

25

25

# Introduction to IBM MQ

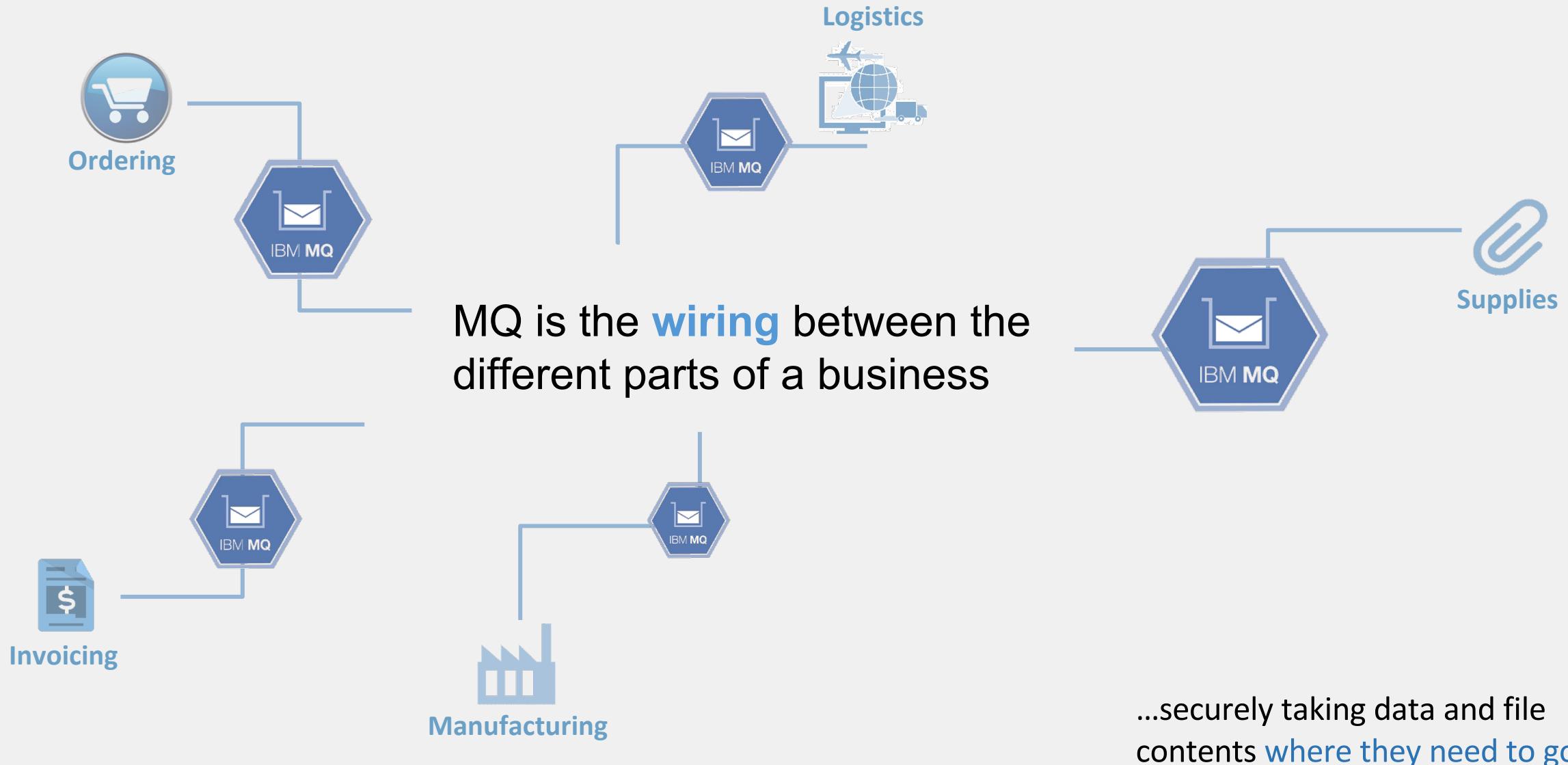
25

25

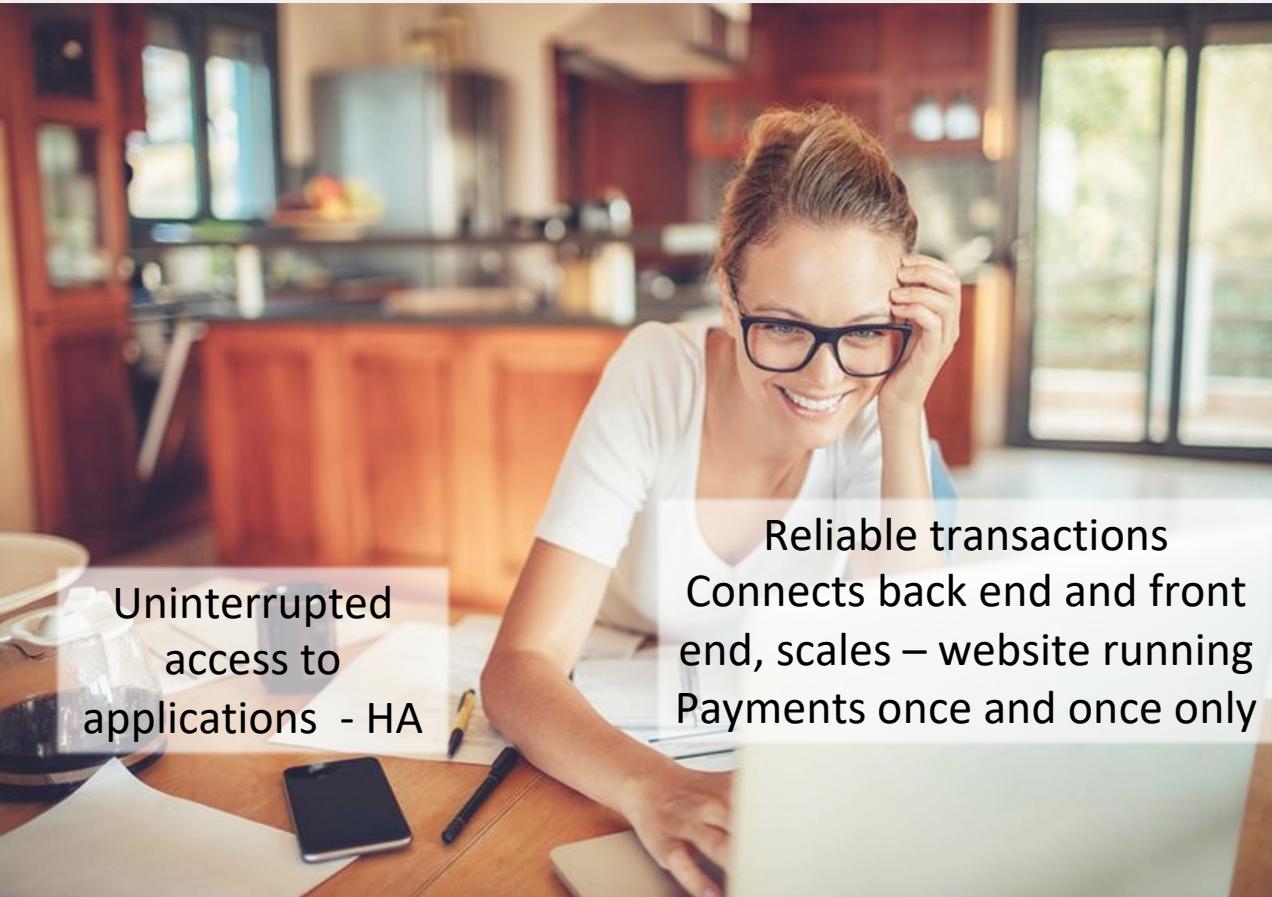
25

25

# But what *is* MQ?



# How does MQ help?



# And so are most successful companies

**85%** of the Fortune 100 are MQ customers

**Over 70%** of the Global 500 are MQ customers

**94%** of the top 100 global banks are MQ customers

**59%** of 100 top US retailers are MQ customers  
(90% of the top 20)

## In every industry, worldwide

- 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\*



**market leader in messaging**



# And this is why

“It's one of the few things we have that just runs, almost to a point where you forget to go back and do some upgrades.”

“We never have an MQ issue”

“It gives us flexibility when it comes to offering different projects or different types of solutions to customers”

“it fits into any commodity that you have”

“It's up 24/7”

“It runs millions of transactions in our environment on a daily basis”

*“It's solid, and it scales”*

“We've been able to get some disparate applications that weren't originally written to be integrated, but we've been able to make that happen”

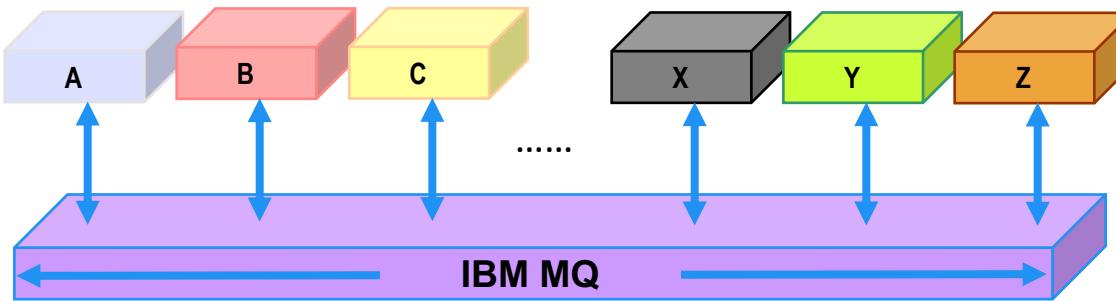
“the biggest thing is that we'll never lose a message”

“Stability is unrivaled”

“The integration it provides makes a lot of stuff easier”

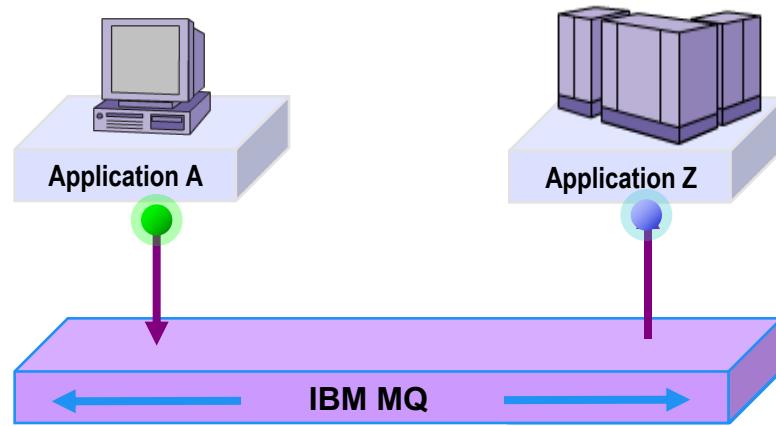
# What Fundamental Problem does IBM MQ solve?

*How to move information around...*



Provides messaging services to applications and Web services that need to exchange data and events with:

- ✓ *Proven reliability*
- ✓ *Transactional integrity*
- ✓ *Consistency*
- ✓ *Time independence*
- ✓ *Ease and Speed*
- ✓ *Flexibility*
- ✓ *High-performance*
- ✓ *Security*
- ✓ *Scalability*
- ✓ *Auditability*



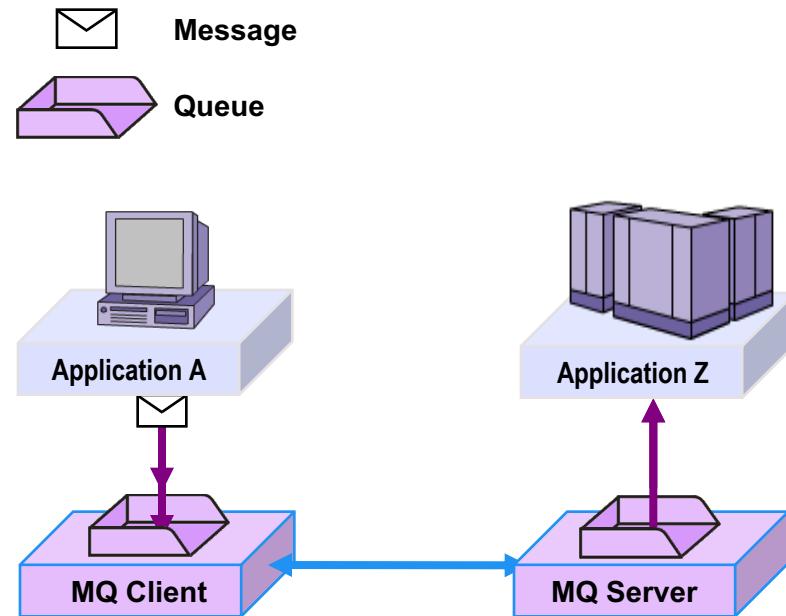
***MQ is like email for application integration  
...but email you can bet the business on***

Messaging services are based on **Queues** that store and forward data based on simple programming commands

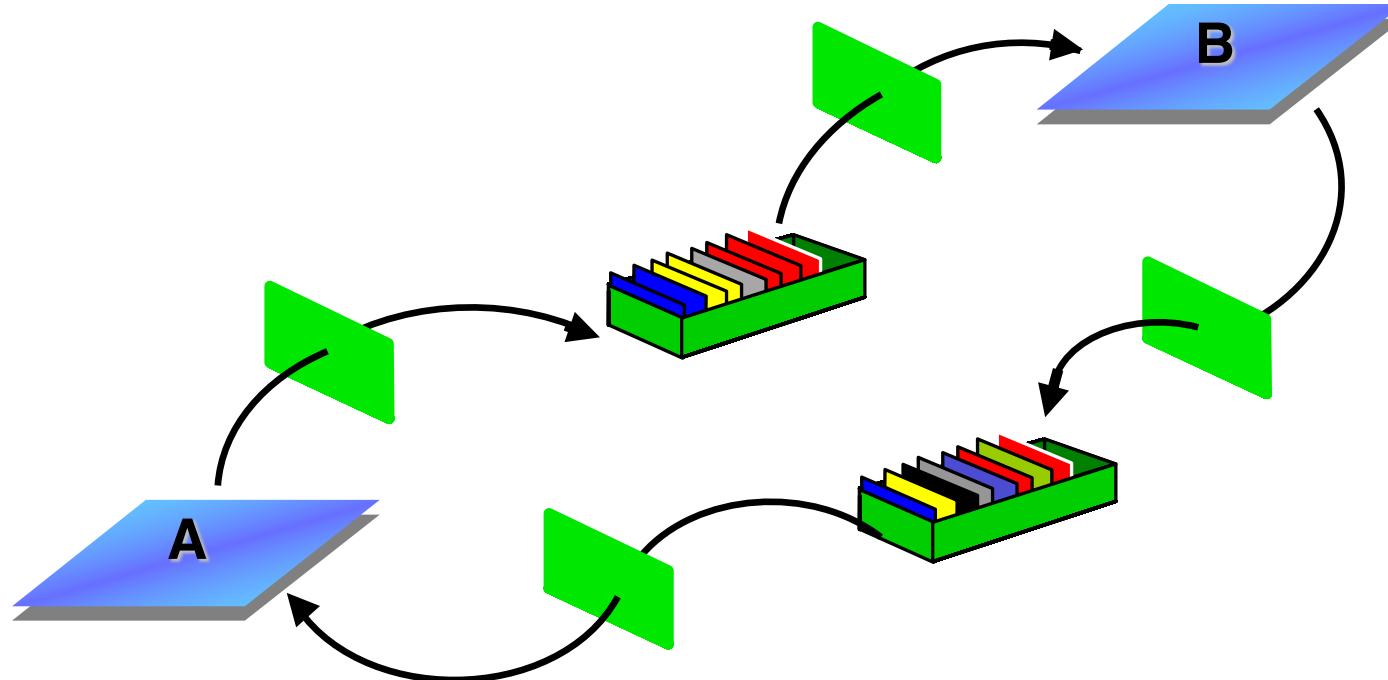
Uses the proven database technique of two-phase commit **transactions** to ensure messages are not lost or duplicated

Uses **publish/subscribe** to route messages dynamically based on keywords or “topics”

