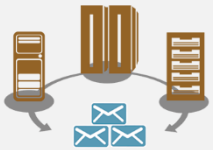
Hybrid messaging with MQ Light, MQ's beta support for AMQP, and Bluemix

Matthew Whitehead
WebSphere MQ Development
mwhitehead@uk.ibm.com

Agenda

- Quick recap of MQ Light
- How MQ Light and AMQP applications work with IBM MQ (beta)
- Hybrid Messaging with MQ and Bluemix

IBM Messaging



Enterprise
Messaging

Deliver Messaging Backbone for Enterprise

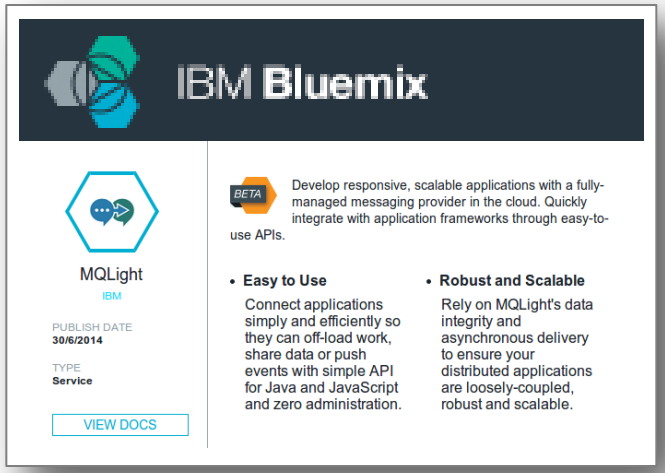
Focus on traditional MQ values, rock-solid enterprise-class service, ease-of-operation, breadth of platform coverage, availability, z/OS exploitation

Application
Messaging

Enable Developers to build more scalable, responsive applications

Focus on app use cases, breadth of languages, ease-of-deployment, micro services, integration with developer frameworks

MQ Light : Software and Cloud



- **Messaging that application developers will love to use, helping them make responsive applications that scale easily**
- **3 ways to get it:**
  - **Bluemix service**
  - MQ Light software download
  - **MQ Version 8.0.0.3 beta support for MQ Light**

THIS PRESENTATION

- **Open APIs crafted to feel natural in a growing range of popular languages**
- **Tooling that makes modular app development easy**

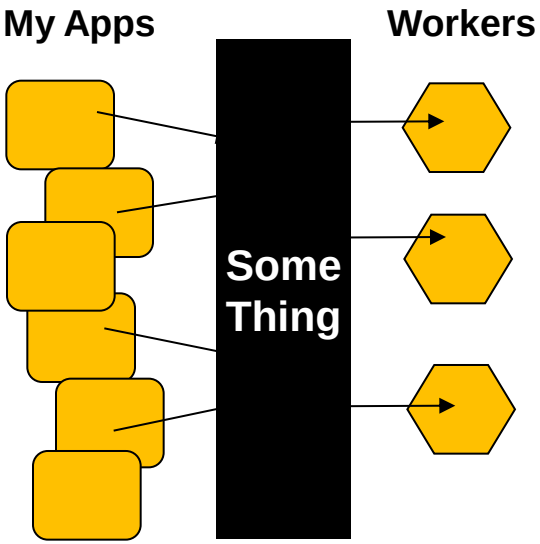
The journey that got us here

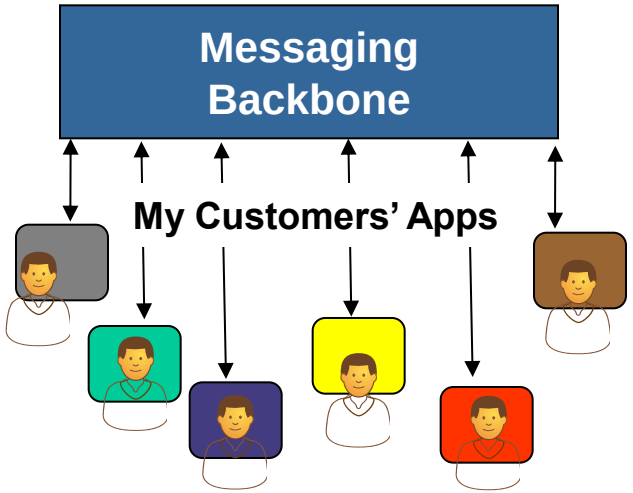
I want to execute code without taxing my Web app processes

