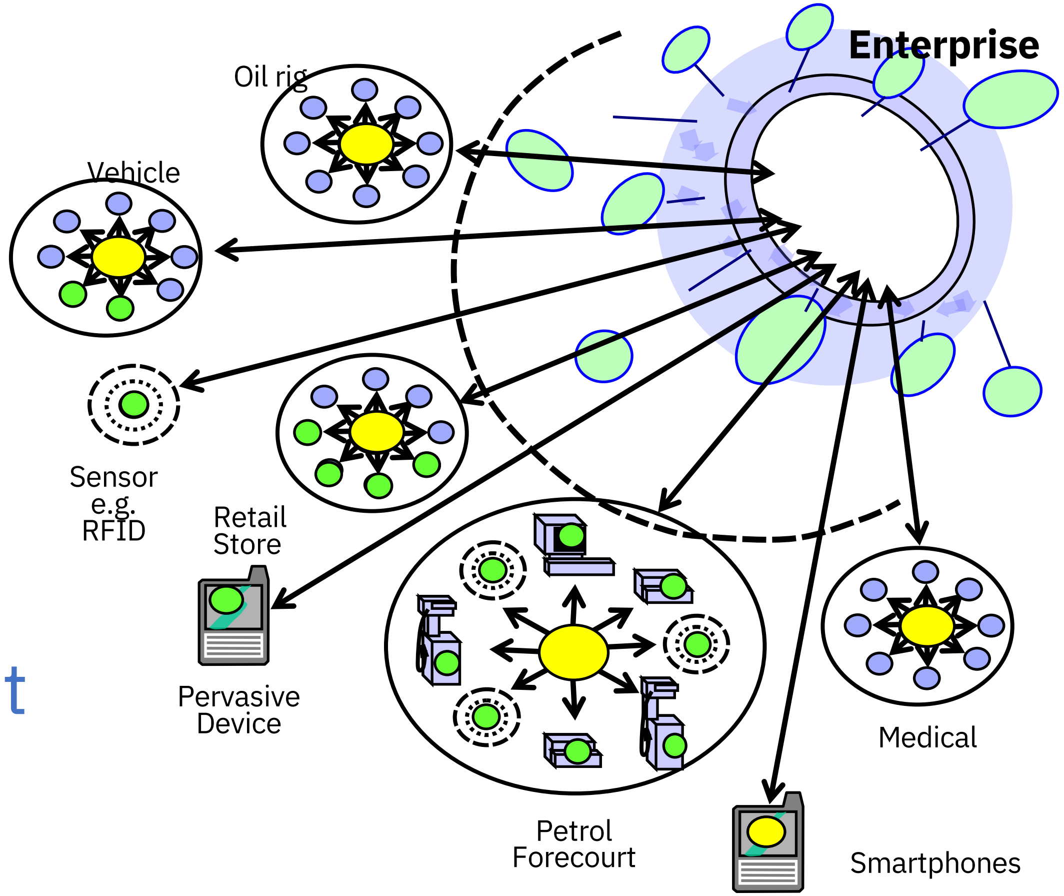
MQ APIs - MQ Light

- AMQP based API
- Node.js, Java, Ruby
- Connects cloud applications to MQ!

```
# Receive:
var mqlight = require('mqlight');
var recvClient = mqlight.createClient({service:
'amqp://localhost'});
recvClient.on('started', function() {
recvClient.subscribe('news/technology');
recvClient.on('message', function(data, delivery) {
          console.log(data);
     });
});
```

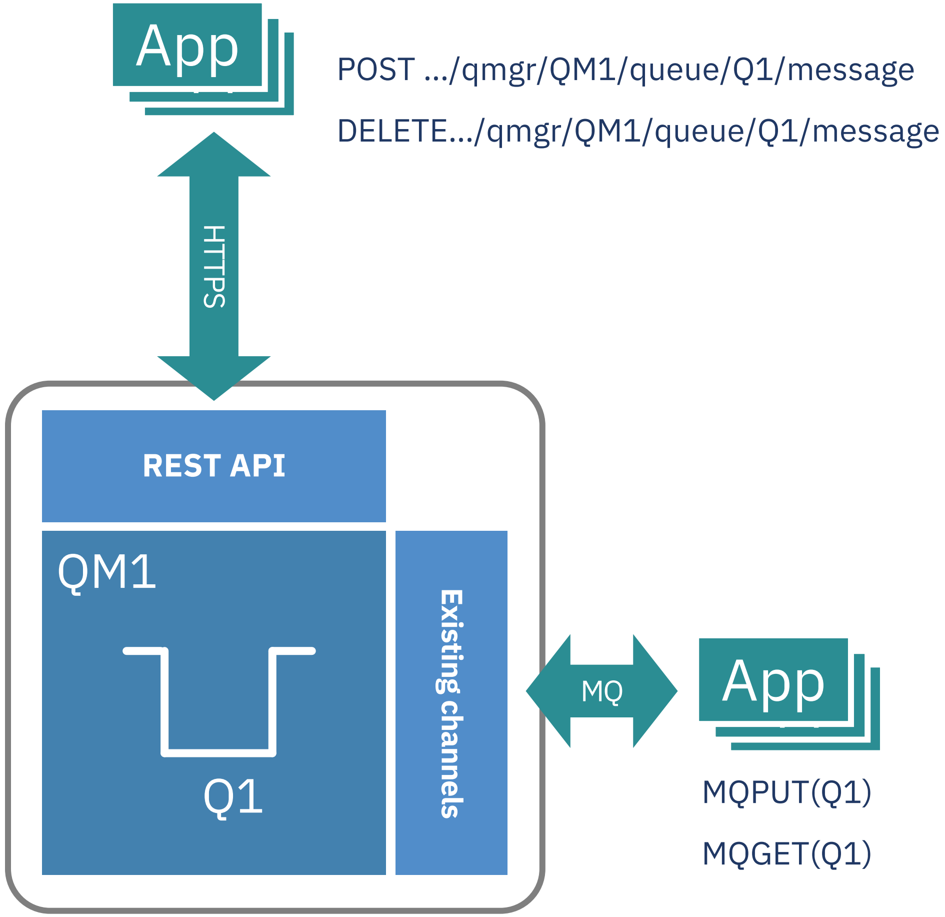
MQ APIs - MQ Telemetry (MQTT)

- Product extension supports connectivity for smart devices to the enterprise
- Utilises the open standard MQTT protocol
    - a lightweight, public, low bandwidth messaging protocol for scenarios where enterprise messaging clients are too big or bandwidth intensive.
- Java, C and JavaScript libraries provided, but you can “roll your own” that implement the MQTT v3 spec