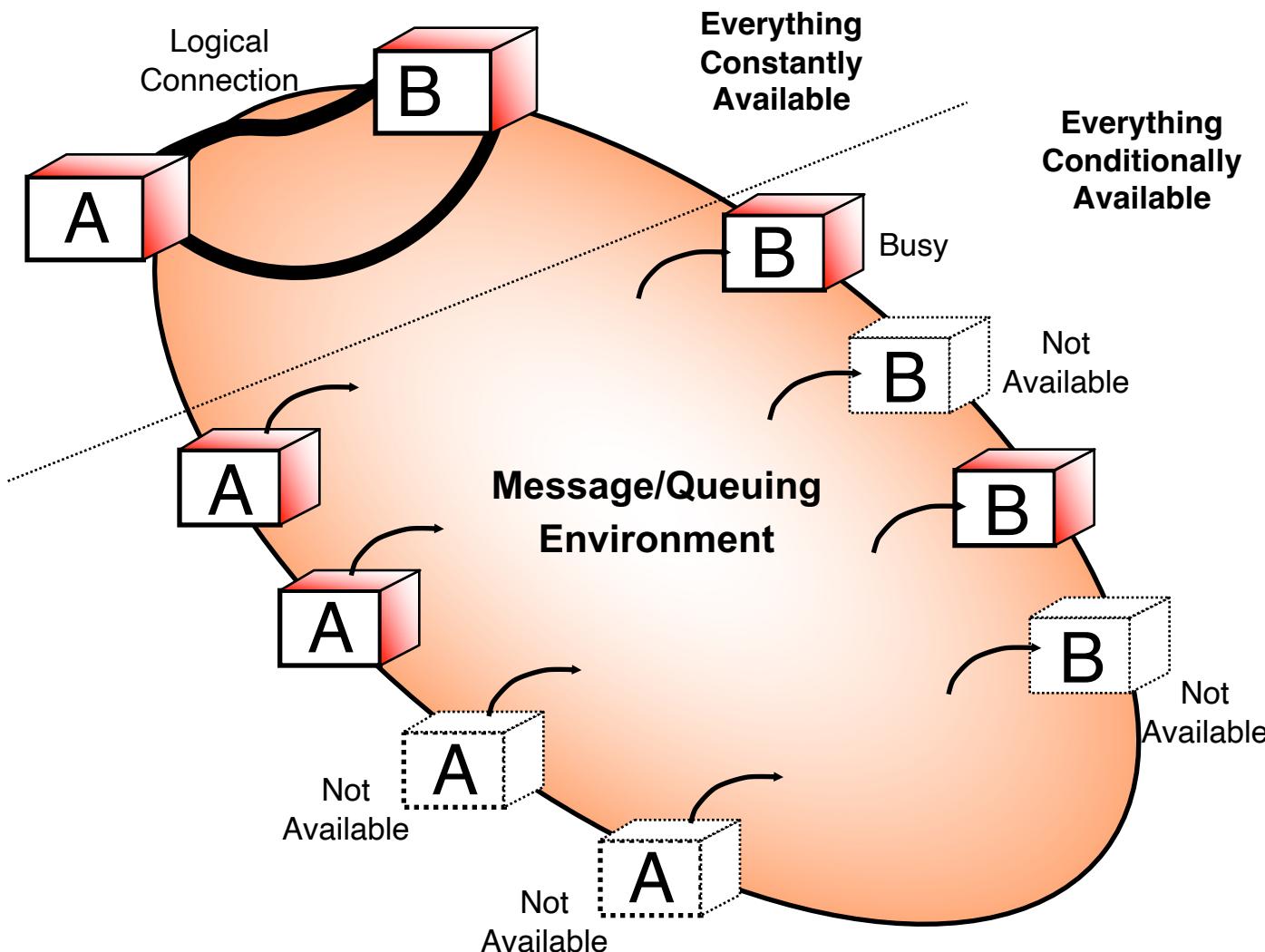
Uses multi-processor threading and **clustering** to accelerate throughput of messages



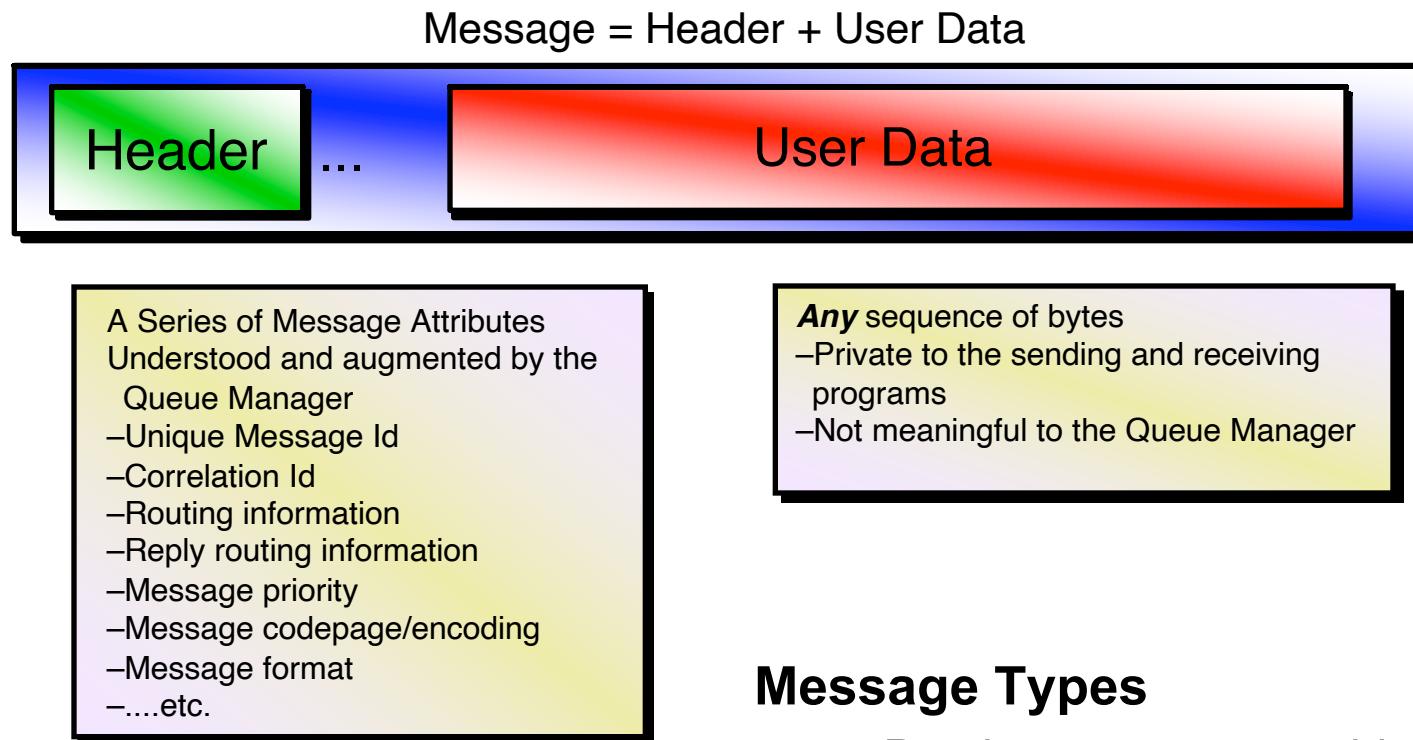
# IBM MQ – A Technical View



# Availability Choices: Synchronous vs Asynchronous



# What is a Message?



## Message Types

- Persistent ... recoverable
- Non Persistent

# The Message

A message is considered to be the unit of data to be moved from one application to another

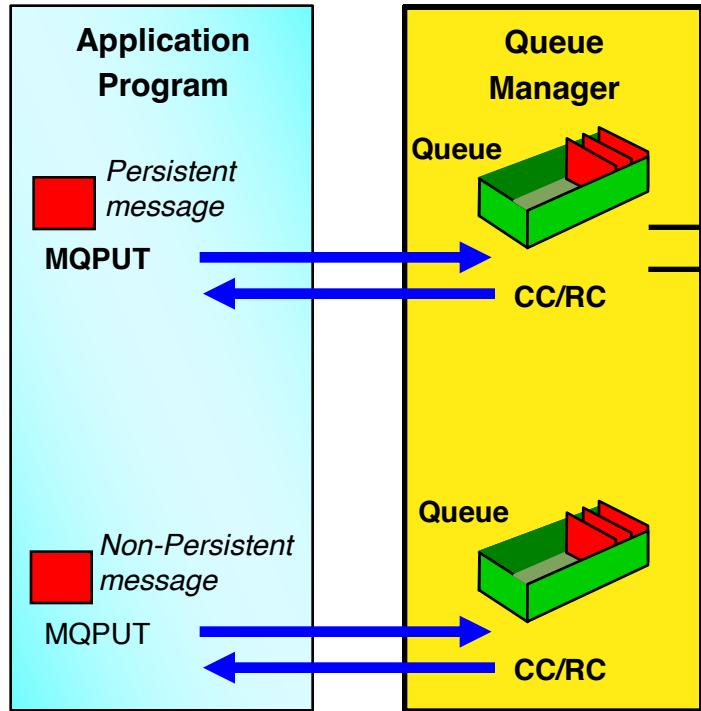
A message is built by an application

A message is consumed by a different application

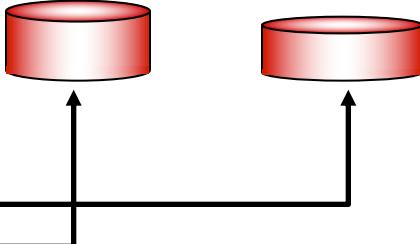
Message can contain any kind of data:

- ✓ Binary data
- ✓ Text data
- ✓ Structured data (C Structures, COBOL CopyBook, Serialized Java objects)
- ✓ Anything!!

# Message Persistence



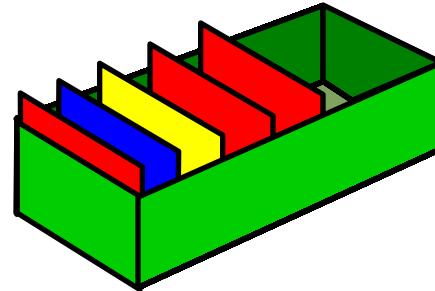
## Queue Files Logs



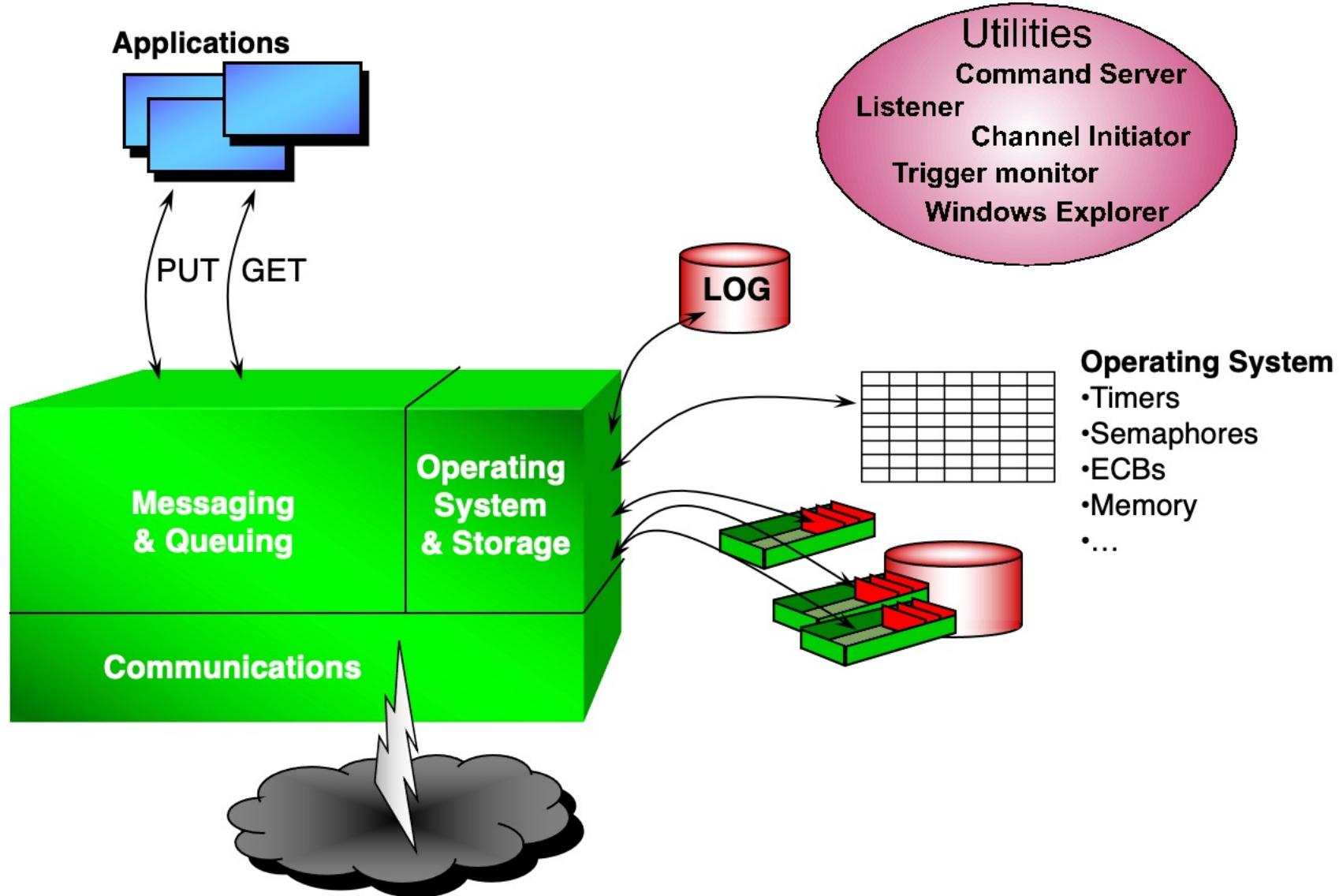
- Logging has implication on performance
- Persistent Messages are always recoverable
- Non-persistent Messages have 2 classes of service:
  - Messages are retained for the life of the Qmgr
  - Messages can survive a normal shutdown and restart of Qmgr

# What is a Queue ?

- **Messages** delivered asynchronously to a **Queue**
- **Place to hold messages**
- **Queue creation**
  - Pre-defined
  - Dynamic definition
- **Message Access**
  - FIFO
  - Priority
  - Direct
  - Destructive & non-destructive access
- **Parallel access by applications**
  - Managed by the queue manager