Andy
Developer

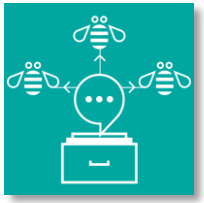
Iain
Infrastructure Guy

My job is run a communications service for my customers' apps

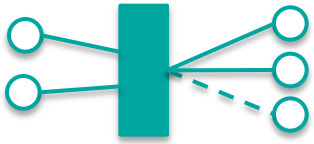




Use cases

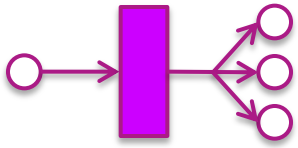


Worker offload



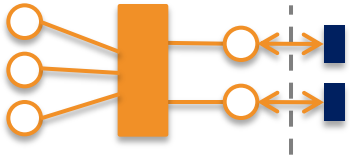
- Image processing
- Text analytics

Event driven



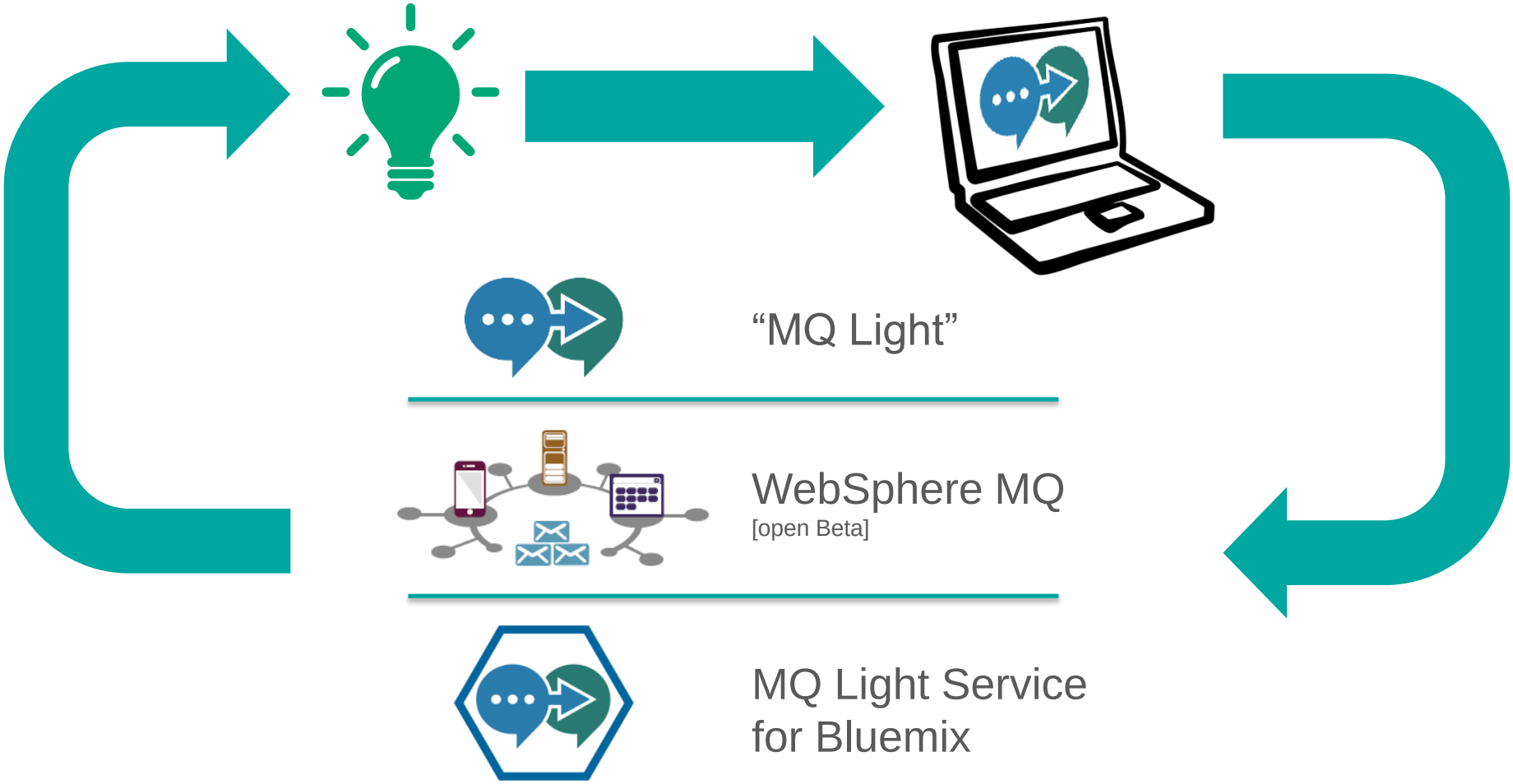
- Posting video to multiple social sites after transcoding
- Respond to external events