REST Messaging

- The new HTTP server support in MQ 9.0.x provides the platform for a properly integrated REST API solution
- Allowing applications to put and get messages from a queue without installing any MQ software locally
- Ideal for environments with native REST support, such as common JavaScript libraries including NodeJS, and AngularJS
- Can only be used for point to point messaging
- For full functionality and resiliency an MQ client should still be used



POST .../qmgr/QM1/queue/Q1/message

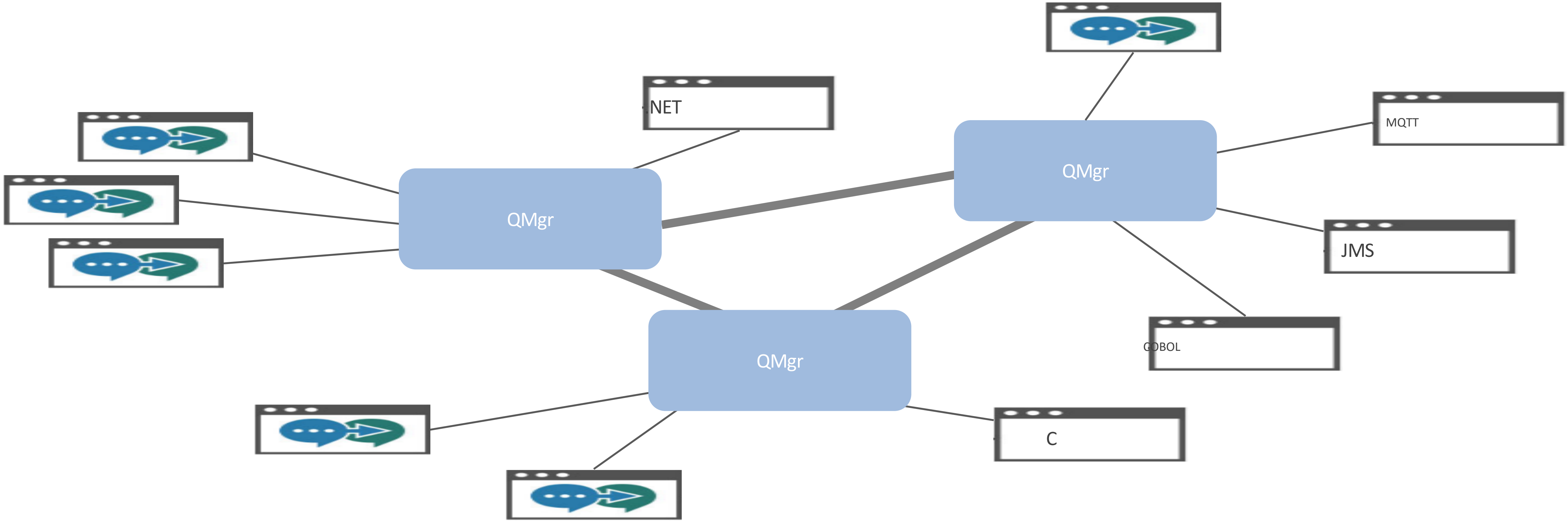
DELETE.../qmgr/QM1/queue/Q1/message

MQPUT(Q1)

MQGET(Q1)

Messaging APIs

- All interoperate with each other!
  - Any application can receive messages from any other application



“Once and once only delivery”

Message persistence

- **Persistent messages**
  - Stored to disk
  - Queued messages are recovered following a server failure
    - No matter what the failure, as long as the disks are intact, so will your messages be
- **Non-persistent messages**
  - Kept in memory as much as possible (better performance)
  - Queued messages are lost in the event of a server failure or restart

Transactions

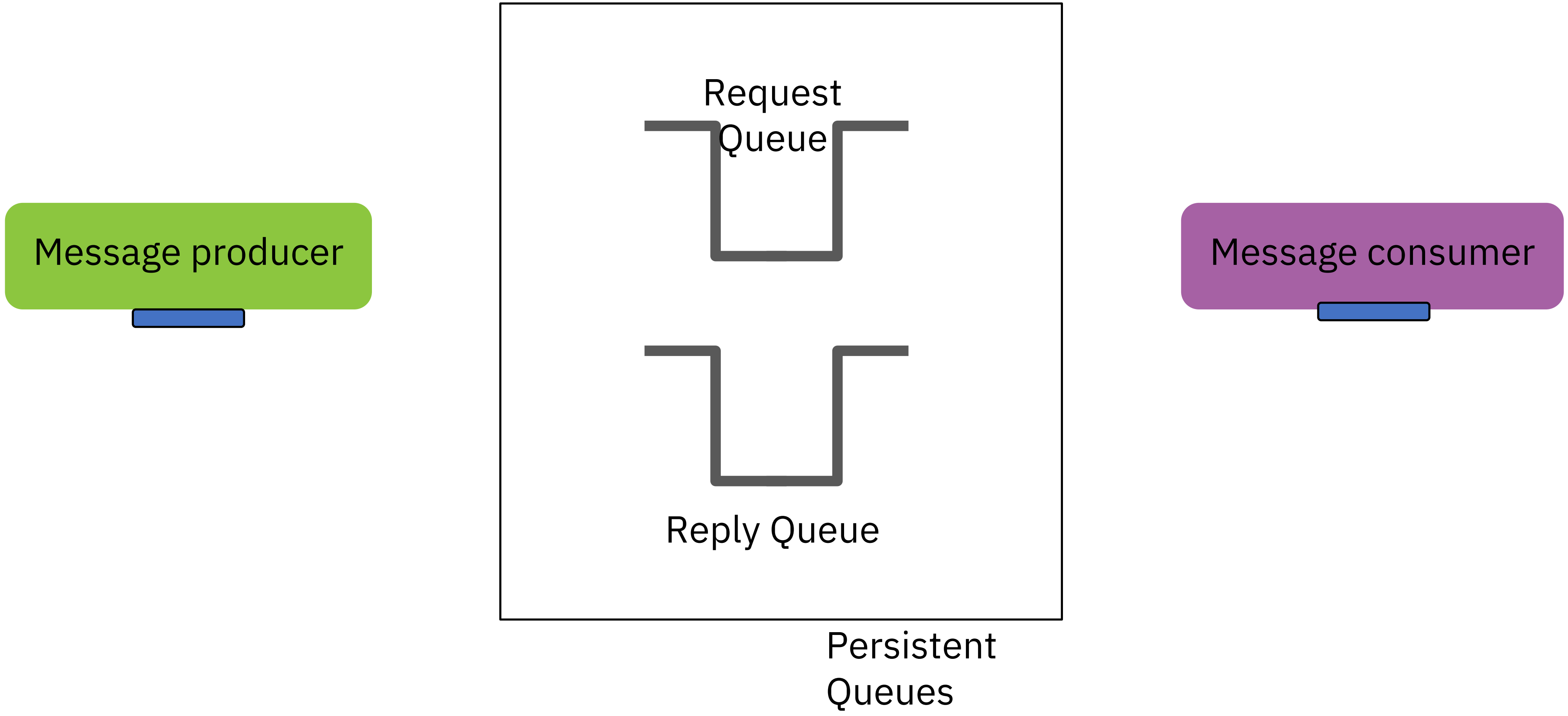
- Multiple messaging operations can be coordinated as a transaction
- Messaging applications are often updating other resources based on messages
  - E.g. Receive a message, insert the data to a database
- MQ applications can coordinate messaging operations with other transactional resources
  - A queue manager can be an XA transaction coordinator
  - Or coordinated externally, for example a JEE application server such as WebSphere Application Server
- Available in MQI, JMS and XMS APIs