# What's a Queue Manager?



# Channels

Used for Queue Manager to Queue Manager communication

- Usually in pairs

- Uni-directional

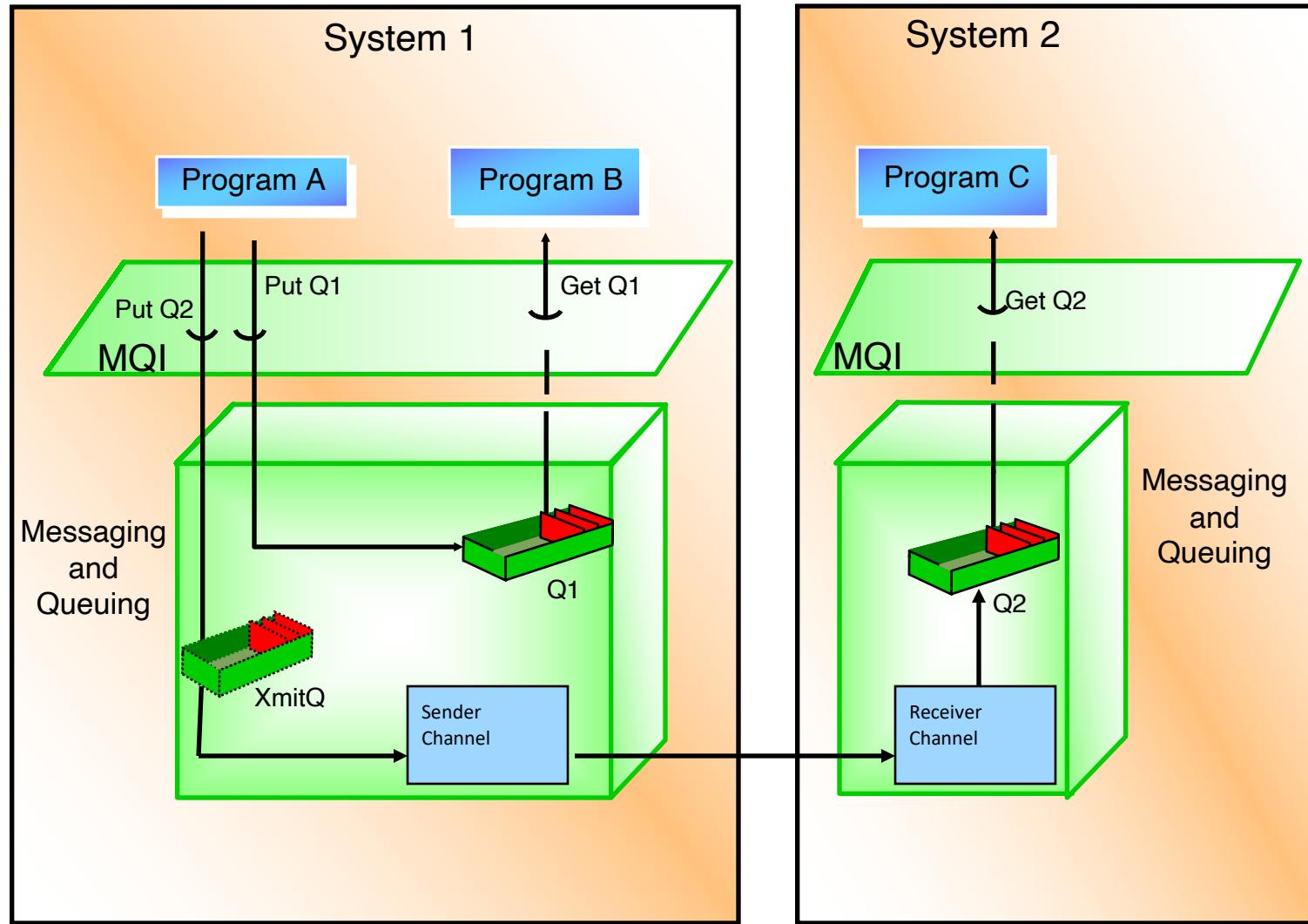
Used for MQ Client to Queue Manager

- Single, bi-directional

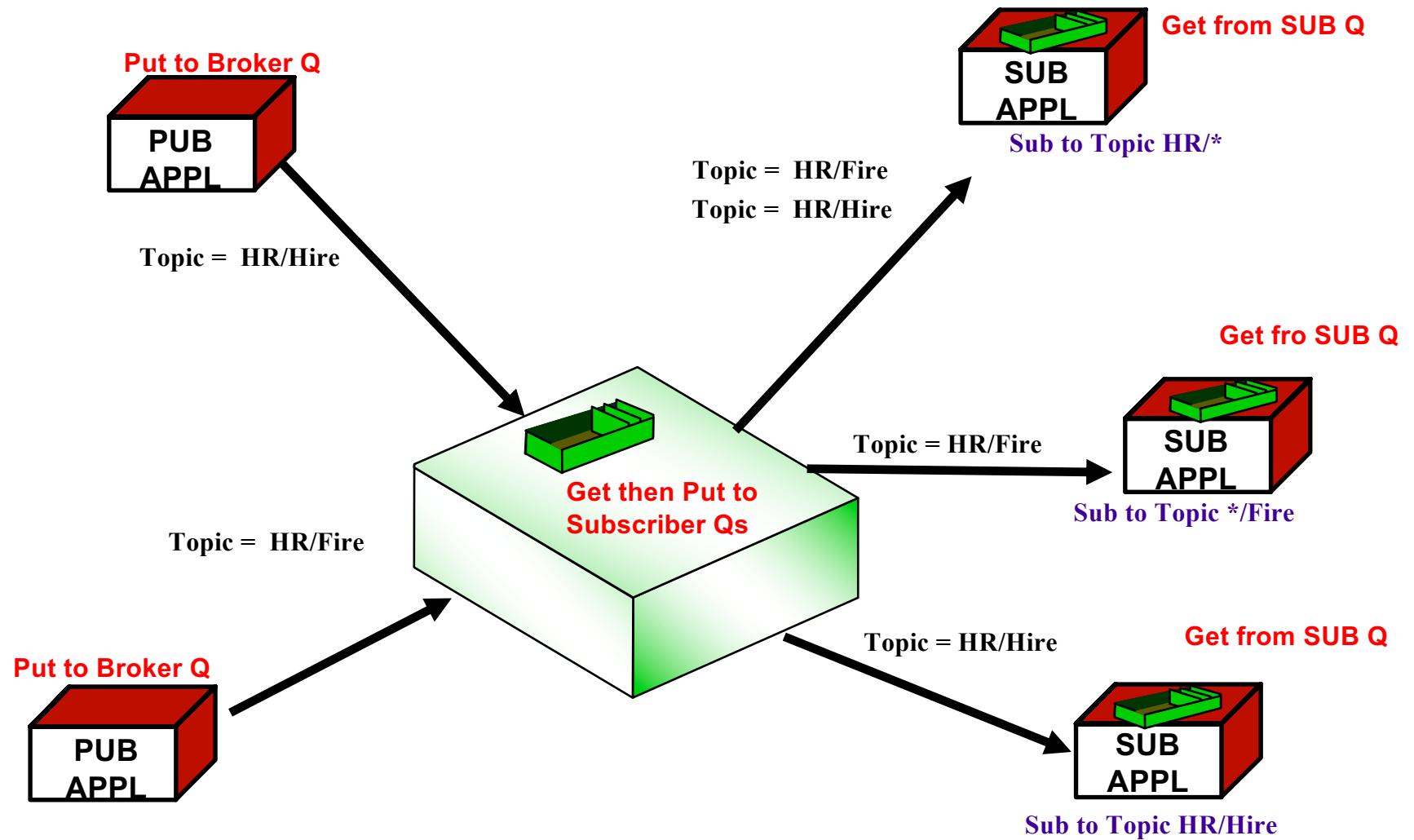
Building block for ESB



# Applications can use Point to Point



# Publish / Subscribe



# Applications can be Transactional

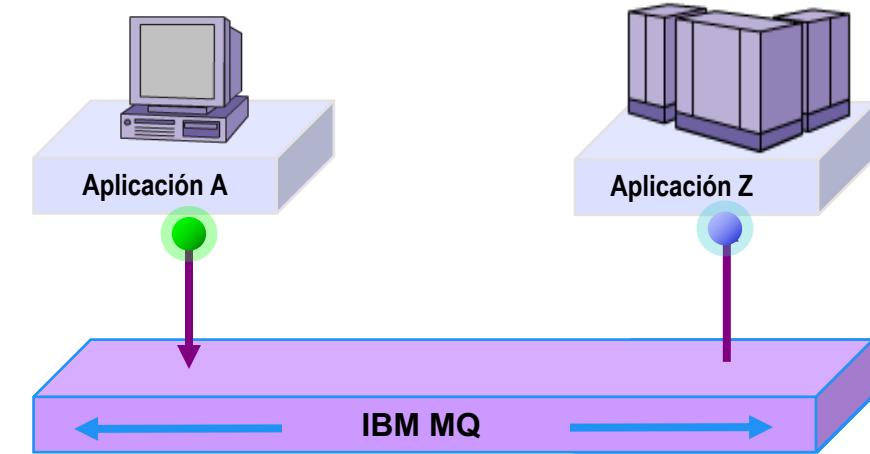
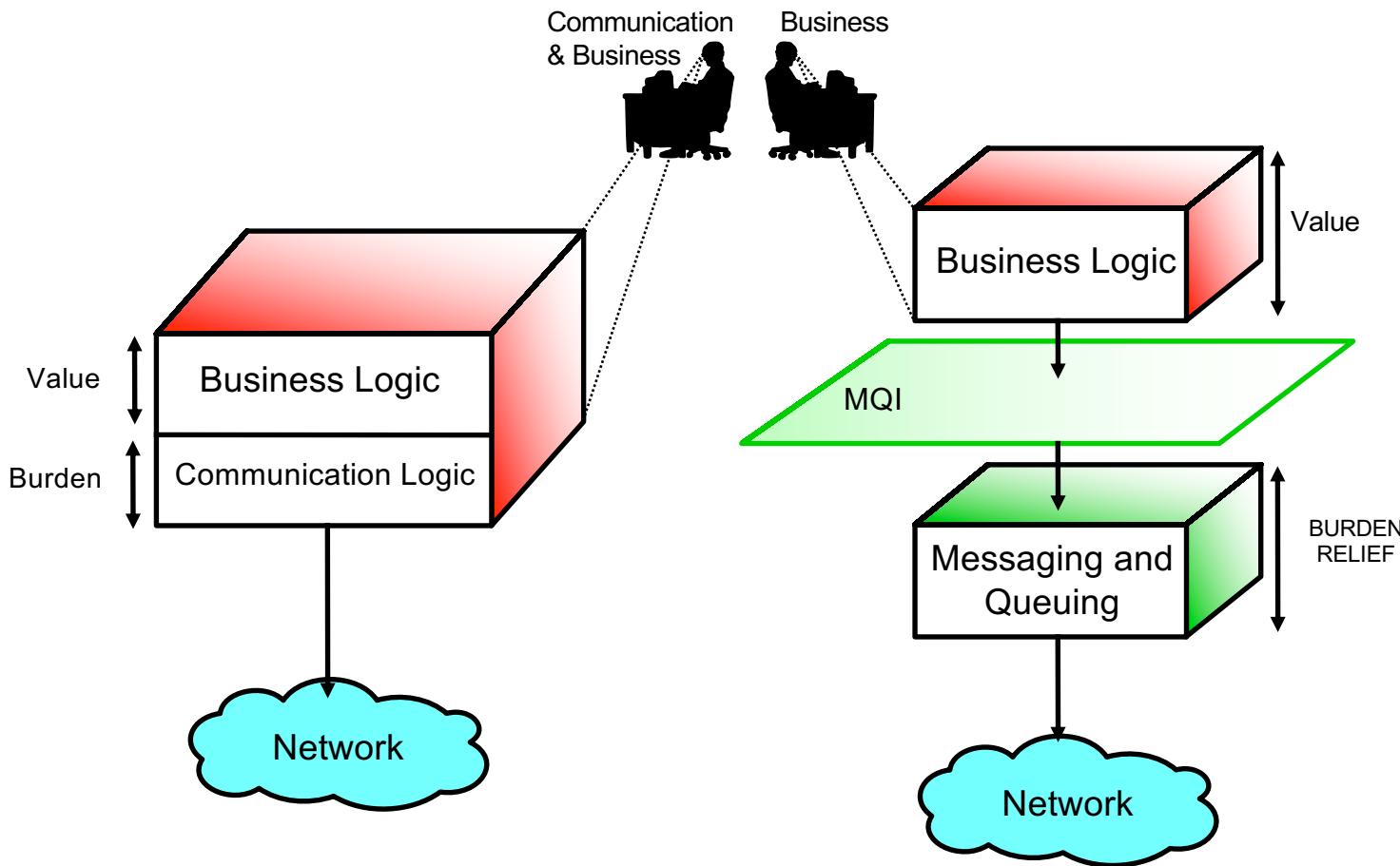
Messaging can be performed under transaction control

- ✓ Messages can be put or got under a logical unit of work
- ✓ Messages can be committed or rolled back as an atomic unit

Distributed transactions are supported

- ✓ IBM MQ can be XA resource manager
- ✓ IBM MQ can be XA transaction manager

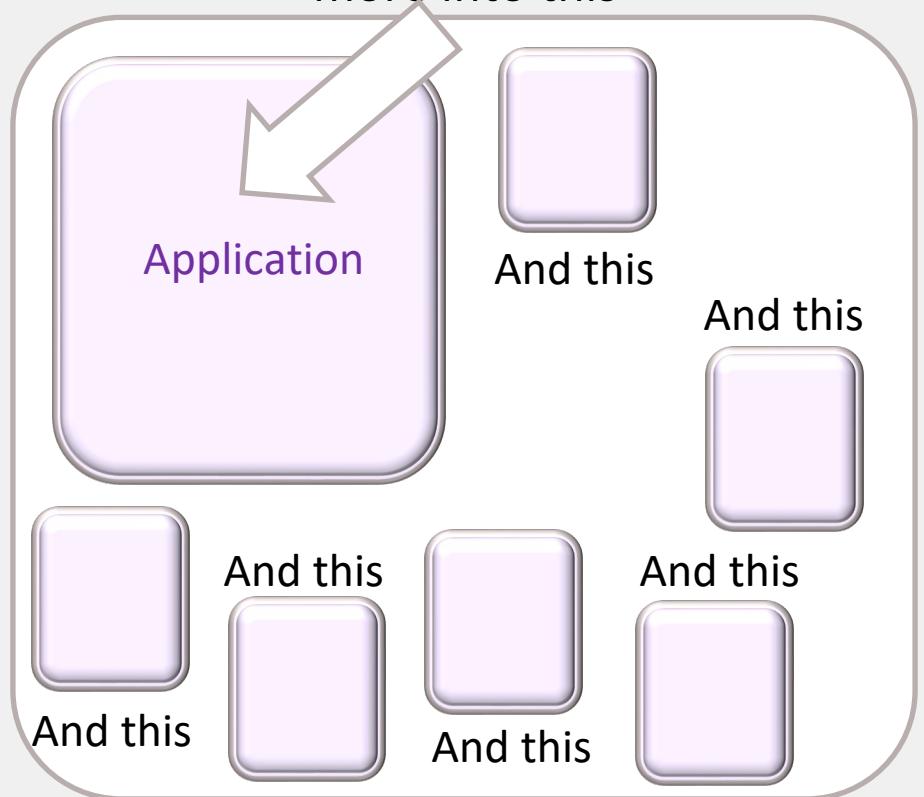
# Shielding Developers from Networking Complexities



# Why does MQ matter?

## Option A:

Build protection, transactionality, manual wiring between application types, and more into this



...and every future application

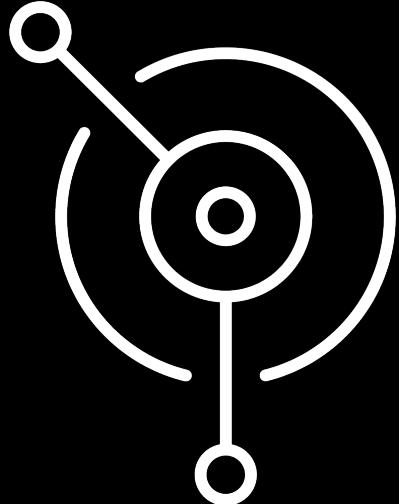
## Option B:

Get **ALL OF THAT** and more with **MQ**

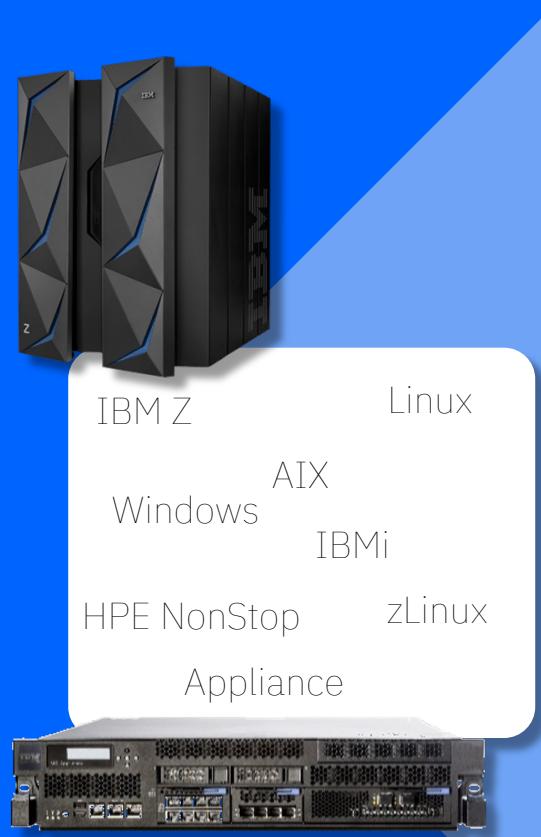


...keep adding applications without worrying about these changes

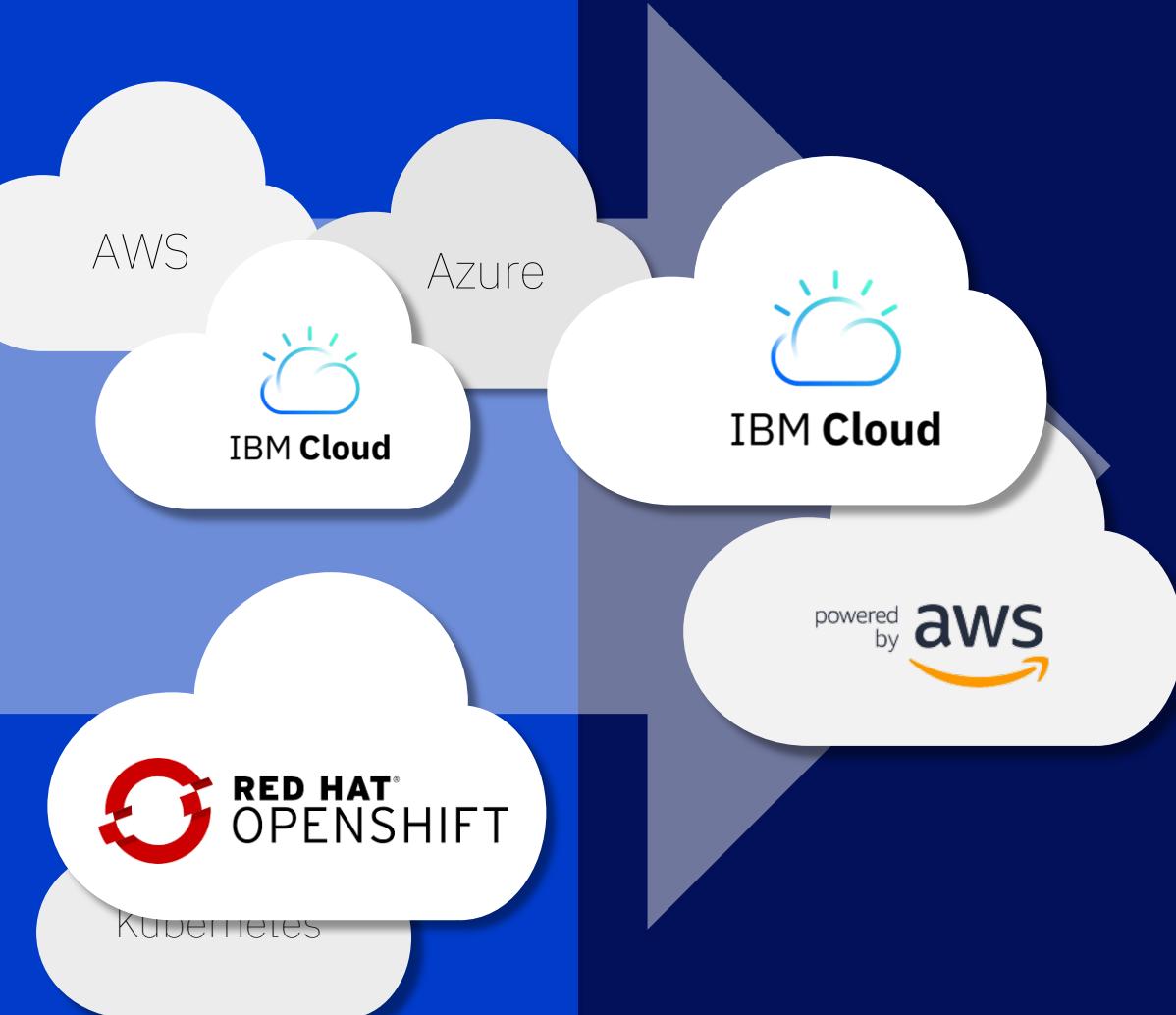
A focus on where  
you need MQ today



On-premise, software  
and the MQ Appliance,  
exactly as you need it

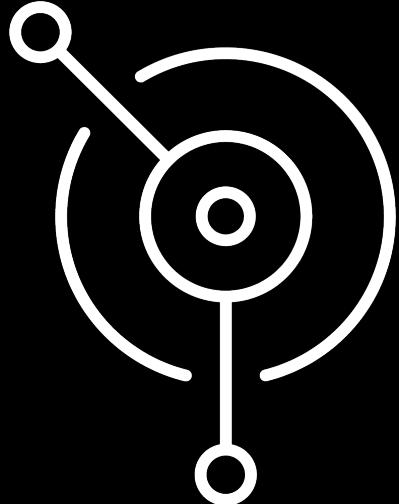


Run MQ yourself in  
public or private clouds,  
virtual machines or  
containers

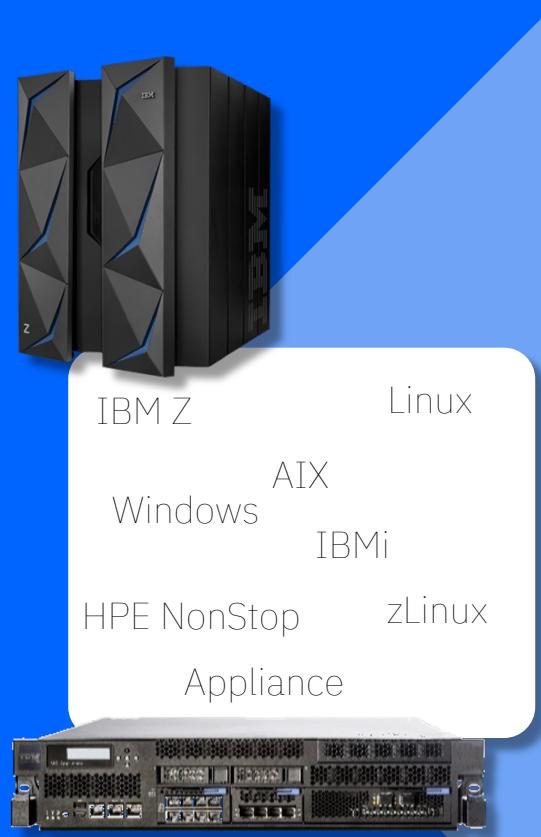


Let IBM host MQ for you  
with its managed SaaS  
MQ service in public  
clouds,  
IBM Cloud and AWS