Connecting external systems



- Updating external booking app
- Posting updates to twitter

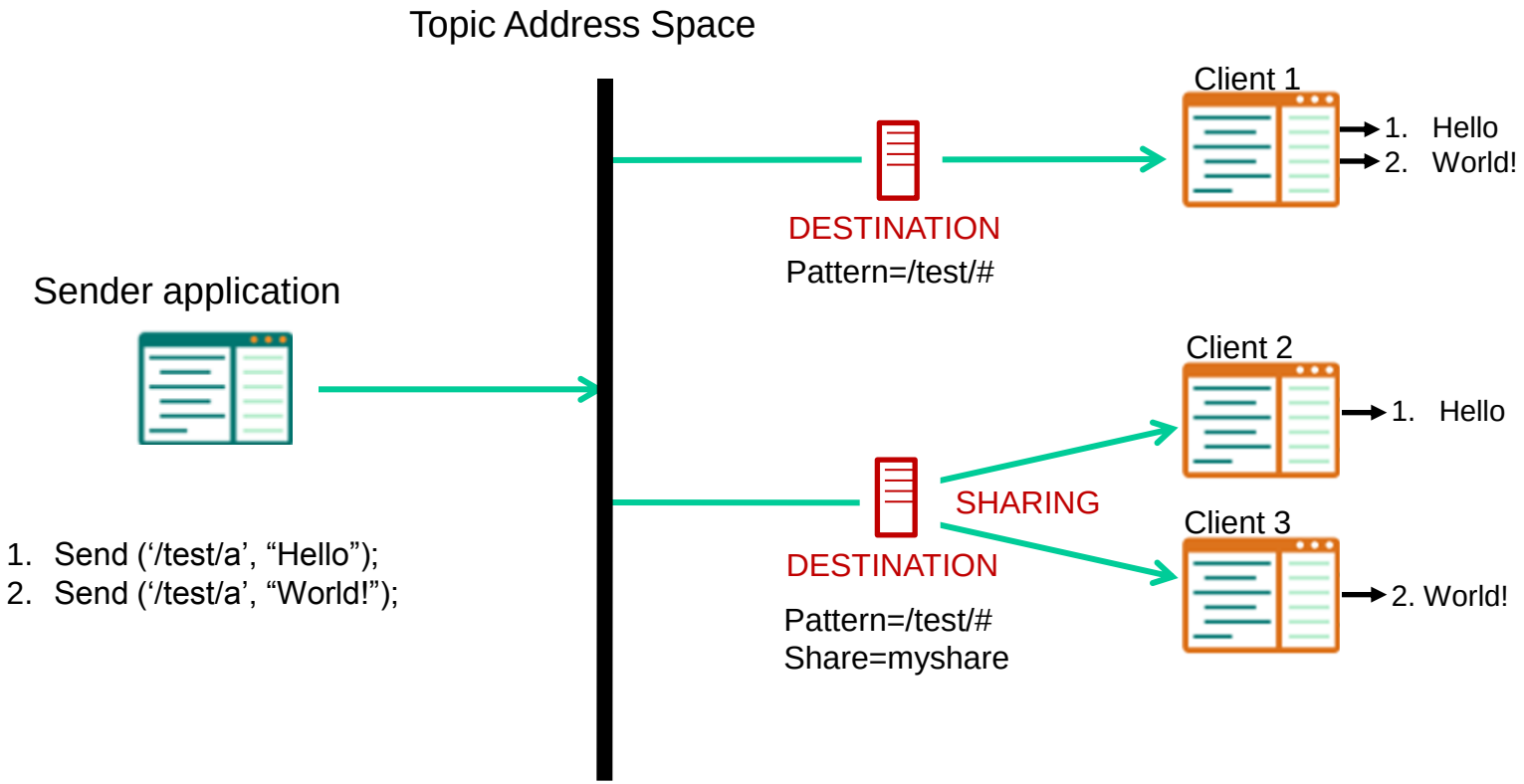
Deployment Options



MQ Light API - Language support

- Simple, programming Language neutral messaging model
- Idiomatic language & framework API Mappings
  - Frictionless development
- Open wire protocol
  - Open Source client libraries
  - Facilitates community drivers for languages & frameworks

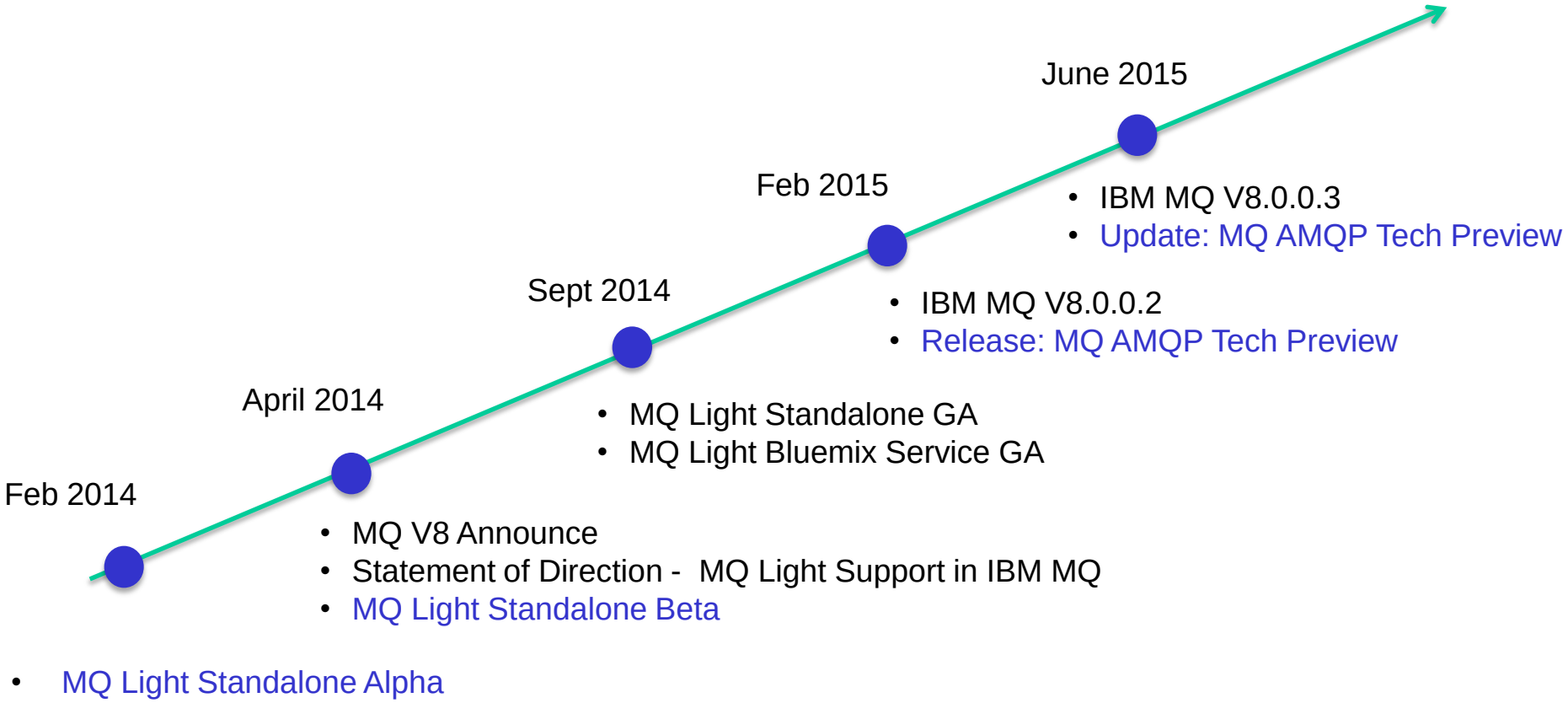
MQ Light Messaging Model



Agenda

- Quick recap of MQ Light
- How MQ Light and AMQP applications work with IBM MQ (beta)
- Hybrid Messaging with MQ and Bluemix