- *Combining persistent messages with transactions gives you once and once only delivery of messages from an application’s point of view*

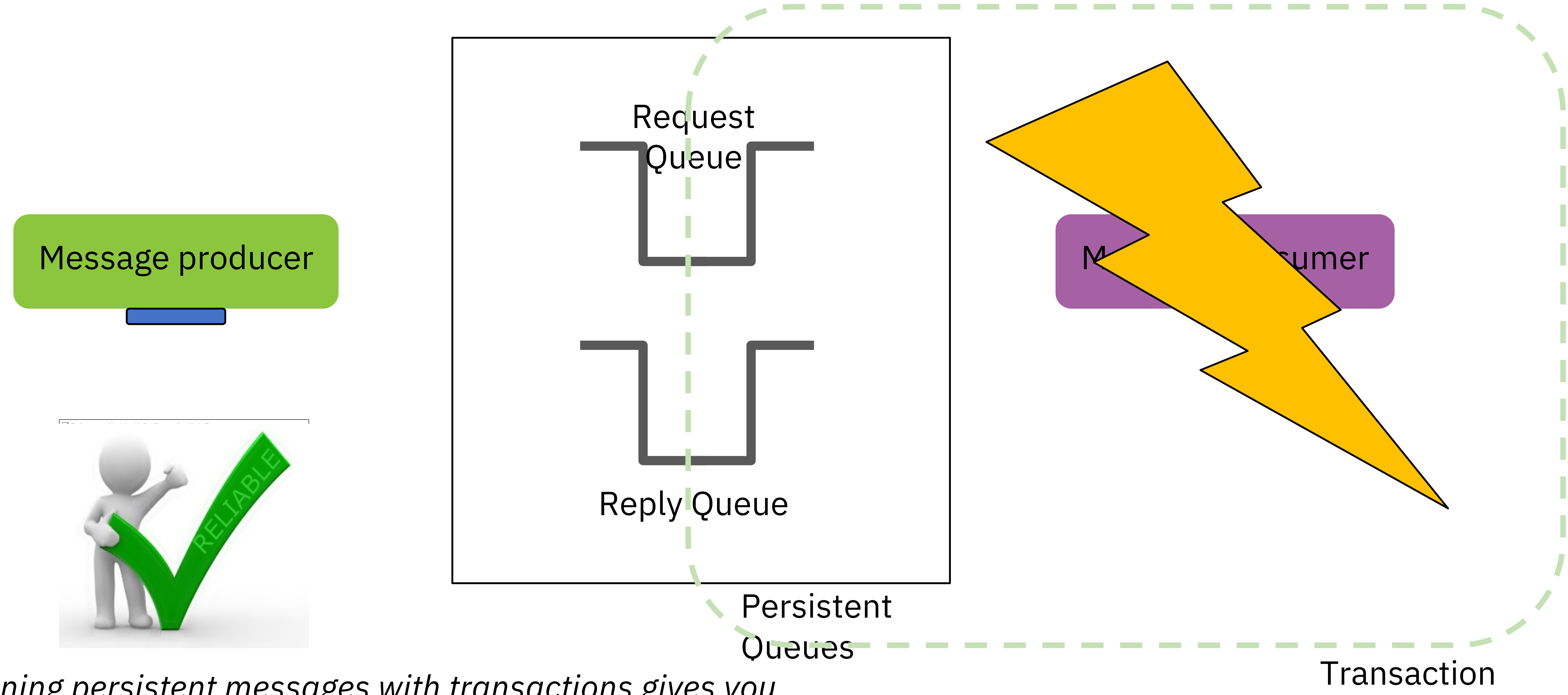
Transactional Messaging

- Non Persistent

- Persistent



Transactional Messaging



Combining persistent messages with transactions gives you once and once only delivery of messages from an application's point of view