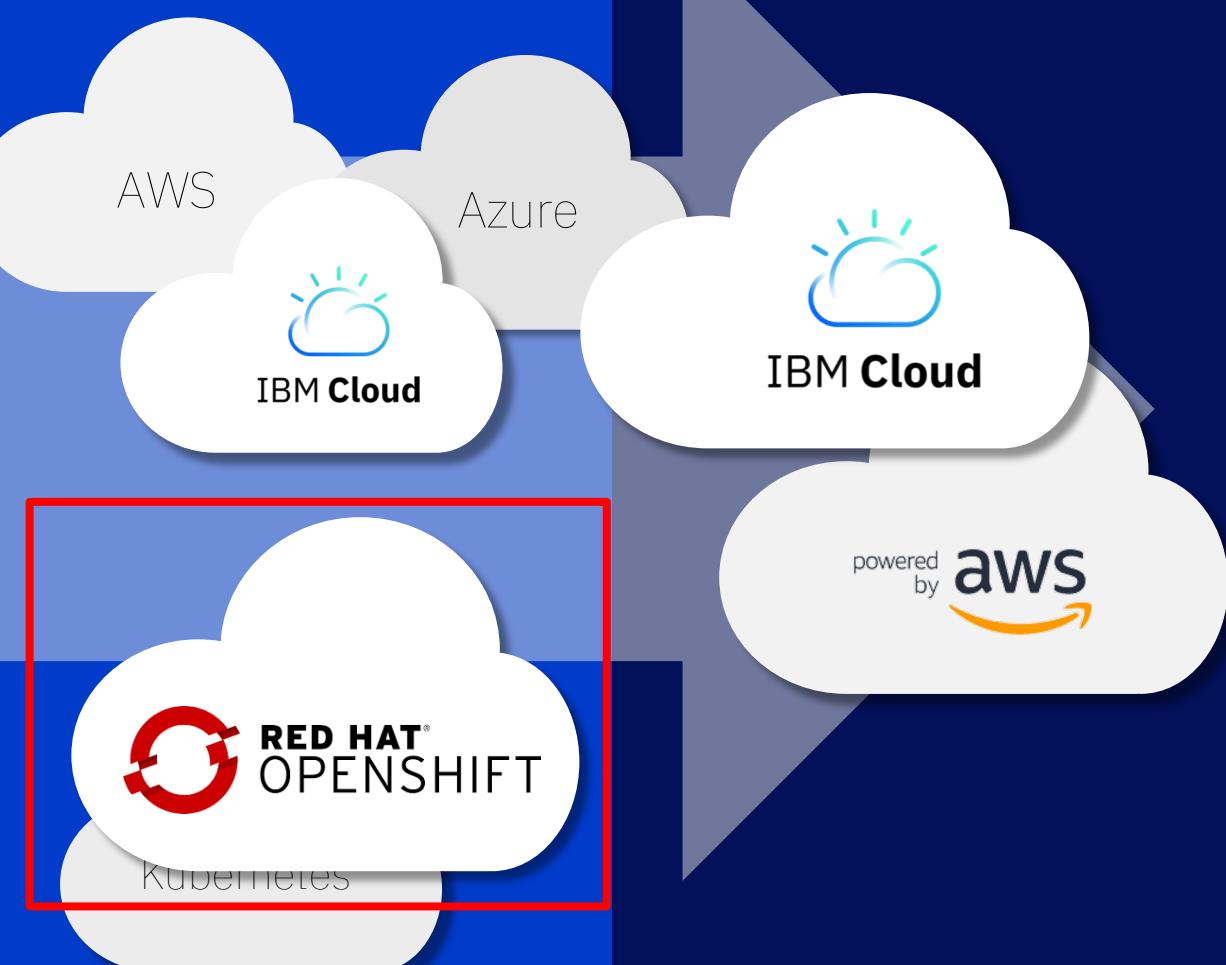
A focus on where  
you need MQ today



On-premise, software  
and the MQ Appliance,  
exactly as you need it



Run MQ yourself in  
public or private clouds,  
virtual machines or  
containers



Let IBM host MQ for you  
with its managed SaaS  
MQ service in public  
clouds,  
IBM Cloud and AWS

# IBM MQ in Containers

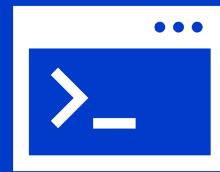
## Small



MQ is designed to be lightweight and to scale to run at any size

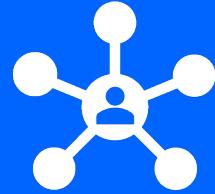
IBM itself maintains  $\frac{1}{2}$  core queue managers in production

## Application centric



Lightweight container based queue managers encourage an application centric microservice approach over monolithic, central, messaging systems

## Manage at Scale



Templated deployments coupled with centralised administration and monitoring

Building large dynamic networks of messaging servers is fundamental to MQ

## Highly available



Deploy highly available queue managers with ease

Active/active, horizontally scaled deployment patterns are a natural fit for containers

# MQ within the Cloud Pak for Integration

## Strategic focus

IBM is committed to building ever increasing value into its IBM and Red Hat OpenShift platform



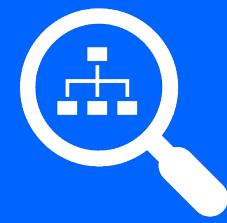
## Certified Container

Production ready container images with a Kubernetes Operator that simplifies the operational activities



## Deep Insight

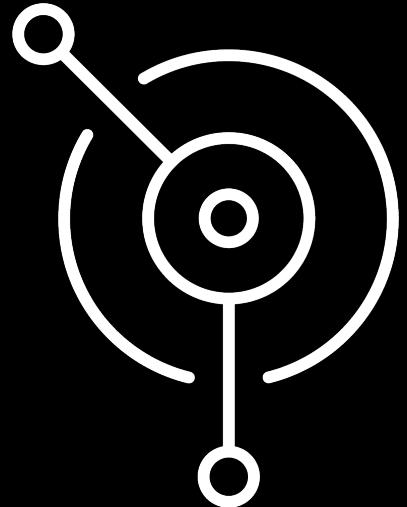
Built-in emission of logging and tracing data, empowering developers and administrator to observe and troubleshoot



## Flexible Adoption

A flexible deployment model allowing traditional software and container technology to be adopted at your own speed

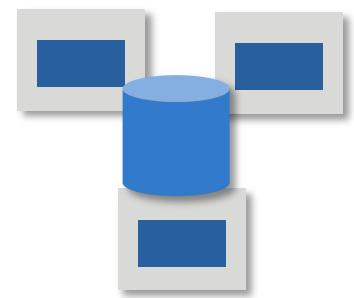




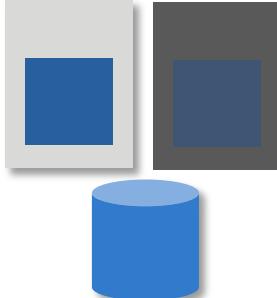
# MQ message availability

Protecting your critical data

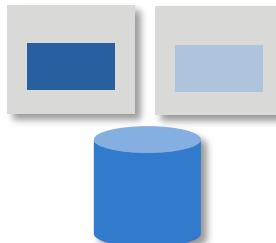
# Constantly evolving to meet your availability needs



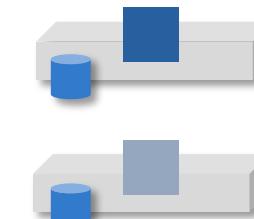
z/OS Queue  
Sharing Groups



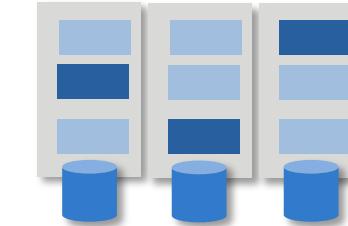
System  
managed HA



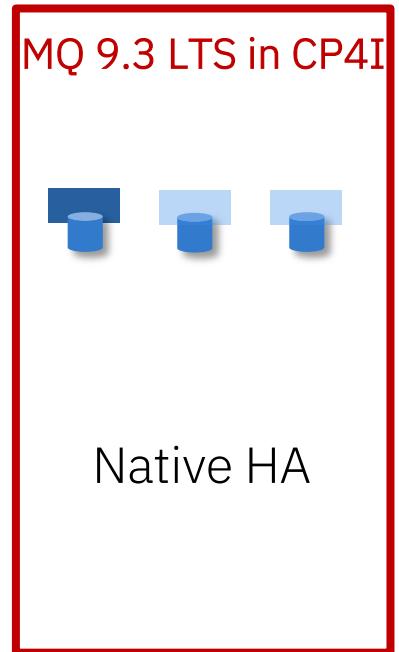
Multi-instance  
queue  
managers



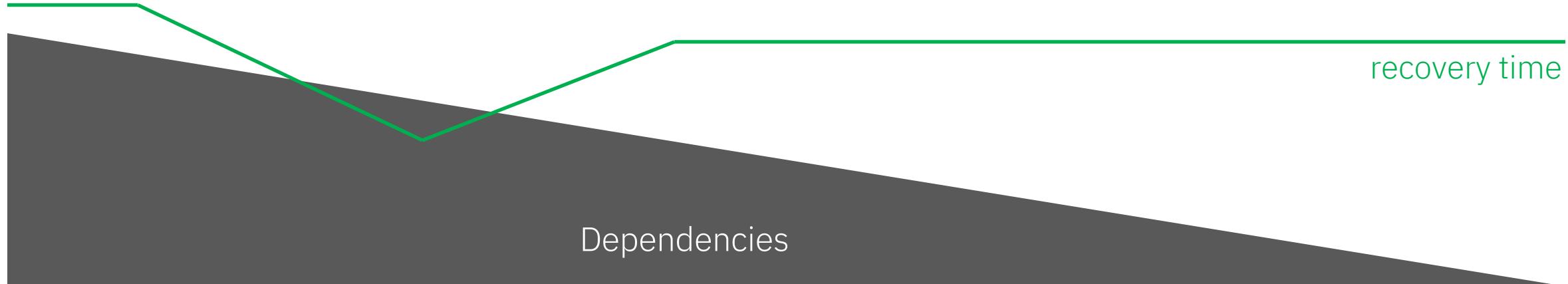
MQ Appliance



Replicated data  
queue manager



Native HA



# Replicated Data Queue Managers

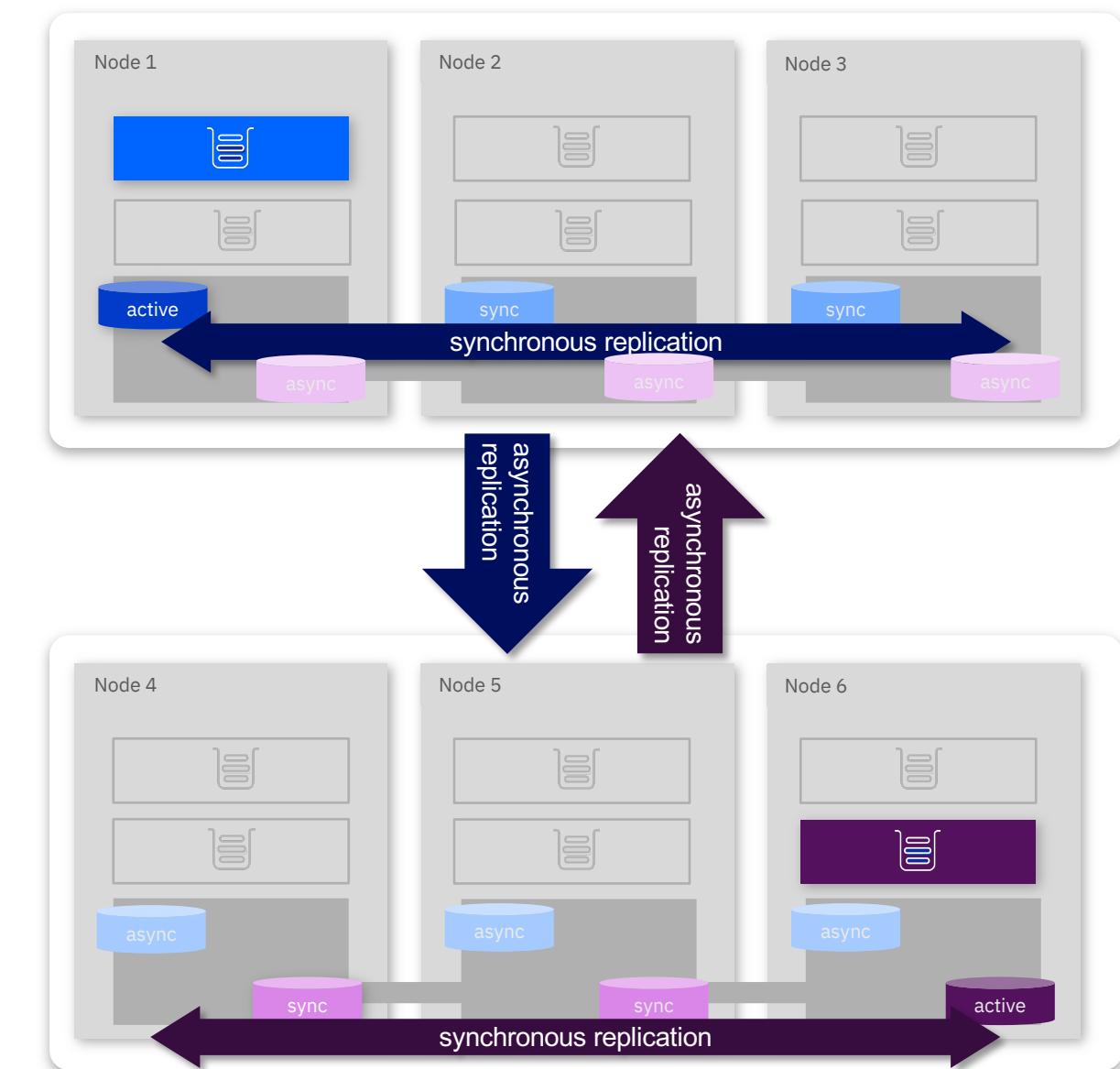
RHEL x86, MQ Advanced HA solution with no need for a shared file system or HA cluster

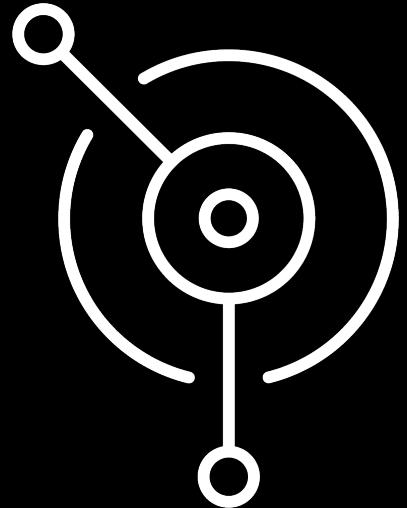
All components are provided and supported through MQ Advanced

Synchronous replication and fast quorum based take over for HA scenarios

Asynchronous replication between quorum groups to support long distance DR deployments

All nodes support concurrently running multiple different active queue managers, with bidirectional asynchronous replication, supporting active/active HA and DR topologies