MQ Light Support in IBM MQ (Beta)

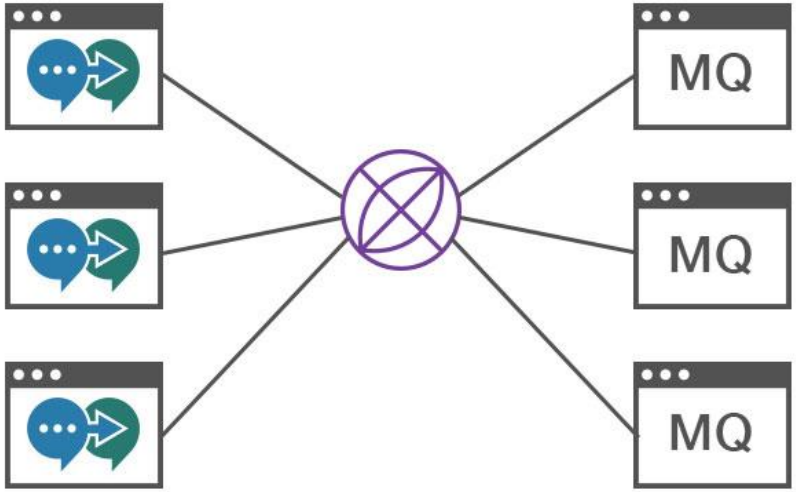


IBM MQ – MQ Light Tech Preview (Beta)

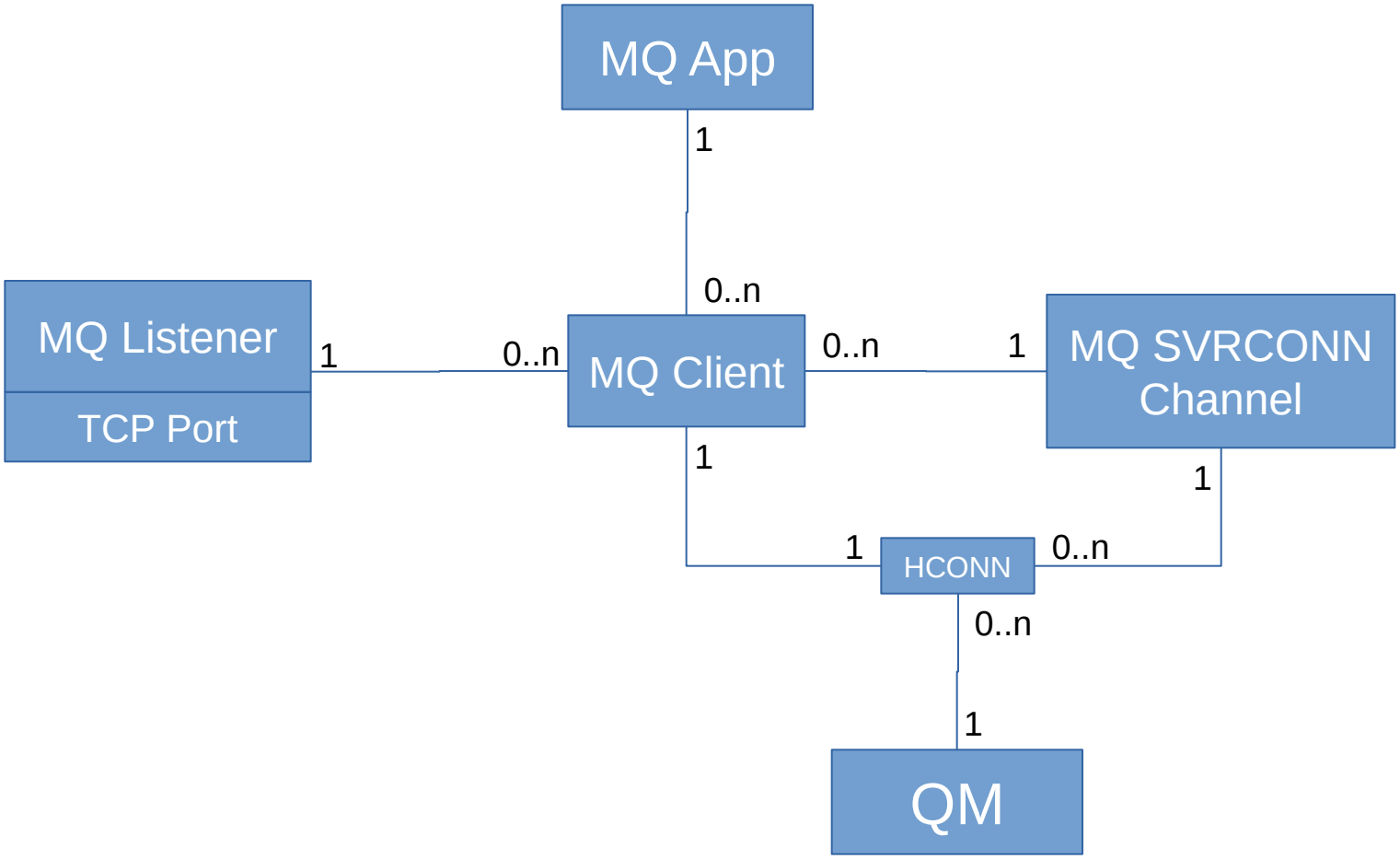
- **Platforms**
  - Windows 64 Bit
  - Linux x86_64
- **Beta Installation**
  - Get code/instructions from MQ Dev Blog: http://bit.ly/mqlightmq [1]
  - Prereq is IBM MQ V 8.0.0.3
  - Add Tech Preview install media
    - Linux – RPM which is installed along side the other MQ RPMs
    - Windows – Zip which is manually extracted to an MQ installation