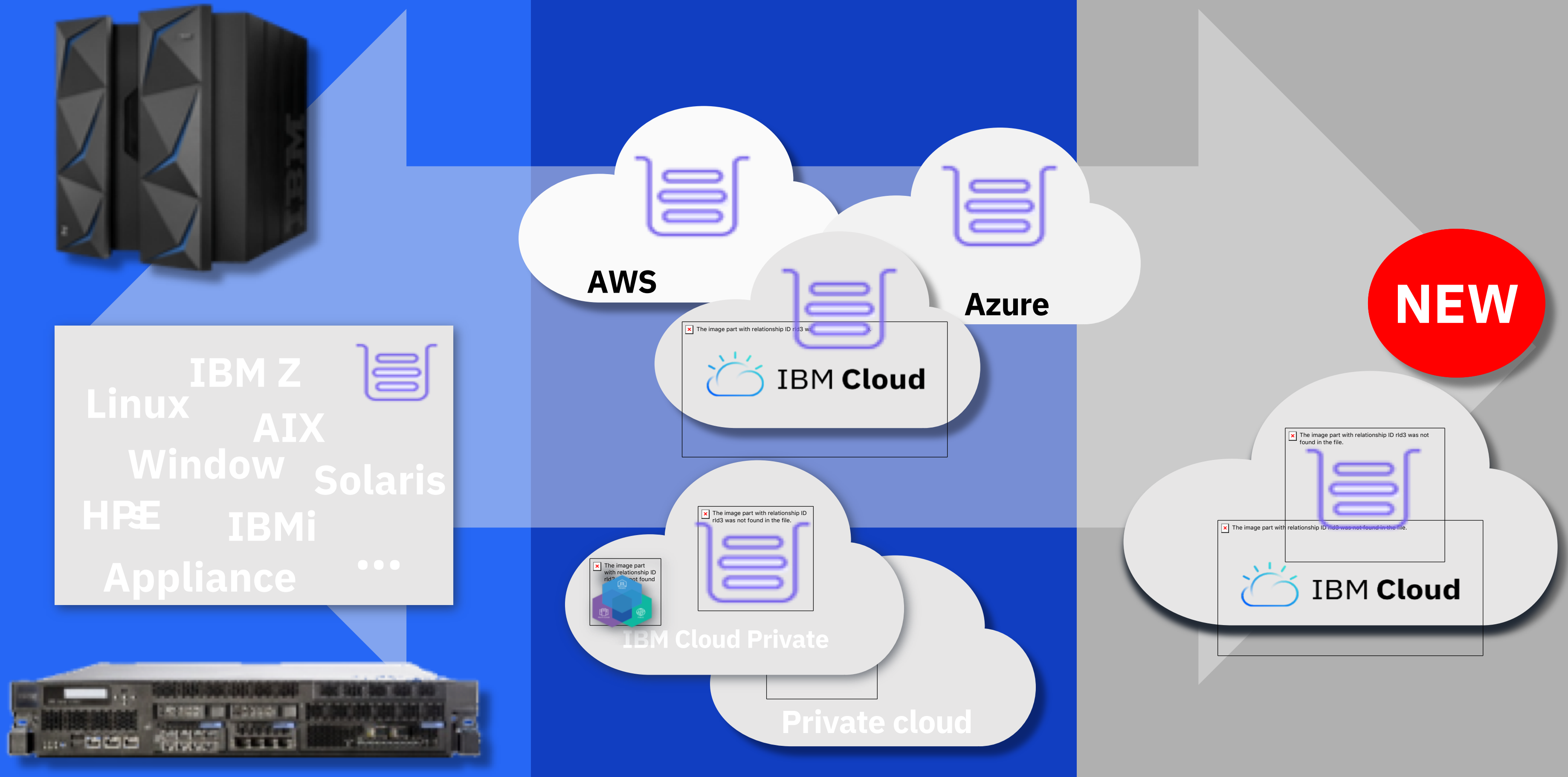
- Run IBM MQ in any location or cloud exactly as you need it

- On-premise, software and the MQ Appliance

- Run it yourself in any cloud, public or private

- Let IBM host it for you with its new managed MQ service in IBM Cloud

IBM MQ



MQ on IBM Cloud

Provision queue managers directly into IBM Cloud

IBM owns the infrastructure and the responsibility to keep the systems up to date and running

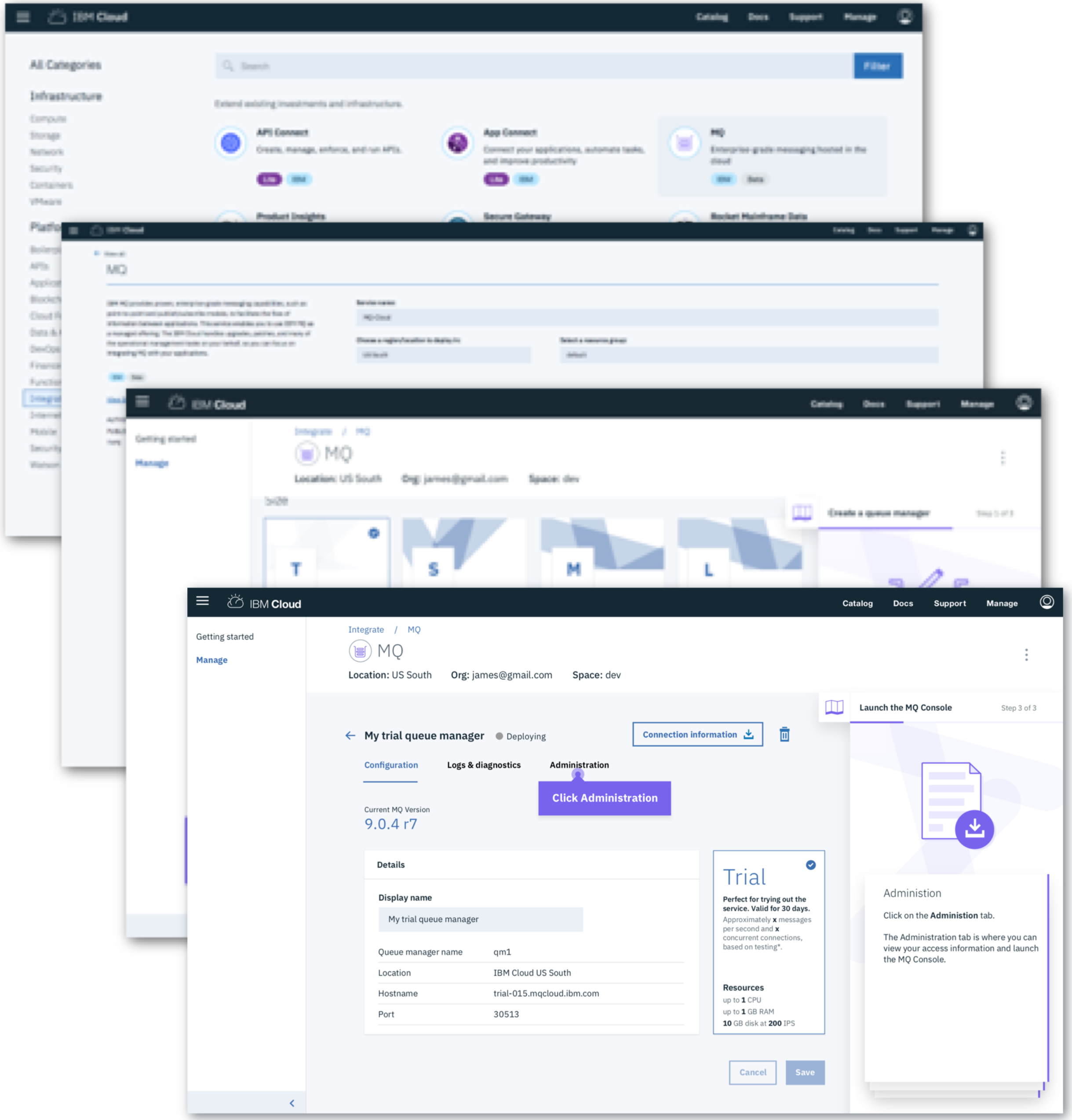
The customer owns the configuration and the monitoring of the messaging

Try the service for free at:

console.bluemix.net/catalog/services/mq

Hosted on

IBM Cloud



Location: US South Org: james@gmail.com Space: dev

My trial queue manager ● Deploying

Current MQ Version
9.0.4 r7

Details

Display name: My trial queue manager

Queue manager name qm1

Location IBM Cloud US South

Hostname trial-015.mqcloud.ibm.com

Port 30513

Trial

Perfect for trying out the service. Valid for 30 days. Approximately x messages per second and x concurrent connections, based on testing*.