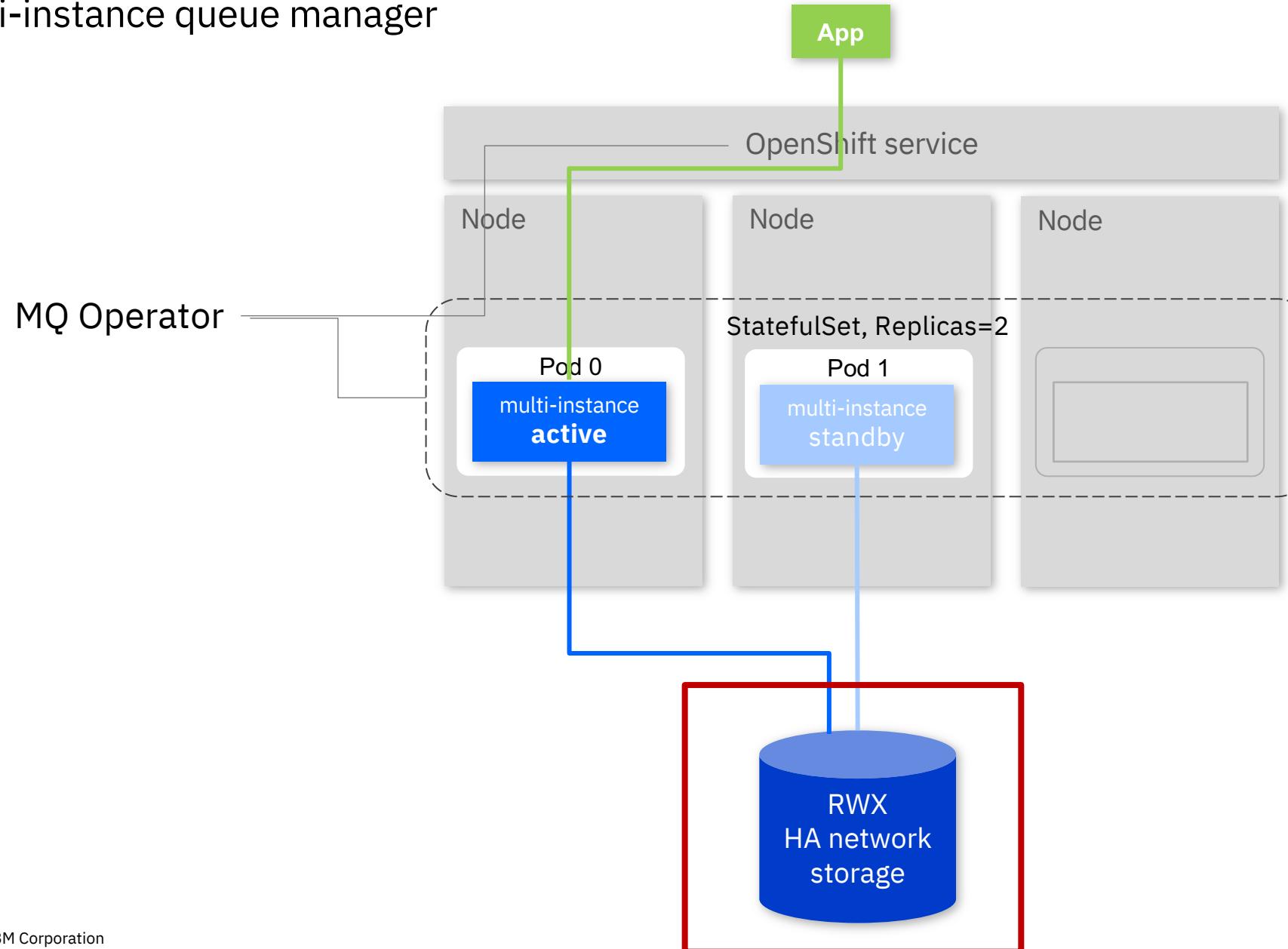




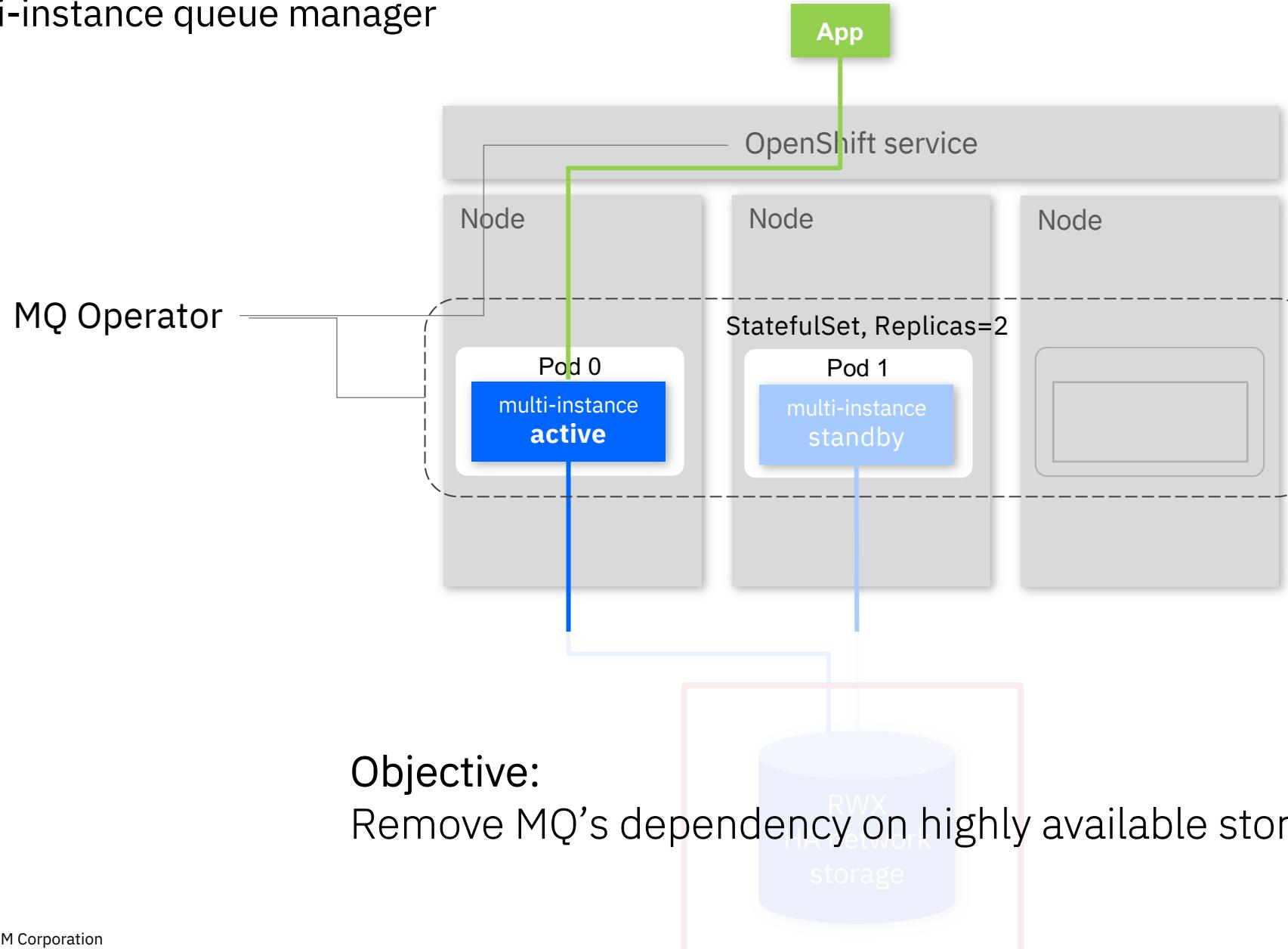
# Cloud native availability

MQ Native HA

# Best HA for OpenShift before Multi-instance queue manager

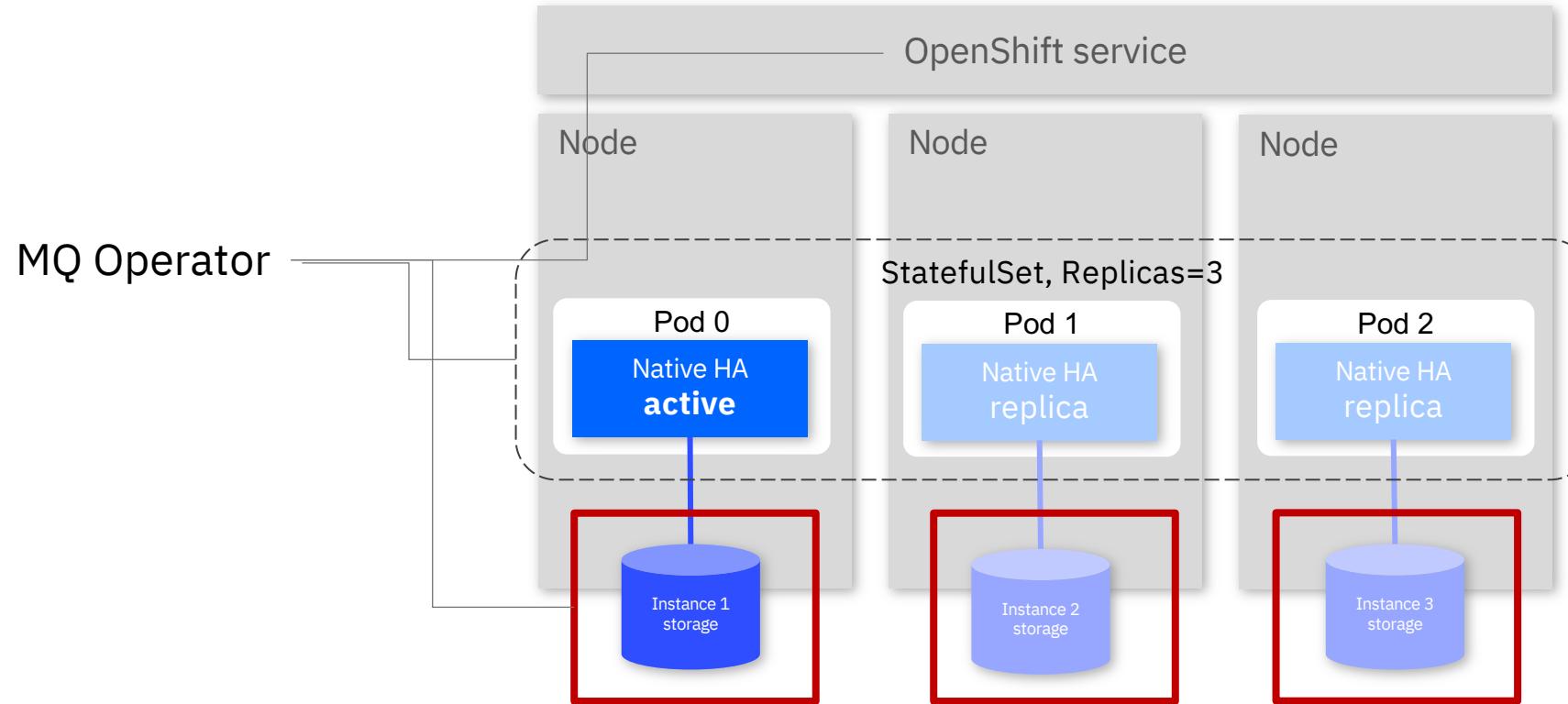


# Best HA for OpenShift before Multi-instance queue manager



# Best HA for OpenShift today

## Native HA queue manager



**Availability:**

cross AZ, fail overs take “seconds”

**Compatibility:**

simple RWO block storage requirement

**Cost:**

Included in CP4I license (MQ Advanced ratio)

**Complexity:**

No external services to manage

**Performance:**

Network + block storage

# MQ Native HA

Solution: Convert MQ's persistence layer to be cloud native

**Problems to solve:** MQ persistent data replicated across AZs  
Consistency across replicas guaranteed  
Fast and reliable failure detection and fail over

## Raft

A proven, yet *understandable*, consensus algorithm

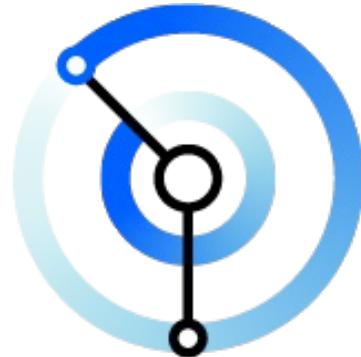
Based on the concept of a **sequential log of state changes**



## IBM MQ

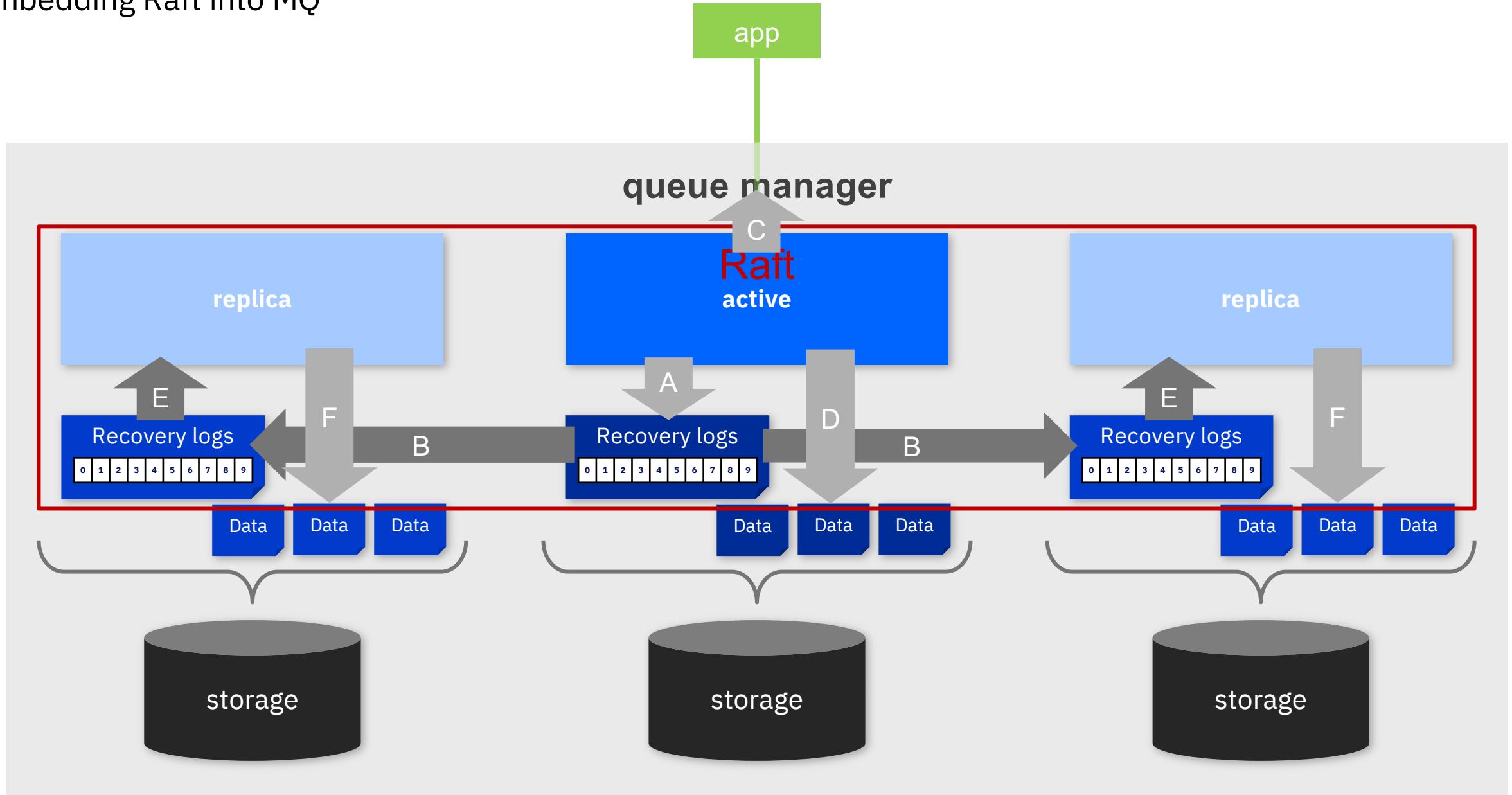
A proven, high performing and reliable, messaging solution