New AMQP channel type (Beta)

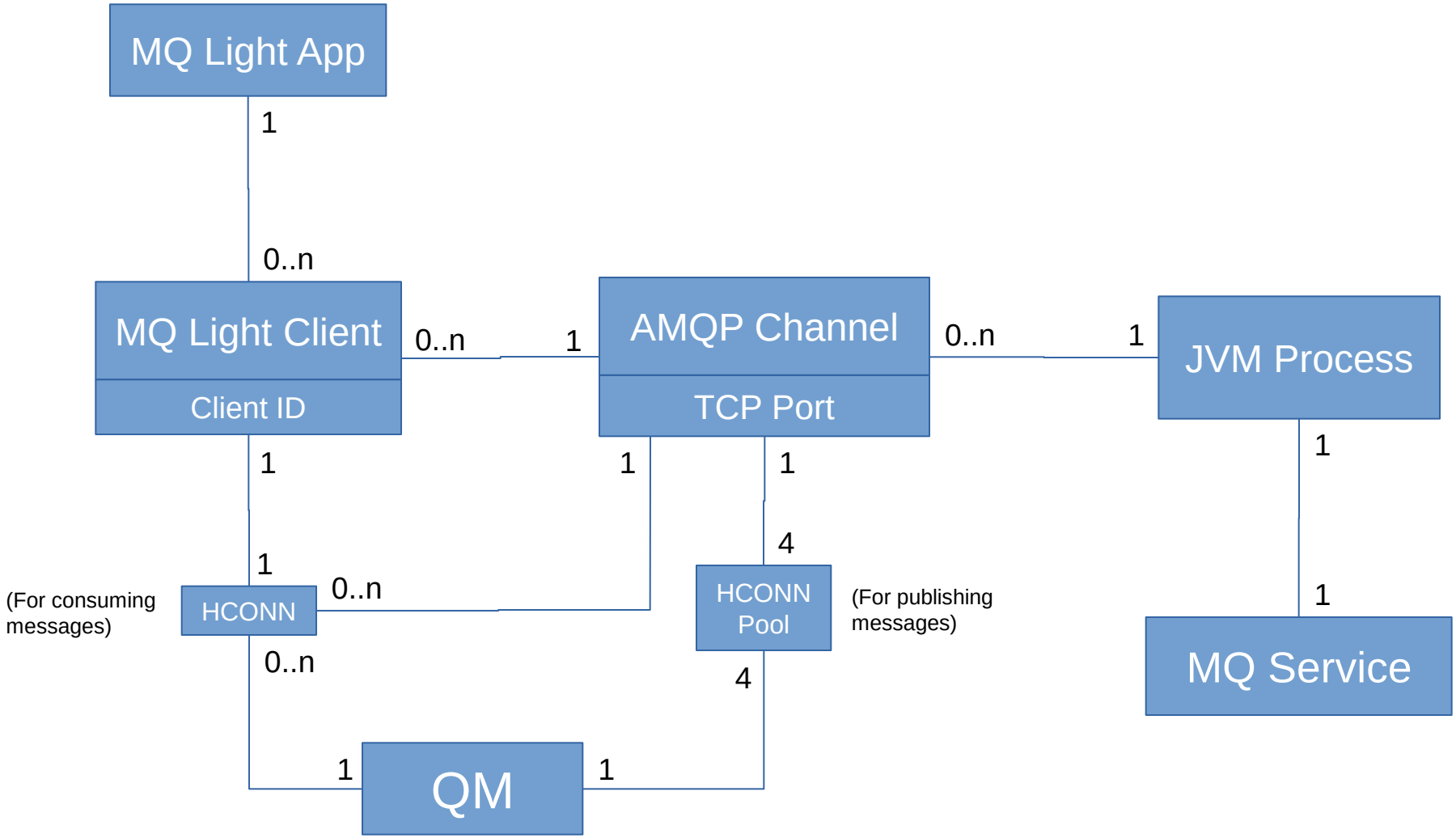
- Adds a channel type of “AMQP”
- Support a subset of the AMQP 1.0 Oasis specification
- Interoperable with MQ FAP and MQTT applications (see later slides for details)



The architecture of an MQ server-conn channel (Beta)



The architecture of an MQ Light channel (Beta)



AMQP channels (Beta)