Resources
up to 1 CPU
up to 1 GB RAM
10 GB disk at 200 IPS

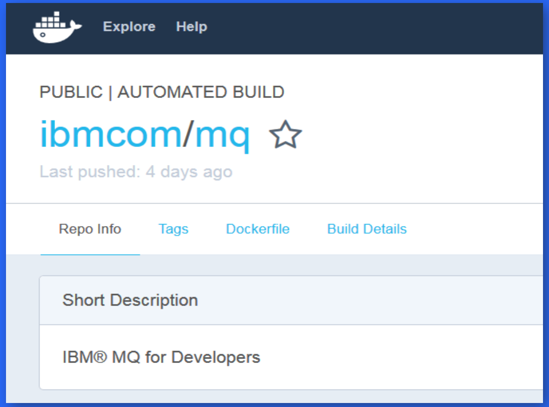
Administion

Click on the Administion tab.

The Administration tab is where you can view your access information and launch the MQ Console.

MQ in Containers

- MQ has been supporting Docker containers since 2015 with images on **Docker Hub** and **Docker Store** and sample setups on **Github**



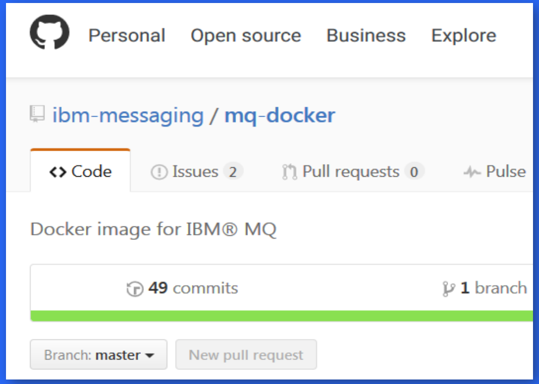
PUBLIC | AUTOMATED BUILD

ibmcom/mq ☆

Last pushed: 4 days ago

Short Description

IBM® MQ for Developers

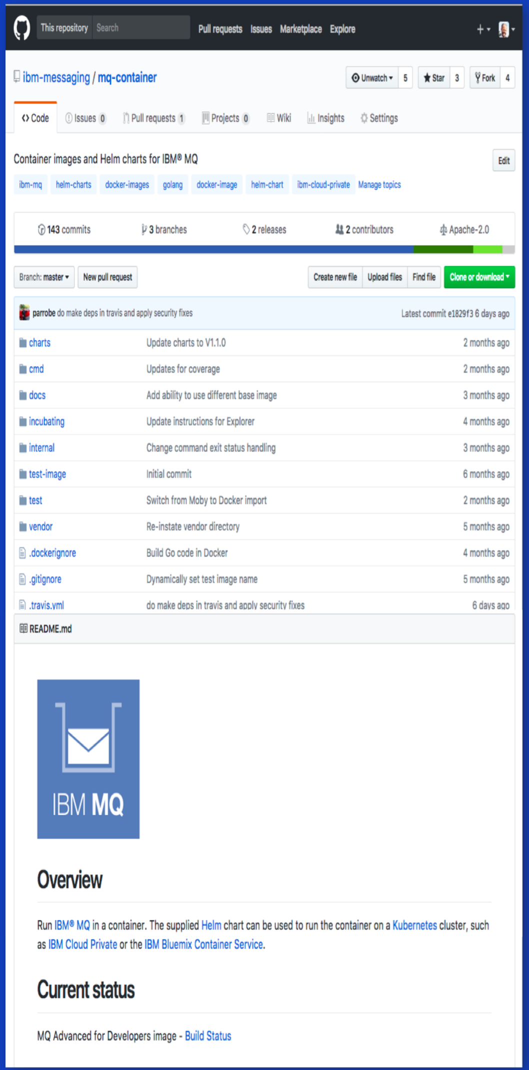


ibm-messaging / mq-docker

Docker image for IBM® MQ

49 commits · 1 branch

- More recently it has been demonstrating how to get the most from containers using **Kubernetes**



ibm-messaging / mq-container

Container images and Helm charts for IBM® MQ

143 commits · 3 branches · 2 releases · 2 contributors · Apache-2.0

parrobe do make deps in travis and apply security fixes — Latest commit e1829f3 6 days ago

charts	Update charts to V1.1.0	2 months ago
cmd	Updates for coverage	2 months ago
docs	Add ability to use different base image	3 months ago
incubating	Update instructions for Explorer	4 months ago
internal	Change command exit status handling	3 months ago
test-image	Initial commit	6 months ago
test	Switch from Moby to Docker import	2 months ago
vendor	Re-instate vendor directory	5 months ago
.dockerignore	Build Go code in Docker	4 months ago
.gitignore	Dynamically set test image name	5 months ago
.travis.yml	do make deps in travis and apply security fixes	6 days ago

README.md

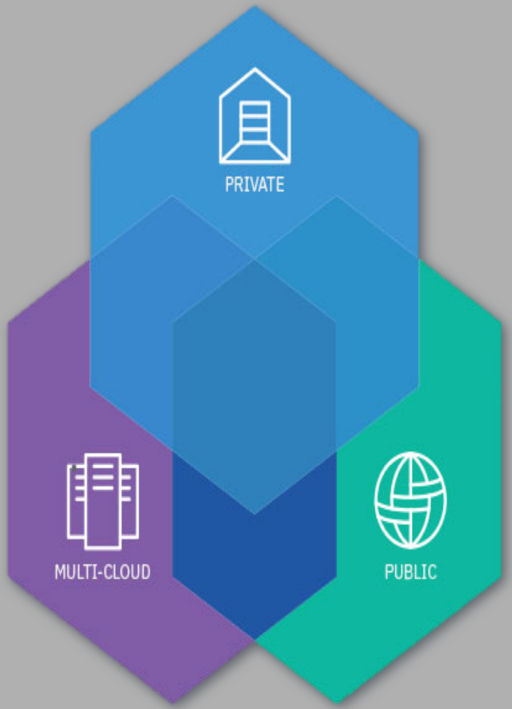
Overview

Run IBM® MQ in a container. The supplied Helm chart can be used to run the container on a Kubernetes cluster, such as IBM Cloud Private or the IBM Bluemix Container Service.