Built from day one around a **sequential log of state changes**



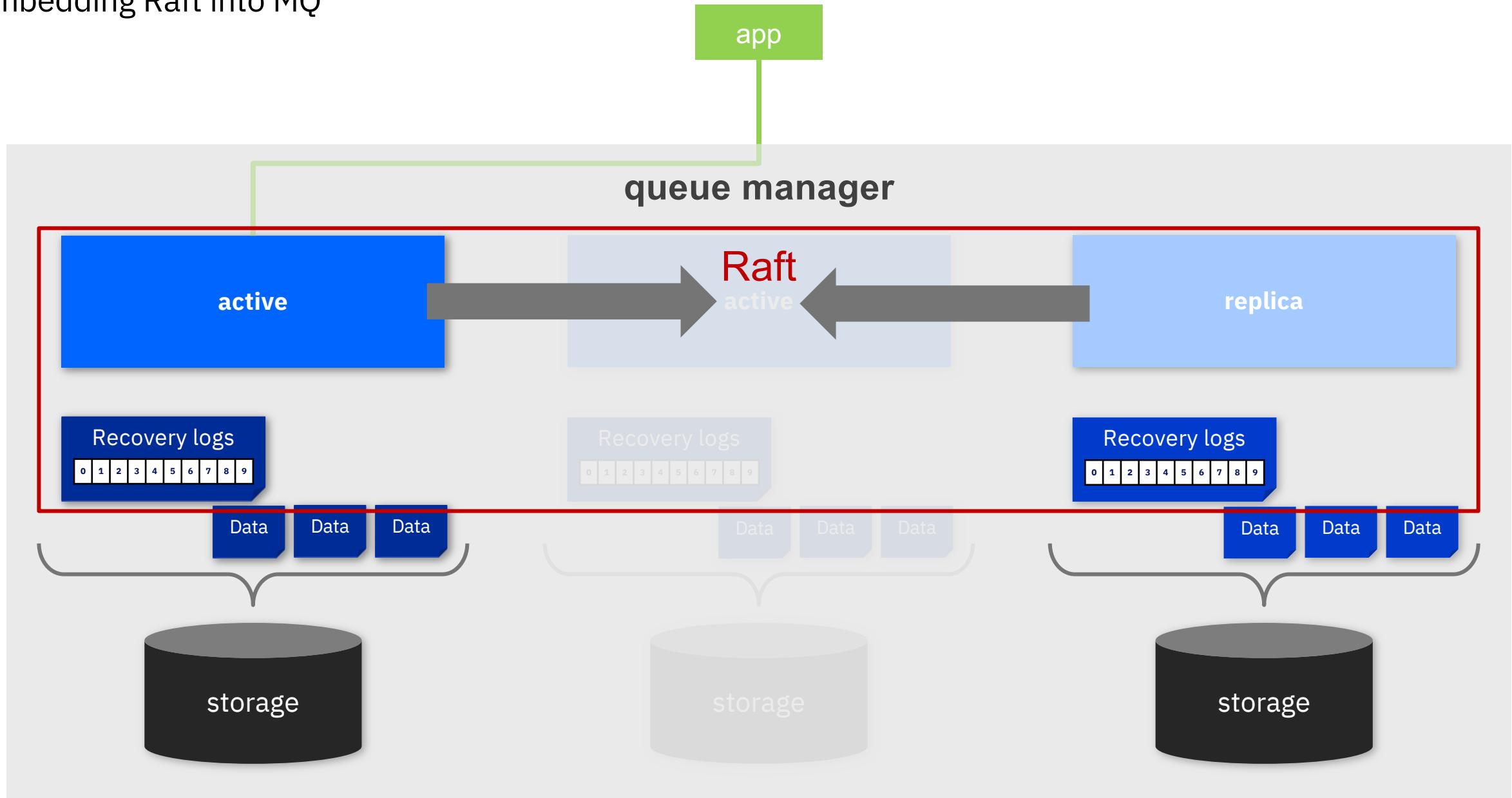
# MQ Native HA queue manager

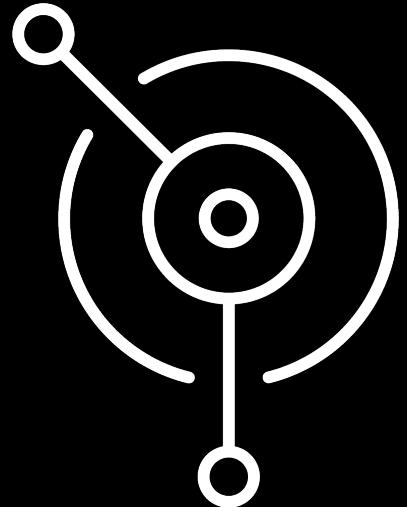
Embedding Raft into MQ



# MQ Native HA queue manager

Embedding Raft into MQ





# Active/active messaging

Building scalable, fault tolerant, solutions

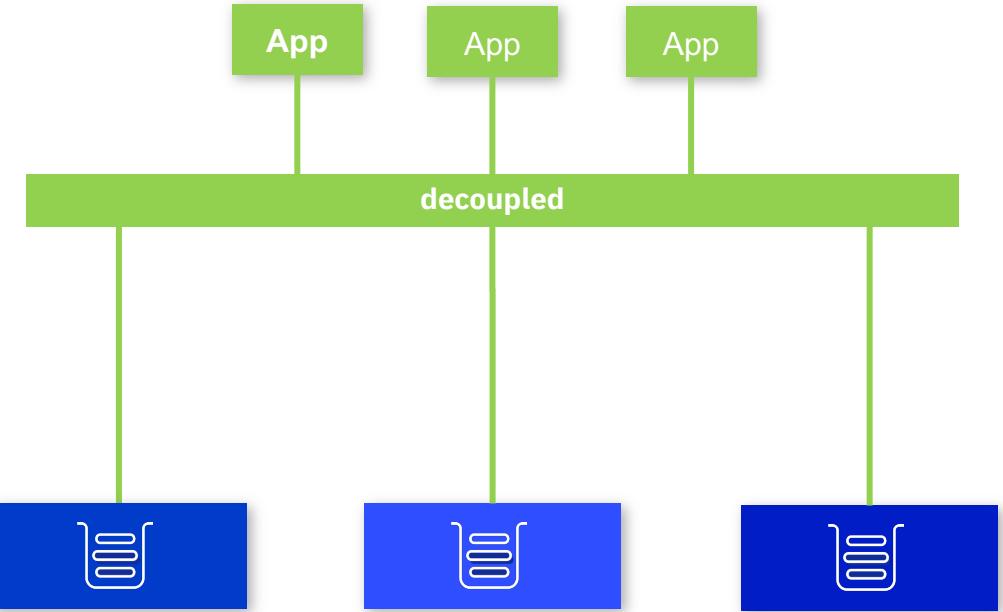
# Building scalable, fault tolerant, solutions

To provide an active/active, solution you need to consider multiple active queue managers acting as a *single service*

Applications also run as multiple instances for availability and scale

Applications treat the queue managers as interchangeable and want to connect to the group in the most efficient and available distribution

MQ introduced the **Uniform Cluster** capability across the 9.1.x CD releases to enable such deployments much more easily



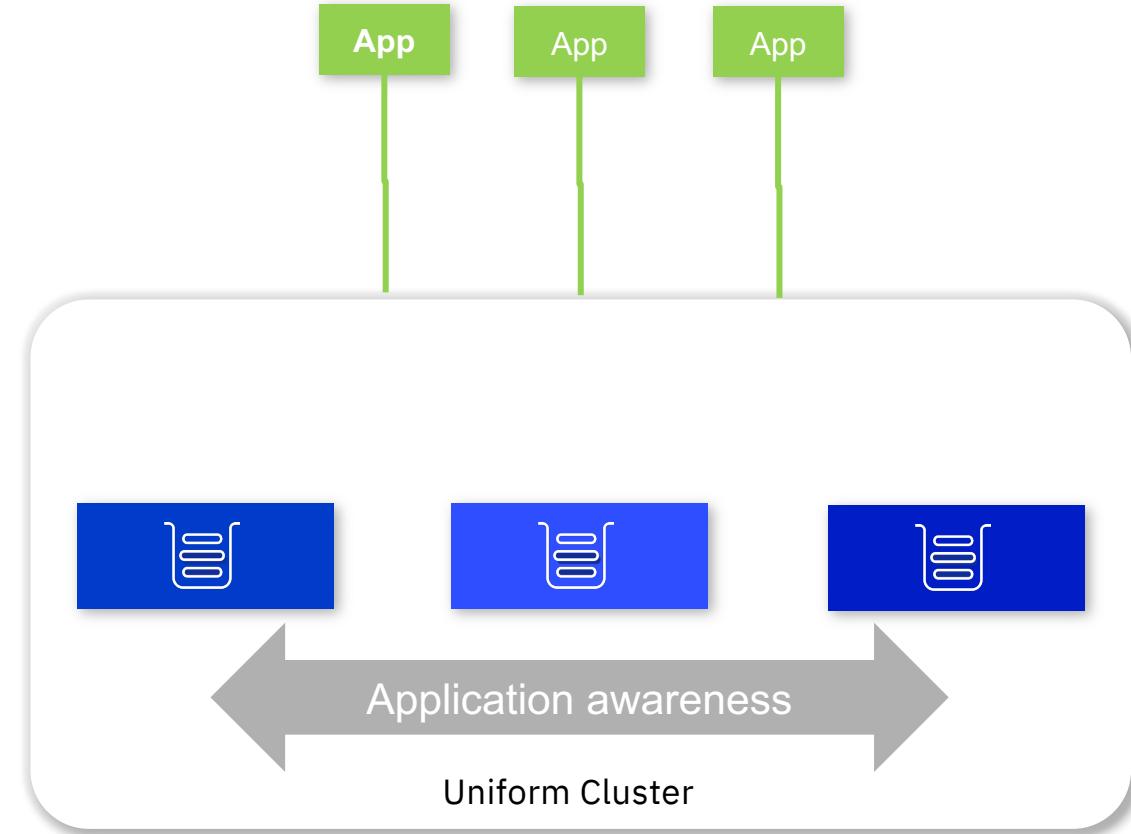
# Building scalable, fault tolerant, solutions

Uniform Clusters are a special type of MQ Cluster. One where all the queue managers provide the same service, such as queues.

Application instances are dynamically distributed across the available queue managers, adjusting as queue managers and application instances stop and start.

A uniform cluster builds on top of existing MQ building blocks -

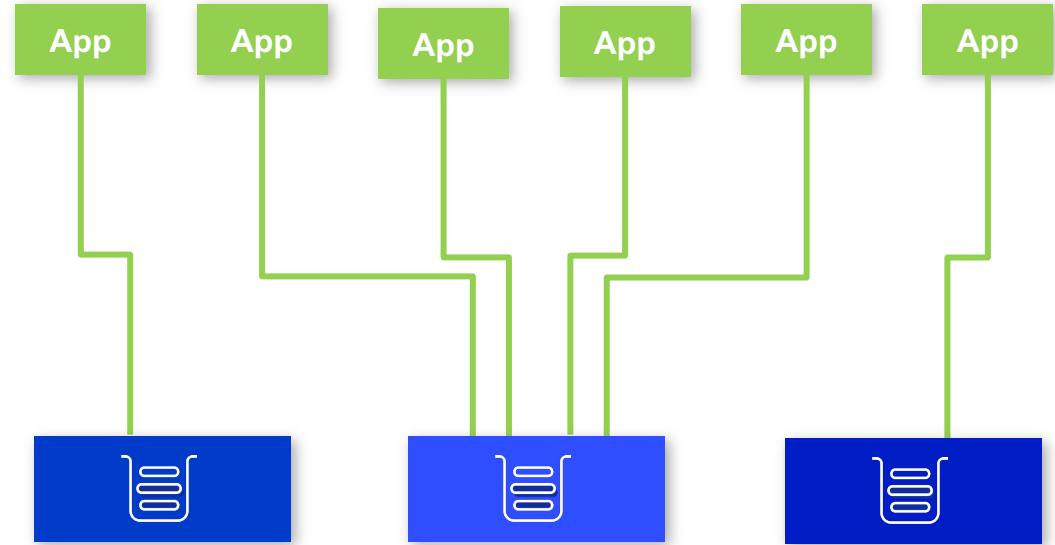
- Client auto-reconnect
- CDDT queue manager groups
- MQ Clustering



# Automatic Application balancing

When multiple instances of the *same\** application connect into a Uniform Cluster they may connect to any of the queue managers

Often, the connections will not be evenly distributed\*\*



\* Based on application name

\*\* For example when using a CCDT queue manager group

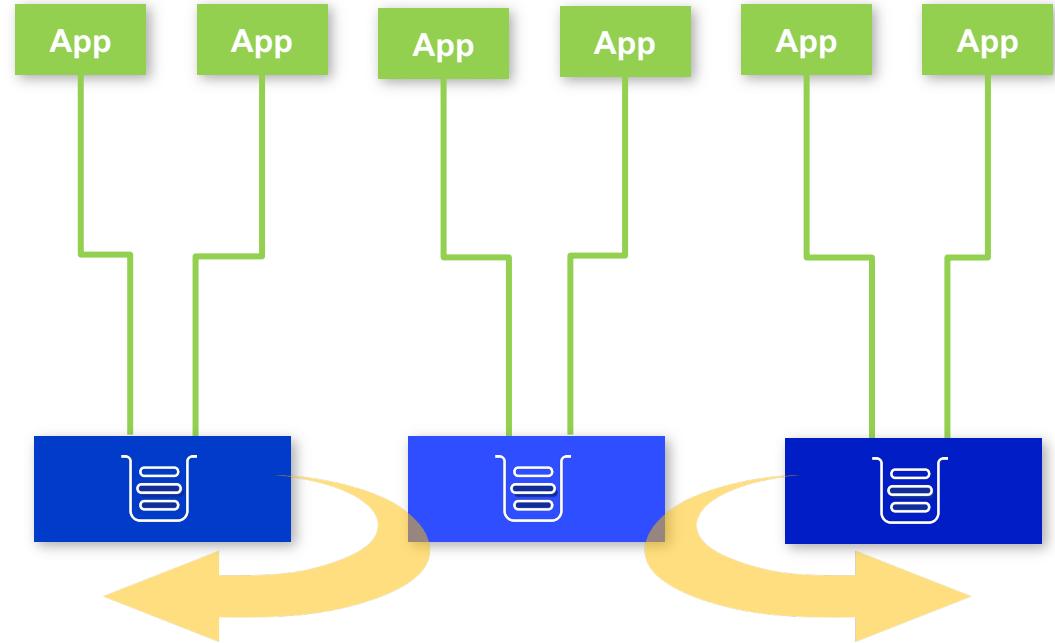
# Automatic Application balancing

When multiple instances of the *same\** application connect into a Uniform Cluster they may connect to any of the queue managers

Often, the connections will not be evenly distributed\*\*

The Uniform Cluster is constantly monitoring the connections and will automatically move connections between queue managers to maintain an even balance

The applications do not see the reconnection occurring



\* Based on application name

\*\* For example when using a CCDT queue manager group

# Automatic Application balancing

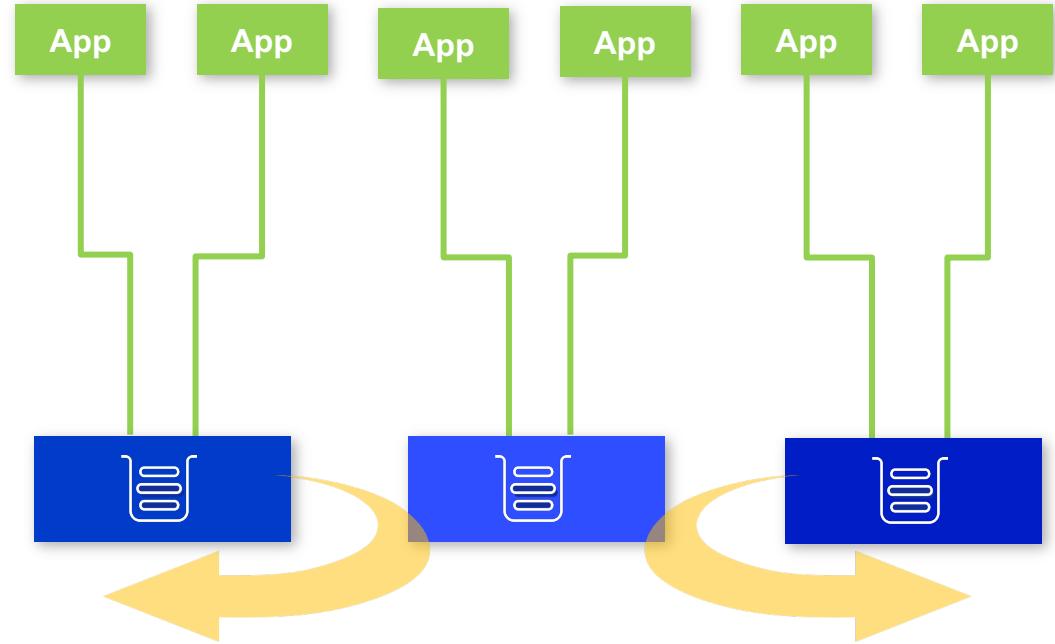
When multiple instances of the *same\** application connect into a Uniform Cluster they may connect to any of the queue managers

Often, the connections will not be evenly distributed\*\*

The Uniform Cluster is constantly monitoring the connections and will automatically move connections between queue managers to maintain an even balance

The applications do not see the reconnection occurring

This even occurs when queue managers stop and start or are added to the cluster



\* Based on application name

\*\* For example when using a CCDT queue manager group

# Automatic Application balancing

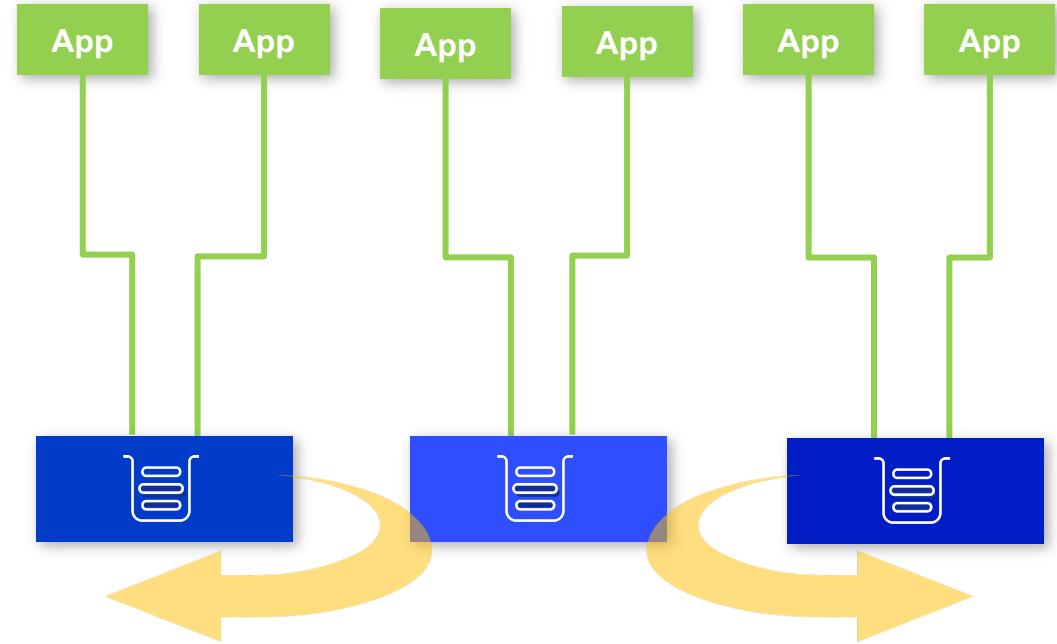
When multiple instances of the *same\** application connect into a Uniform Cluster they may connect to any of the queue managers

Often, the connections will not be evenly distributed\*\*

The Uniform Cluster is constantly monitoring the connections and will automatically move connections between queue managers to maintain an even balance

The applications do not see the reconnection occurring

This even occurs when queue managers stop and start or are added to the cluster



See it in action [ibm.biz/UniClusterDemo](http://ibm.biz/UniClusterDemo)