- Configuration model
  - MQSC and PCF updates allow you to administer AMQP channels in much the same way as other MQ objects
    - DISPLAY CHANNEL(*) CHLTYPE(AMQP)
    - DEFINE CHANNEL(MY.AMQP.CHANNEL) CHLTYPE(AMQP) PORT(5673)
    - START CHANNEL(MY.AMQP.CHANNEL)
    - STOP CHANNEL(MY.AMQP.CHANNEL)
    - DISPLAY CHSTATUS(*) CHLTYPE(AMQP)
  - PCF command types all valid, using MQIACH_CHANNEL_TYPE=MQCHT_AMQP:
    - MQCMD_CREATE_CHANNEL MQCMD_DELETE_CHANNEL, MQCMD_CHANGE_CHANNEL MQCMD_START_CHANNEL, MQCMD_STOP_CHANNEL MQCMD_COPY_CHANNEL, MQCMD_INQUIRE_CHANNEL MQCMD_INQUIRE_CHANNEL_STATUS

AMQP channels (Beta)

mwhitehead@ubuntu-vm: /opt/mqmAMQP

DIS CONN command

```
      :

      :
dis conn(*) type(conn) where (channel eq myamqp) all
    10 : dis conn(*) type(conn) where (channel eq myamqp) all
AMQ8276: Display Connection details.
   CONN(9658C45405250020)
   EXTCONN(414D5143514D35202020202020202020)
   TYPE(CONN)
   PID(3446)                                TID(8)
   APPLDESC(WebSphere MQ Advanced Message Queuing Protocol)
   APPLTAG(java)                            APPLTYPE(SYSTEMEXT)
   ASTATE(STARTED)                          CHANNEL(MYAMQP)
   CLIENTID(recv_729b7a3)                   CONNAME(192.168.56.1)
   CONNOPTS(MQCNO_HANDLE_SHARE_NO_BLOCK,MQCNO_FASTPATH_BINDING)
   USERID(mqm)                              UOWLOG( )
   UOWSTDA( )                               UOWSTTI( )
   UOWLOGDA( )                              UOWLOGTI( )
   URTYPE(QMGR)
   EXTURID(XA_FORMATID[] XA_GTRID[] XA_BQUAL[])
   QMURID(0.0)                              UOWSTATE(NONE)
```

Note the new client ID attribute set on the MQ connection

AMQP channels (Beta)

mwhitehead@ubuntu-vm: /opt/mqmAMQP

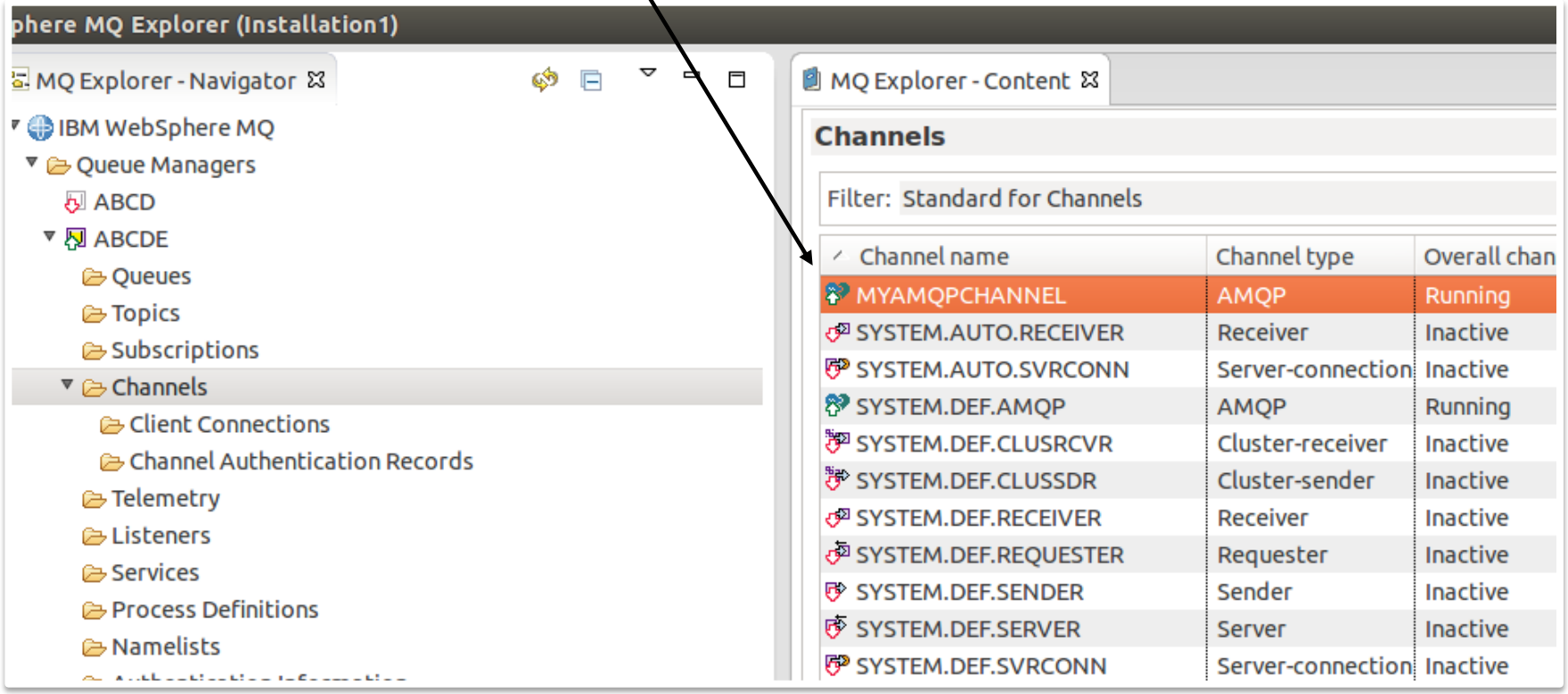
DIS CHSTATUS command

```
        :

        :
display chstatus(*) chltype(AMQP) clientid(*) all
     2 : display chstatus(*) chltype(AMQP) clientid(*) all
AMQ8417: Display Channel Status details.
   CHANNEL(MYAMQP)                          CLIENTID(recv_fced6d9)
   STATUS(RUNNING)                          CONNAME(192.168.56.1)
   KAINT(0)                                 MCAUSER(mwhitehead)
   CLNTUSER( )                              MSGSNT(3)
   MSGRCVD(0)                               INDOUBTIN(0)
   INDOUBTOUT(0)                            PENDING(0)
   LMSGDATE(2015-01-25)                     LMSGTIME(03.29.31)
   CHLSDATE(2015-01-25)                     CHLSTIME(02.46.16)
   PROTOCOL(AMQP)
```