Current status

MQ Advanced for Developers image - Build Status

- And now MQ Advanced is available as a fully supported product with **IBM Cloud Private**, a Kubernetes-based solution from IBM



MQ Highly Available

IBM MQ HA capabilities

- Support for HA clusters and network storage
- Multi-instance queue managers (Windows, Linux, UNIX)
- IBM MQ Appliance

Client connectivity

- Automatic reconnection
- CCDTs
- Pre-connect exit

IBM **MQ**

- Replicated Data Queue Managers (Linux)
- Queue-sharing groups (z/OS)
- Support for cloud orchestration frameworks
  e.g. Kubernetes, Docker Swarm, Apache Mesos

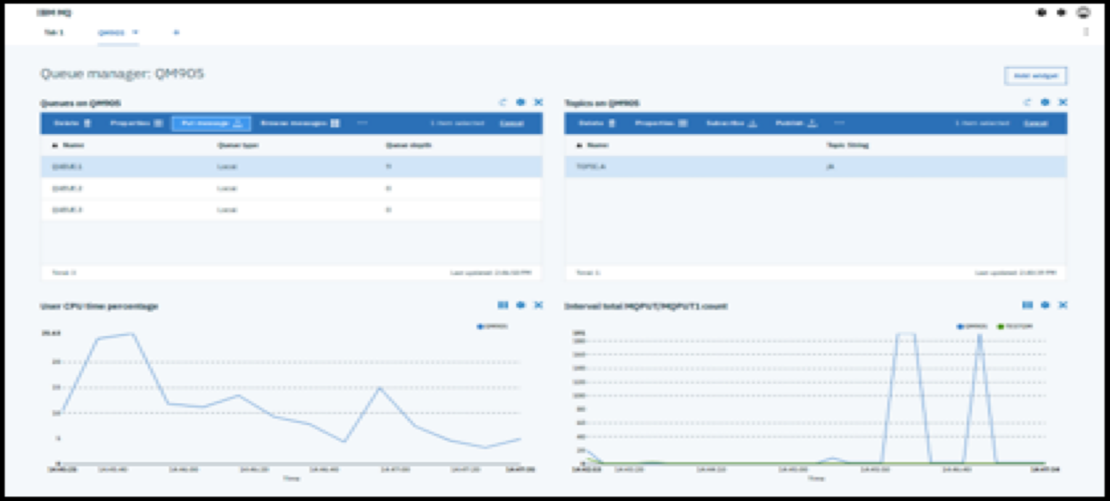
MQ Administration

Administration and monitoring

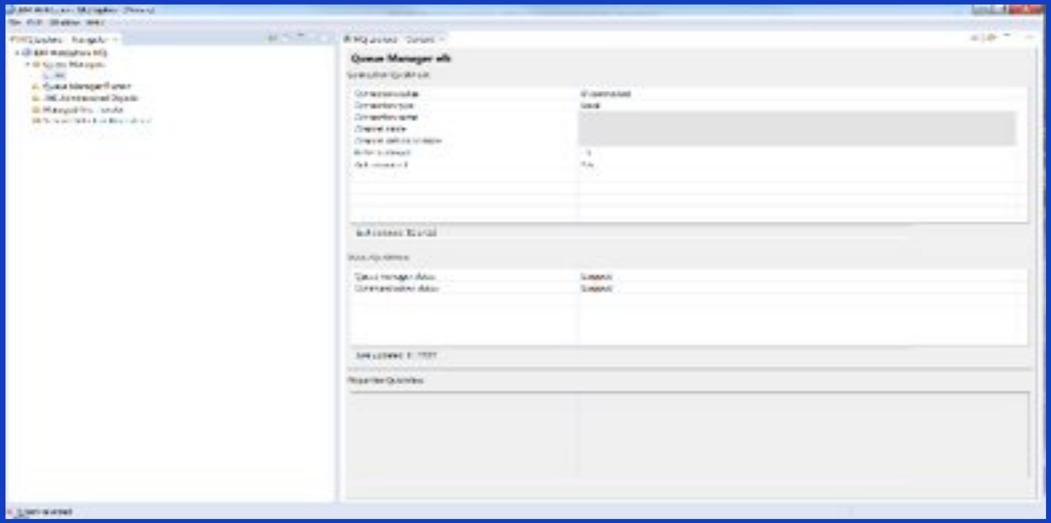
- Command line
- Scripting
- Programmatic APIs
- REST API

- Tivoli and third-party tooling

- Web console



- GUI tooling



AMS & MFT

MQ Advanced Message Security

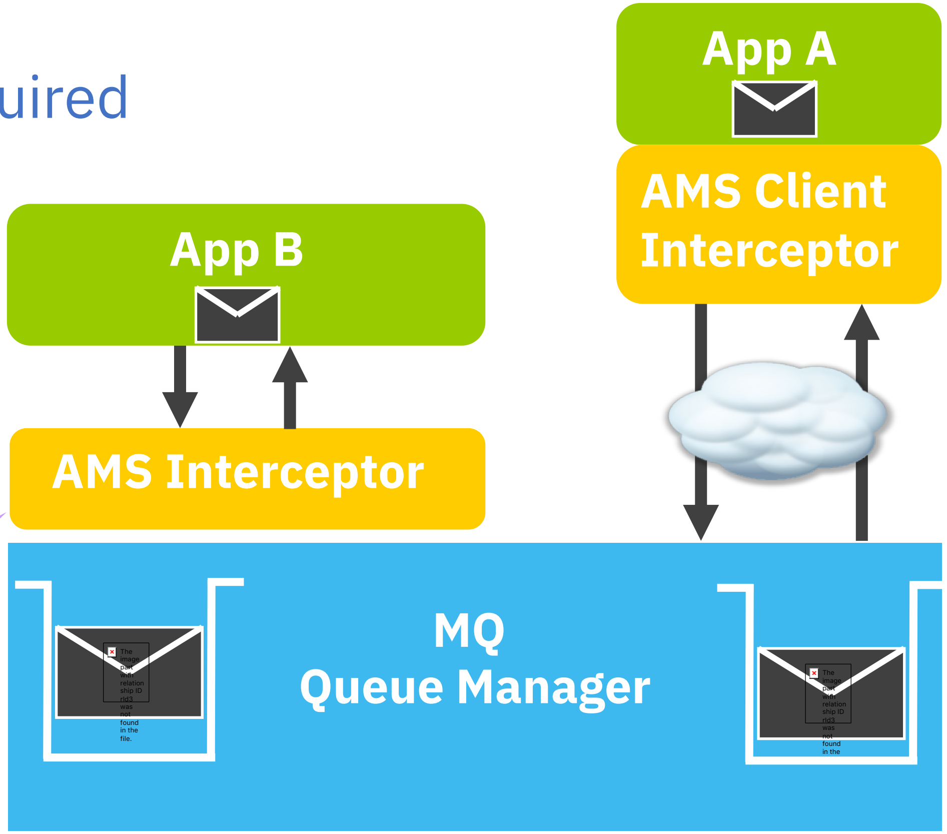
- Secures application data even before it is passed to MQ
- Upgrade from base MQ
  - No changes to existing applications or network required