\* Based on application name

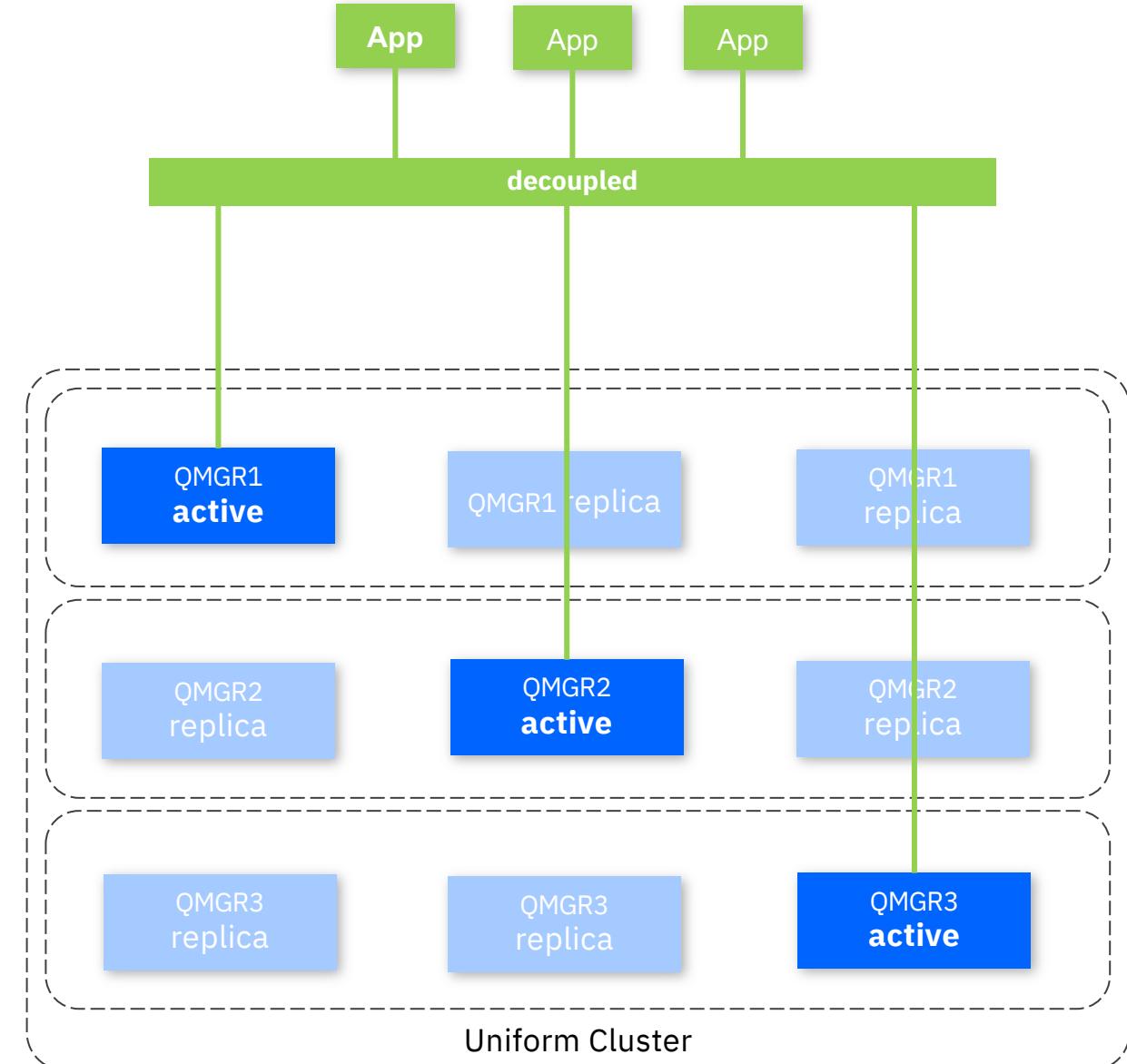
\*\* For example when using a CCDT queue manager group

# Combining message and service availability

Concepts like fault tolerance with RDQM or Native HA go hand in hand with active/active patterns like Uniform Clusters always ensure the highest levels of availability and scalability

There is always somewhere to connect to for new messages

Existing messages are fully recoverable within a few seconds

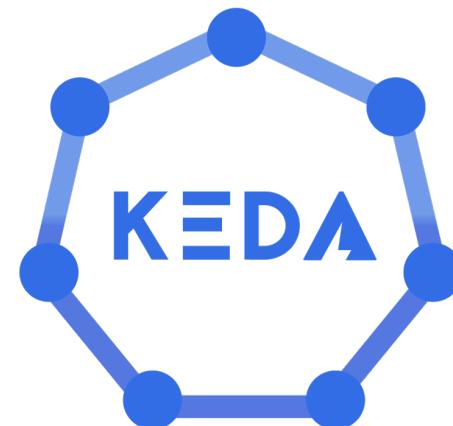
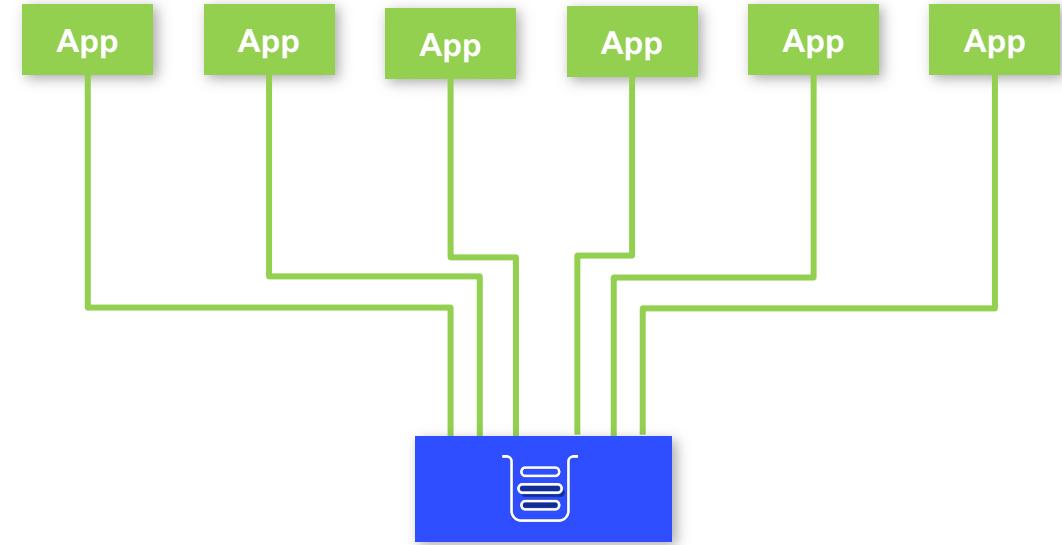


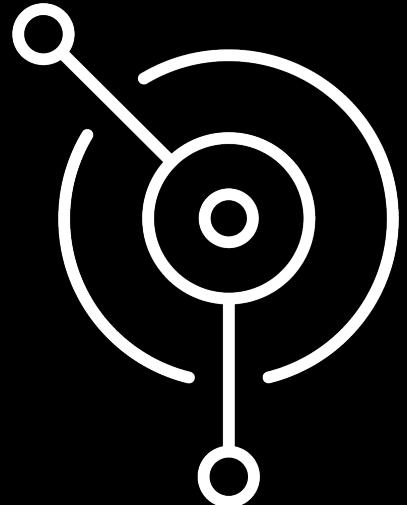
# Application autoscaling in Kubernetes

IBM MQ supports any number of application instances, with no need to pre-configure the system. This allows application instances to grow and shrink as needed before needing to scale the messaging layer

KEDA enables Kubernetes-based **event-driven autoscaling**, automatically increasing and decreasing the number of instances of an application based on the number of messages that need to be processed

[keda.sh/docs/2.0/scalers/ibm-mq/](https://keda.sh/docs/2.0/scalers/ibm-mq/)





# Managing MQ

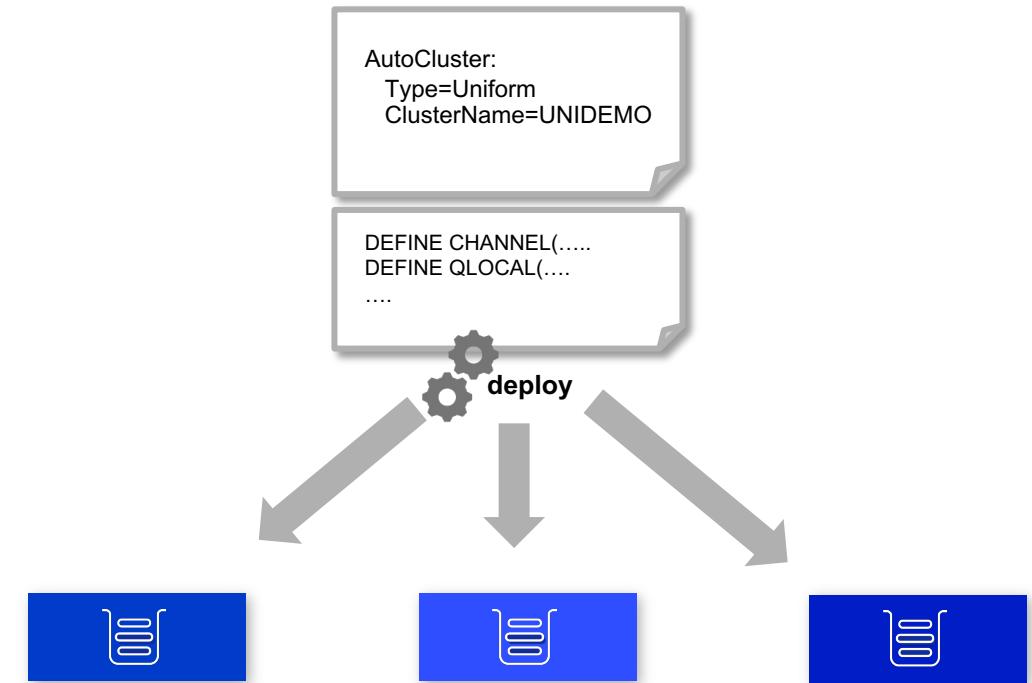
Administering and securing your messaging system

# Automation with MQ

Scripting is key to automation. MQ has supported scripting through MQSC scripts.

MQ has been evolving to make this even easier

- **Remote runmqsc** enables scripts to be deployed from a system remote to the queue manager
- MQSC commands are now more idempotent
- Queue managers can now automatically pull in updated **MQSC scripts** and **ini file** settings at start time (**MQ 9.1.4**)
- New **REST API** support opens up administration over HTTP using JSON (**MQ 9.1.3**)



# REST administration

# JSON format, MQSC style, REST commands

Send request body in **HTTP POST** to  
**admin/action/qmgr/{qmgrName}/mqsc**  
resource

## New command type of “**runCommandJson**”

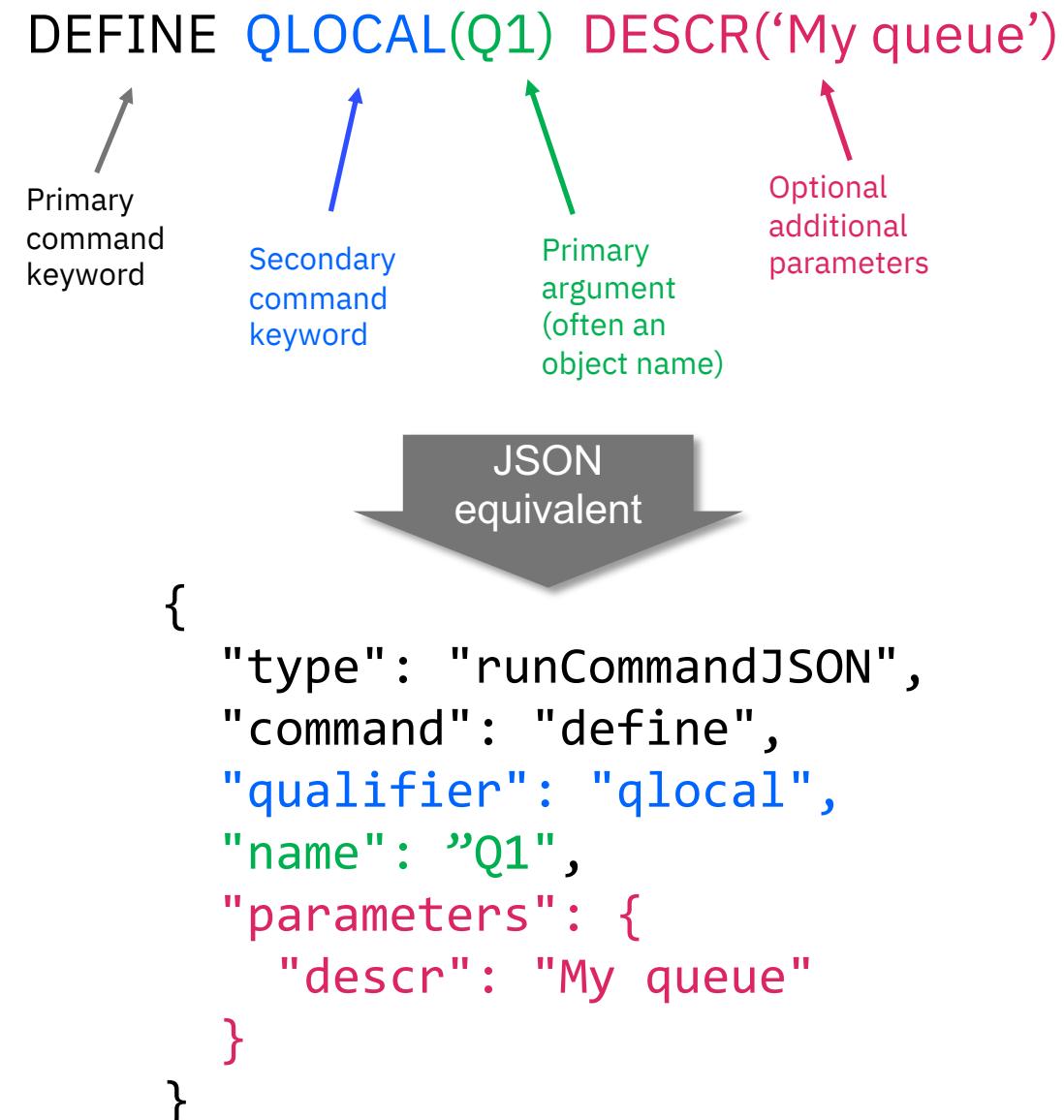
Existing command type of “**runCommand**” can still be used to run a plain text MQSC command

MQ 9.1.5 CD carried these APIs over into a new **V2** of the REST API. Earlier APIs for per-object manipulation have been stabilised at V1

[www.ibm.com/support/knowledgecenter/en/SSFKSJ\\_9.1.0/com.ibm.mq.pro.doc/q133690.htm#q133690](http://www.ibm.com/support/knowledgecenter/en/SSFKSJ_9.1.0/com.ibm.mq.pro.doc/q133690.htm#q133690) restapi/v2

# IBM MQ 9.1.3 CD

## All Platforms



# Code as Config for the applications too

Applications should never encode the MQ connection details, not even the queue manager

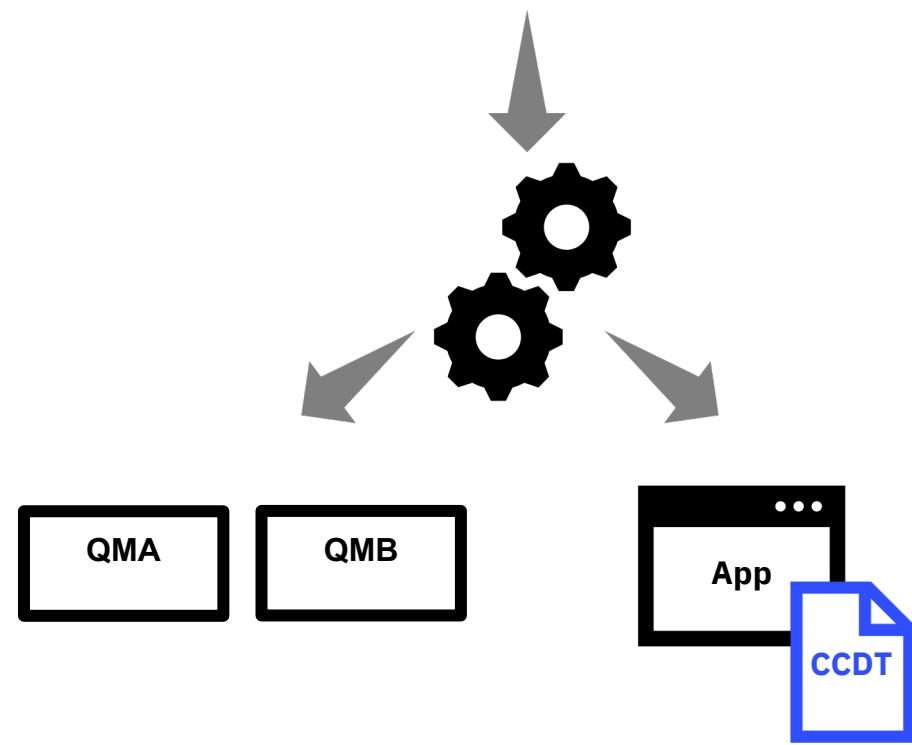
MQ CCDTs encapsulate the connection details

You can now build your own JSON format CCDTs

These can be deployed as part of the same pipeline that deploys your queue managers and applications

Supports multiple channels of the same name on different queue managers to simplify the building of uniform clusters

IBM MQ 9.1.2+ CD Clients



```
{  
  "channel": [  
    {  
      "name": "ABC",  
      "queueManager": "QMA"  
    },  
    {  
      "name": "ABC",  
      "queueManager": "QMB"  
    },  
  ]  
}
```