Note, DIS CHSTATUS usage varies slightly from MQ channels

Administration – MQ Explorer (Beta)

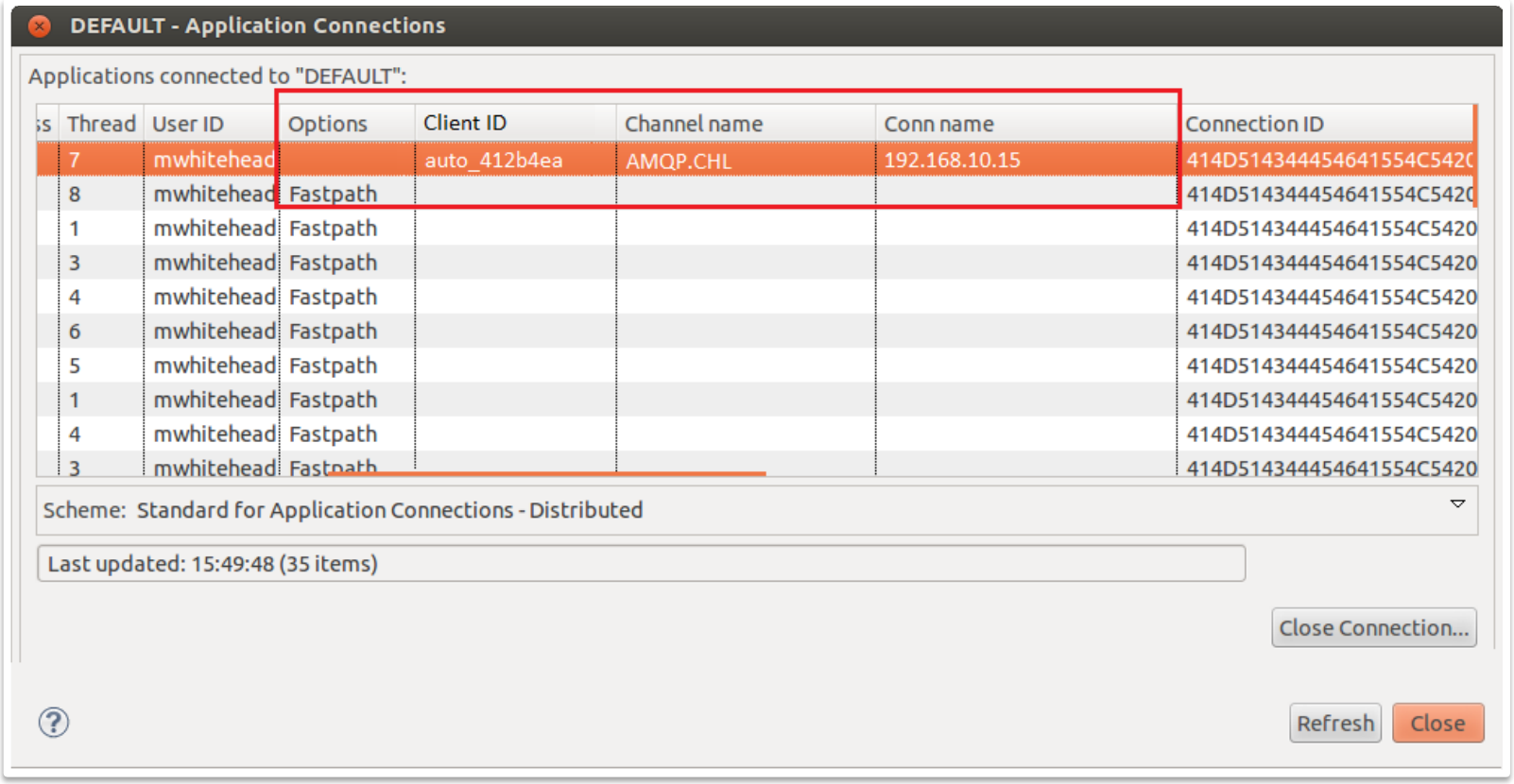
AMQP channels displayed alongside existing channels



Administration – MQ Explorer (Beta)