MQ standard security:

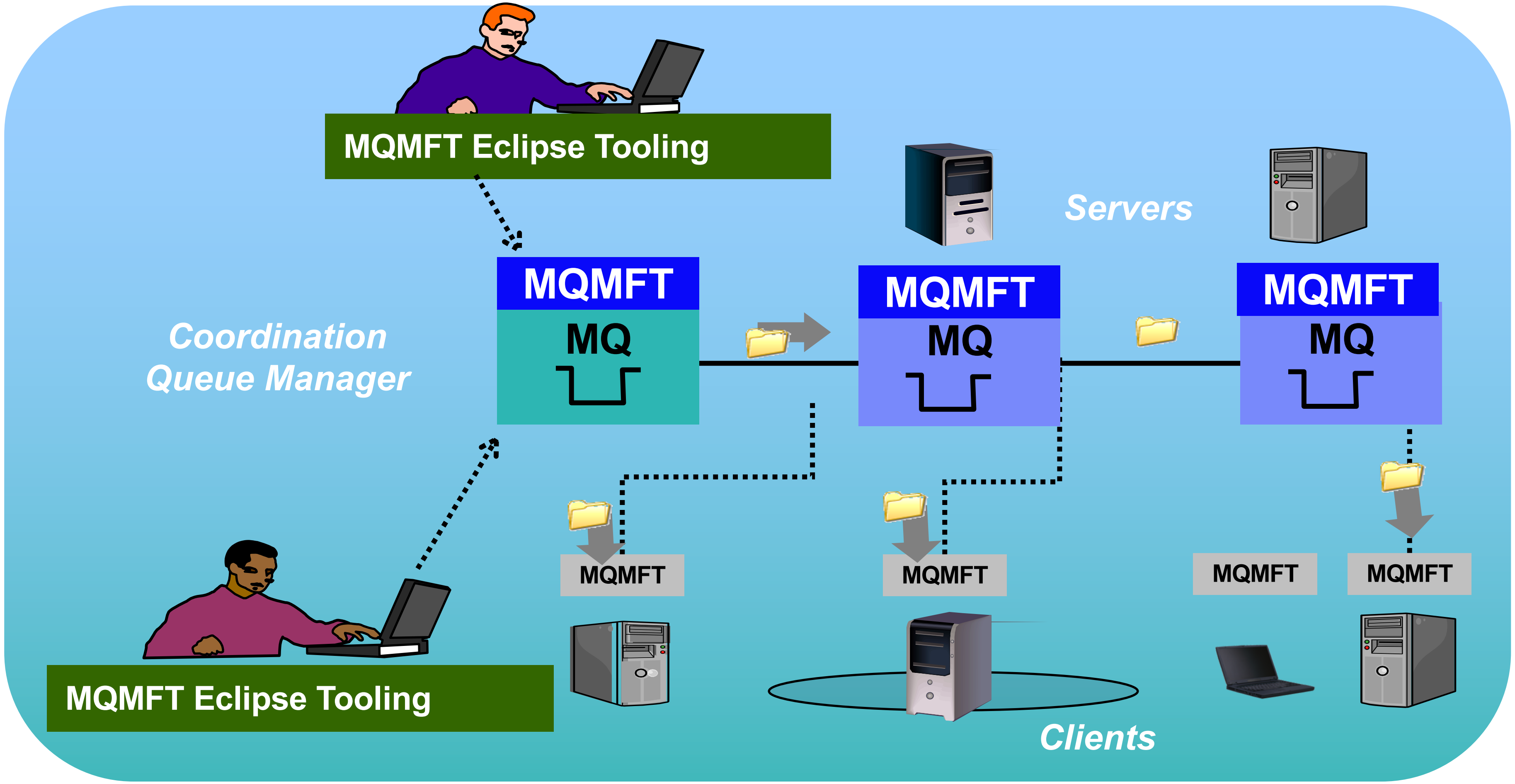
- Industry standard TLS channels (256-bit)
- Certified for Common Criteria
- Authentication is based on Operating System identifier of local process
- Message data can be encrypted in transport but not when it resides in the queues

MQ Advanced Message Security adds:

- Authentication policies are based on certificates associated with each application
- Message data is protected end-to-end – including when it resides in queues
- Much finer granularity in security policies
- No changes needed to applications or queues



MQ Managed File Transfer

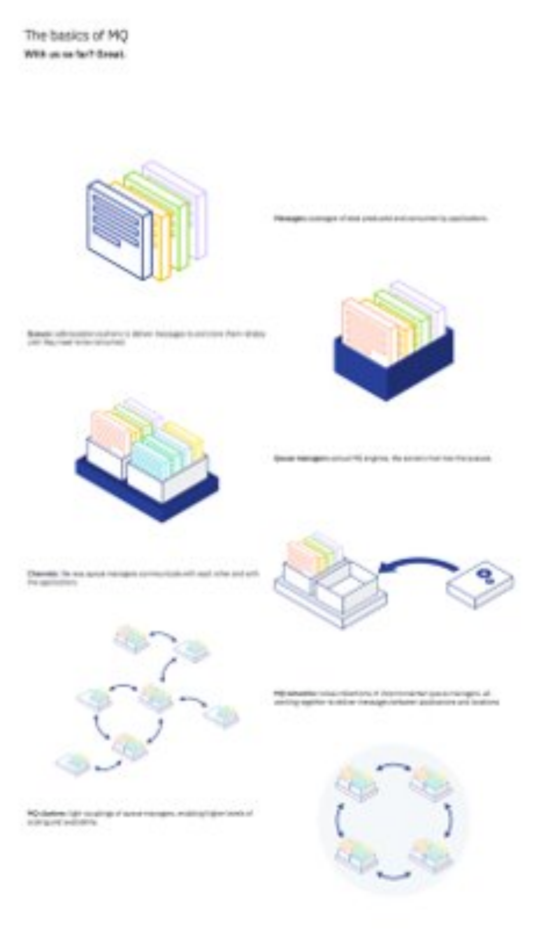


LearnMQ

- Finding it hard to get developers started with MQ?
- Point them to:

developer.ibm.com/messaging/learn-mq

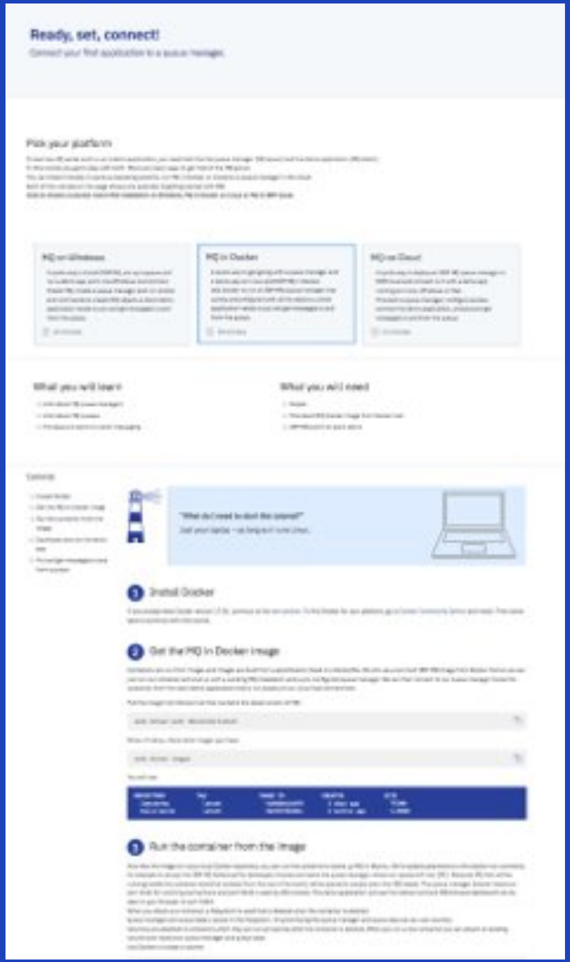
- Totally new to MQ? Learn the basics



The basics of MQ

With us so far? Great.

- Step-by-step guide to getting up and running with MQ



Ready, set, connect!

Pick your platform

MQ on Windows

MQ in Docker

MQ on Cloud

What you will learn

What you will need

"What do I need to start the tutorial?"

1 Install Docker

2 Get the MQ in Docker image

3 Run the container from the image

- Tutorials on building your applications

MQ tutorials, taking you further

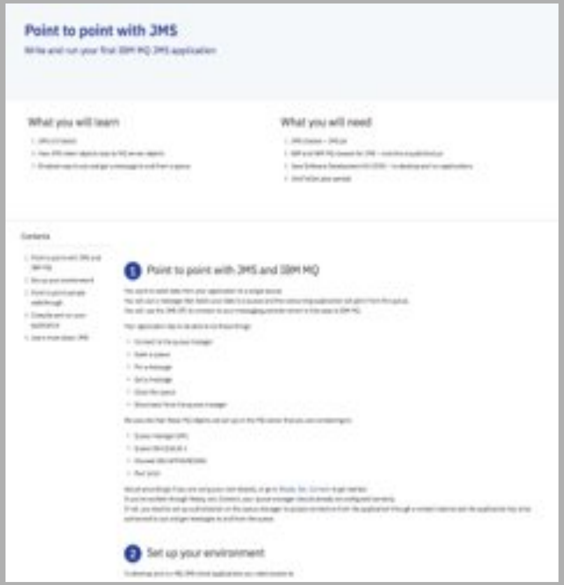
Search by:

Protected: Point to point with JMS

Protected: MQ Essentials

Protected: Ready, Set, Connect! (Windows)

Protected: Ready, set, connect! (Linux)



Point to point with JMS

Write and run your first IBM MQ JMS application

What you will learn

What you will need

1 Point to point with JMS and IBM MQ

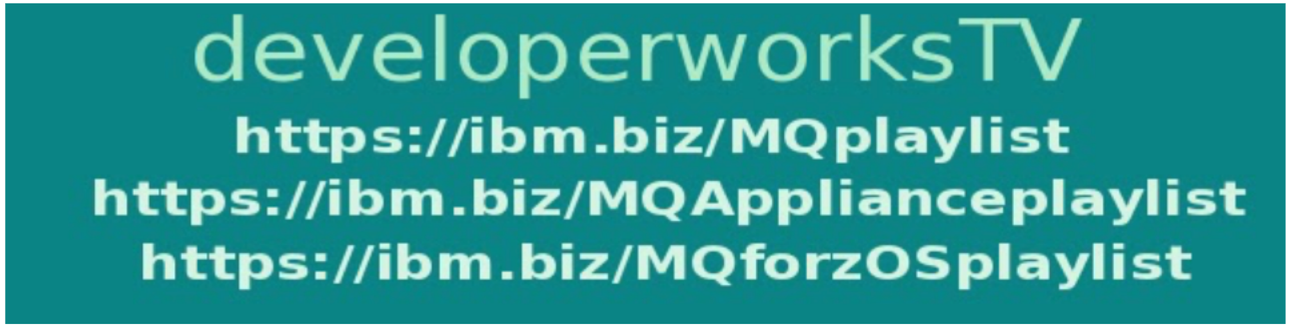
2 Set up your environment

Where do I get more information?

IBM MQ Knowledge Center
http://www.ibm.com/software/integration/wmq/library/

IBM Messaging developerWorks
developer.ibm.com/messaging

Youtube
https://www.youtube.com/user/IBMmessagingMedia



developerworksTV
https://ibm.biz/MQplaylist
https://ibm.biz/MQApplianceplaylist
https://ibm.biz/MQforzOSplaylist

IBM Messaging

Thanks for listening

Questions?

John Waldron IBM UK, IBM MQ L3 Service

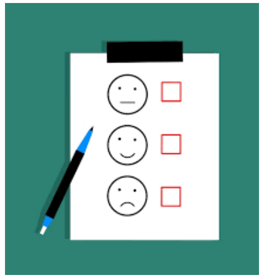
John.Diarmuid.Waldron@.ibm.com

We want your feedback!

- Please submit your feedback online at ….
  - ➢http://conferences.gse.org.uk/2018/feedback/AJ

- Paper feedback forms are also available from the Chair person

- This session is AJ



Copyright and Trademarks