# Enabling automated delivery pipelines

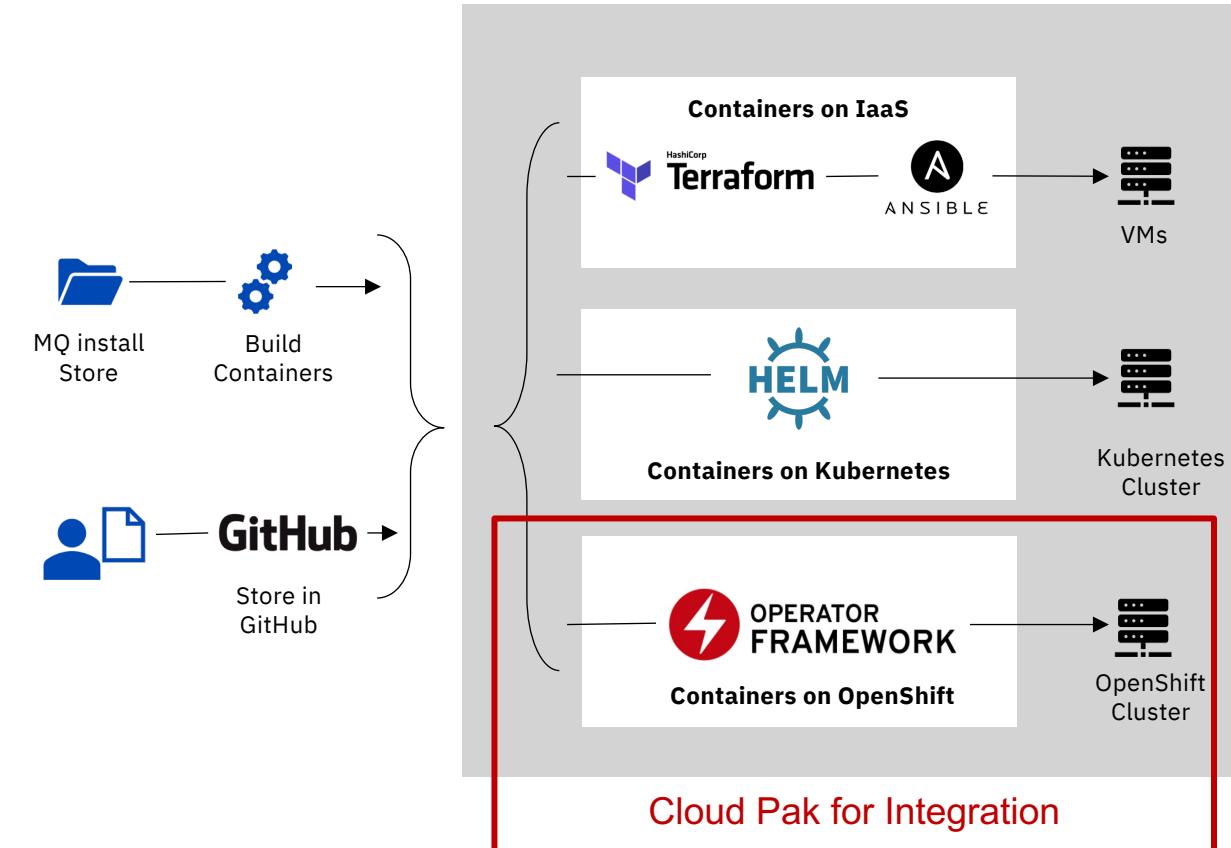
All MQ configuration treated a code

MQ queue manager deployments  
disposable and can be recreated

MQ container images are regularly  
upgraded

MQ's OpenShift Operator makes  
upgrades a one-line update

## Container options



## Managing channel CipherSpecs

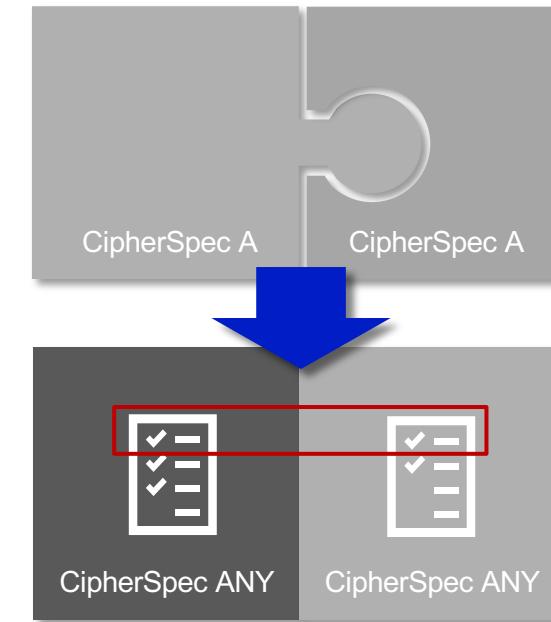
You no longer need to configure a matching single CipherSpec on both ends of a channel

MQ has introduced a range of **ANY\_TLSxx** and **ANY\_TLSxx\_OR\_HIGHER** CipherSpecs and MQ will negotiate the most preferred CipherSpec available to both ends

Migration between versions and ease of configuration are both simplified

If you need more control over the exact CipherSpecs appropriate to a queue manager, this can be configured

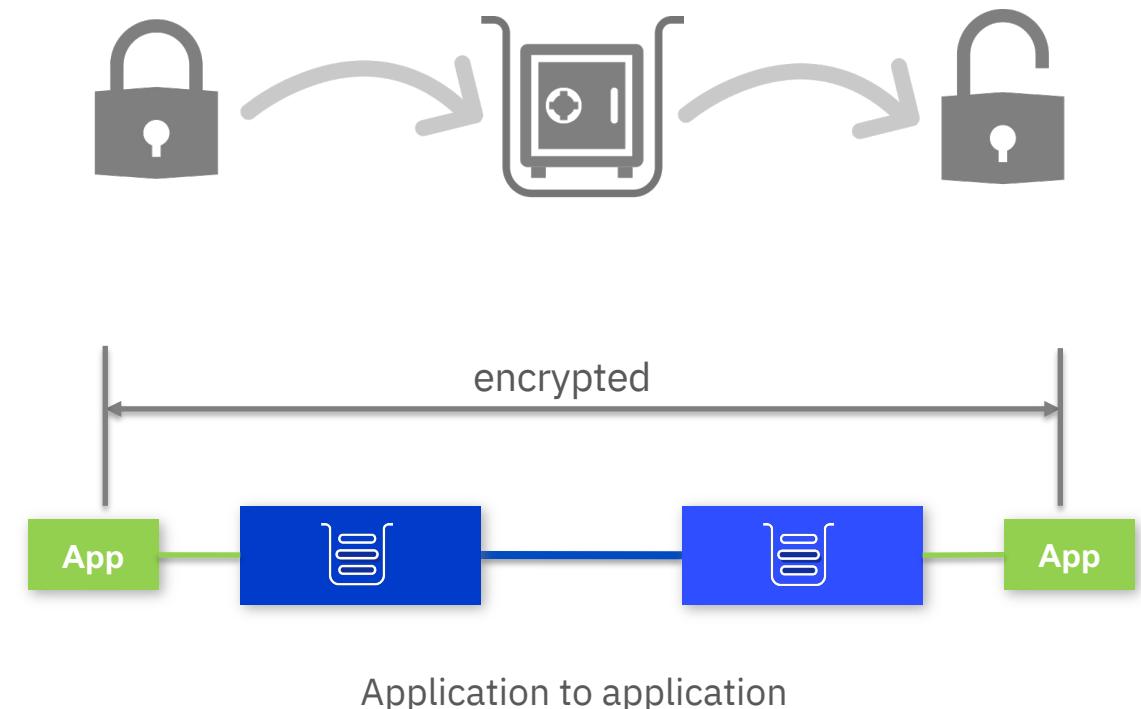
MQ is expanding its support of TLS 1.3 across its range of platforms, languages and protocols



# Advanced Message Security

End-to-end application-to-application encryption may give you the highest level of security, but it's not always possible to use. For example, where the applications are not AMS enabled or where the originators or recipients of the messages are outside of your domain

IBM MQ Advanced  
All platforms

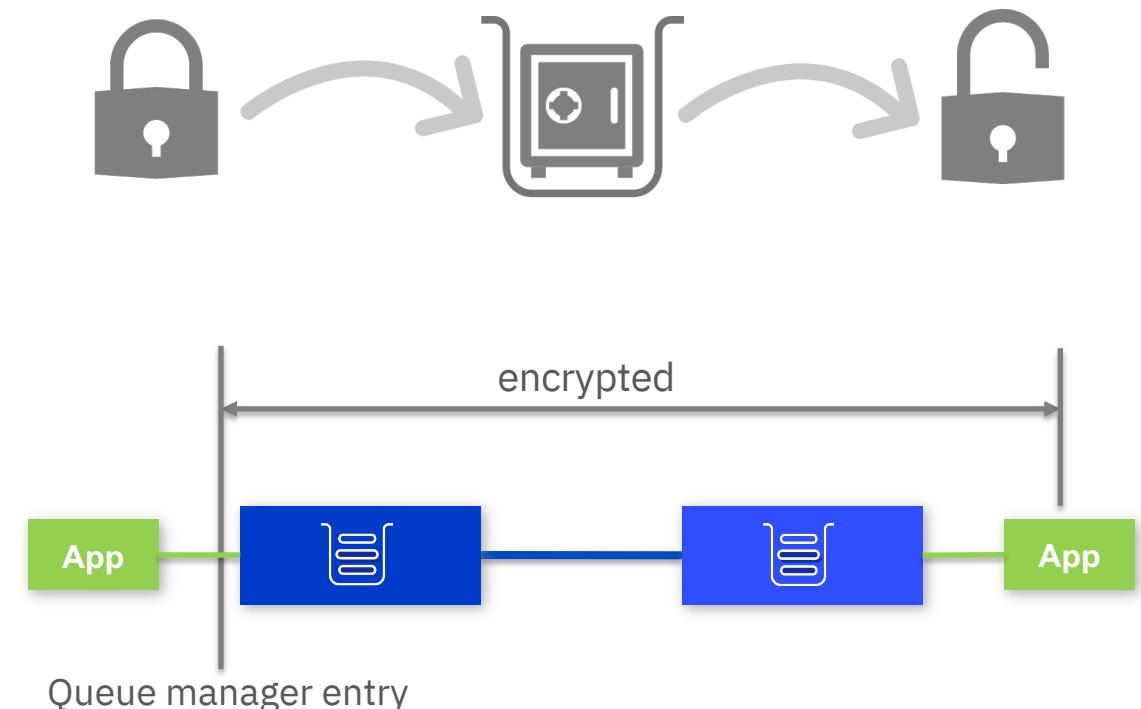


# Advanced Message Security

End-to-end application-to-application encryption may give you the highest level of security, but it's not always possible to use. For example, where the applications are not AMS enabled or where the originators or recipients of the messages are outside of your domain

MQ on Distributed has always had client level interception to apply AMS policies once messages reach or leave their first queue manager

IBM MQ Advanced  
Distributed



# Managed File Transfer

IBM MQ Advanced  
All platforms

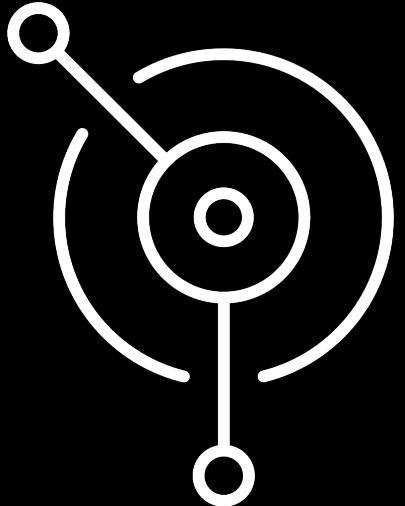
MFT manages your file transfers, with file-to-file and file-to-message.

MQ has focused on resiliency and ease of administration.

Active/standby MFT agent support adds highly available topologies, along with fine grain control over resource monitors

Expanded the breadth of REST APIs, both for monitoring and configuring MFT resources and for initiating file transfers.





# Helping developers

Making it easy to build MQ into your applications

# Getting Started

## A site for Developers

**ibm.biz/MQDevCenter**

Teach yourself the basics of MQ

**ibm.biz/learn-mq**

Build on top of simple samples

**ibm.biz/mq-dev-patterns**

...and prove your skills

IBM MQ

Get started with IBM MQ

Get IBM MQ

Articles

Series

Tutorials

Videos

Community

Blog Posts

Related topics

Messaging

More resources

IBM MQ on ibm.com

IBM MQ Samples

IBM Community: IBM MQ

IBM MQ on Stack Overflow

Series

Learning path: IBM MQ Developer Essentials Badge

Learn how to build powerful messaging applications with IBM MQ.

By Richard J. Coppen  
Updated June 23, 2020 | Published August 3, 2018

Facebook Twitter LinkedIn Email

Products & Services (1)

IBM MQ



Learning objectives

- Understand IBM MQ messaging concepts
- Create and configure a queue manager, queue, and topic
- Develop a simple point-to-point JMS application that can connect and interact with the queue manager
- Demonstrate your learning and skills by taking up a coding challenge and developing a solution to a problem
- Troubleshoot and debug your application

Follow the instructions in these tutorials to get started with a queue manager:

- In a [container](#).
- In the [IBM Cloud](#).
- On various operating systems: [Linux/Ubuntu](#) or [Windows](#). For MacOS, use MQ on Containers.

If you know what to do, or if you need a different version or a different IBM MQ server installer, see the [IBM MQ for administrators downloads page](#).

To download the MQ server or MQ Clients from IBM Fix Central, you'll need to log in with your IBMID, so make sure that you [register for IBMID](#) first. After you encounter issues with Fix Central, try a different browser.

Get MQ client libraries

To develop MQ applications, you need a queue manager, and these three things:

1. MQ client libraries for your language and platform



# Developing applications

Build your applications simply, with no need for an MQ installation

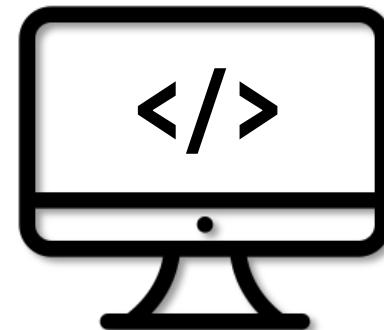
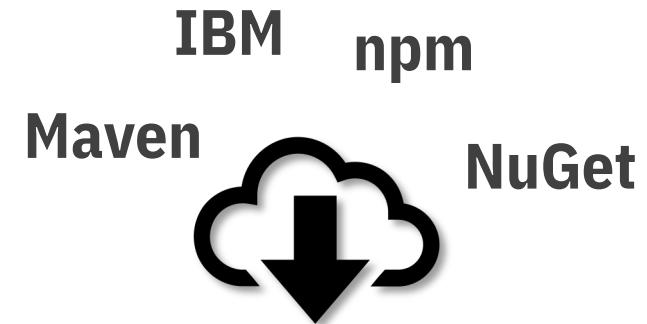
Pull Java directly from the **Maven** repository and .NET from NuGet

The **SDK** has been added to the MQ redistributable client, removing the need to install it to build from

**[ibm.biz/mq-downloads](http://ibm.biz/mq-downloads)**

Develop your applications on the platform of your choice for free

The full MQ Advanced for Developers is available on Windows and Linux with the addition of a MacOS MQ client and SDK for Developers



# Writing new applications

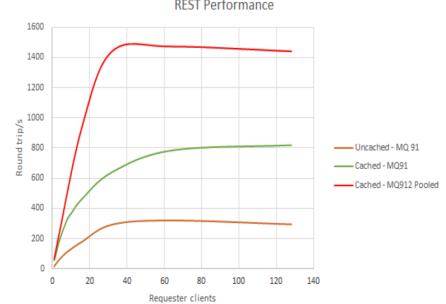
## REST Messaging

Providing a very simple way to get messages in and out of your MQ system

9.1.2 CD boosted the performance capability, 9.1.3 CD added message browse and 9.1.5 CD added publish



REST



put, get, browse,  
publish

## .NET Core

9.1.1 CD brought support for .NET Core on Windows

9.1.2 CD added Linux support  
and now you can develop on MacOS



.NET Core

## Open Source language bindings

Write MQI applications in Node.js and Golang

New simpler JMS style API for Golang

[github.com/ibm-messaging](https://github.com/ibm-messaging)



Node.js MQI



Golang MQI



Golang JMS

## AMQP 1.0

Support for AMQP 1.0 clients to connect and interoperate with other MQ applications. Messaging behaviour follows Apache Qpid JMS, widening the choice of open source clients



AMQP



Apache Qpid JMS

NEW  
9.2.1

# IBM MQ, supporting transformation

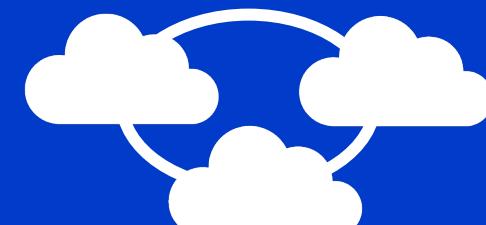
## Developer Agility

Delivery teams are being empowered within the organization. They need to be enabled to complete their day to day operations independently.



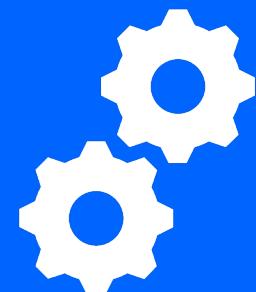
## Adopt Multi-Cloud

Delivery teams are empowered to select their cloud of choice, and expect connectivity to be provided across these.



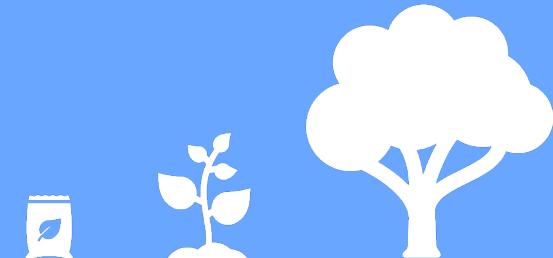
## Operational Agility

IBM MQ operational teams are being challenged to simplify the management of their infrastructure to drive cost savings.



## Organic Growth

Organizations that originally deployed IBM MQ for a project that has matured, and now need improved scalability, availability and security.



# How can customers buy MQ?

## Customizable software

MQ software for use on premise, or in the cloud



## MQ, MQ Advanced



## Preoptimized simplicity

Top specification hardware appliance – get going in minutes!



## MQ Appliance



## Reliability and strength

MQ customized specifically for the mainframe



## MQ for z/OS



## Enterprise messaging in the cloud

Be our guest with our managed service



## MQ on IBM Cloud



One consistent solution across different form factors

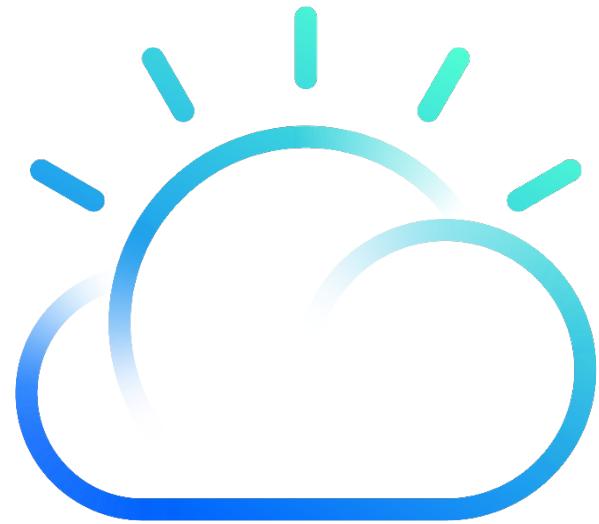
# The reliability and security of MQ wherever you are



On premise



Clouds



Thank You