Application connections view use to display AMQP clients

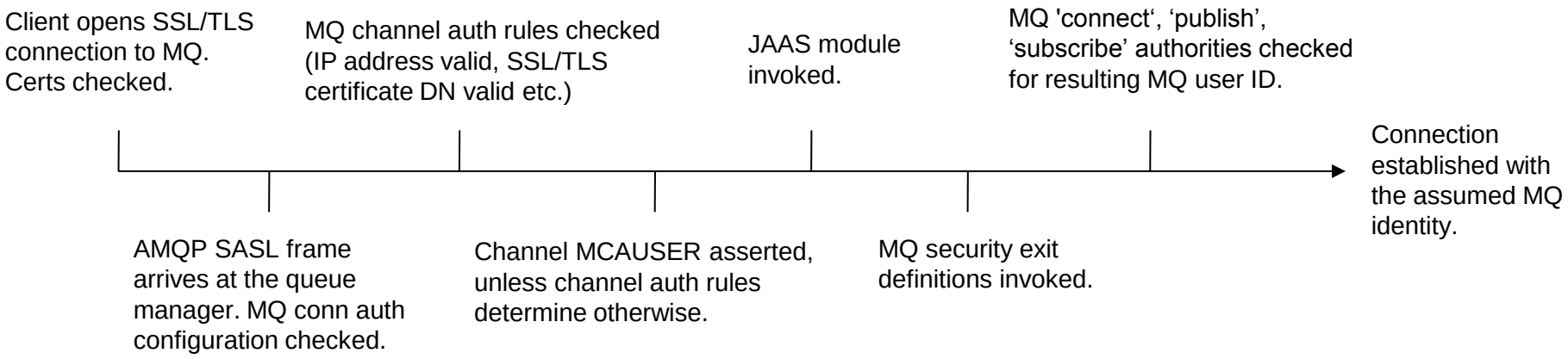
- Connection name and channel name populated to show where the client has connected from
- New **Client ID** attribute in the Application Connections view



Thread	User ID	Options	Client ID	Channel name	Conn name	Connection ID
7	mwhitehead		auto_412b4ea	AMQP.CHL	192.168.10.15	414D514344454641554C5420
8	mwhitehead	Fastpath				414D514344454641554C5420
1	mwhitehead	Fastpath				414D514344454641554C5420
3	mwhitehead	Fastpath				414D514344454641554C5420
4	mwhitehead	Fastpath				414D514344454641554C5420
6	mwhitehead	Fastpath				414D514344454641554C5420
5	mwhitehead	Fastpath				414D514344454641554C5420
1	mwhitehead	Fastpath				414D514344454641554C5420
4	mwhitehead	Fastpath				414D514344454641554C5420
3	mwhitehead	Fastpath				414D514344454641554C5420

Authorities and authentication (Beta)

- MQ authenticates the client and authorises messaging in a similar way to MQ clients. Similarly administration changes and commands for AMQP channels are authorised in the same was as for other MQ channels. Allows you to specify who has authority to:
  - Start or stop a channel
  - Change a channel's configuration
  - Display a channel's status
  - Delete a channel



Monitoring (Beta)

- Events
  - MQ provides events for monitoring different activities
  - Some are available to try in the beta
    - Command events (e.g. request to start a channel)
    - Configuration events (e.g. request to change channel attrs)
  - Some are ones we'd like to do
    - Security events (e.g. an AMQP client failed an authority check)

Administration (Beta)

- Backup/Restore
  - MQ provides tools to saving and restoring queue manager configuration
    - dmpmqcfg and runmqsc
  - These have been updated to include AMQP channel definitions
  - Service may request that custom tuning/service parameters be set in a .properties file. If so, then this file must be manually backed up similar to qm.ini files today
- Logs
  - Located in MQ data path
    - /var/mqm/qmgrs/QM1/**amqp.stdout** and /var/mqm/qmgrs/QM1/**amqp.stderr**
    - /var/mqm/qmgrs/errors/**amqp_*.log**
    - /var/mqm/trace/**amqp_*.trc** (start/end trace using strmqtrc/endmqtrc)

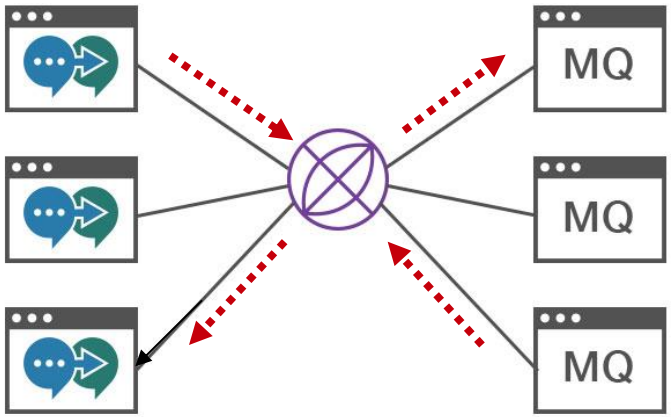
Interoperability (Beta)

- AMQP to MQ FAP and MQTT

warehouse/item/372837

warehouse/item/#

Published AMQP messages MQPUT to an MQ topic



MQ apps can **consume** AMQP publishes by subscribing to matching topic pattern

Consumed AMQP messages MQGET from an MQSUB

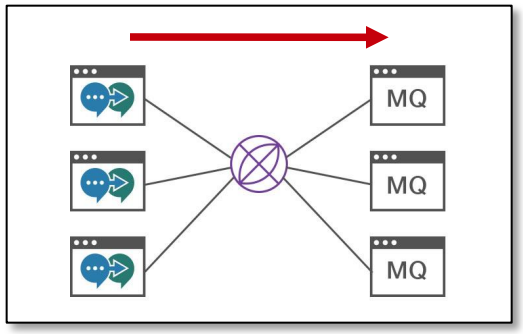
MQ apps can **publish** to AMQP clients by MQPUT to the same topic string

orders/electrical/#

orders/electrical/wiring

Interoperability (Beta)

- AMQP publisher to MQ consumer



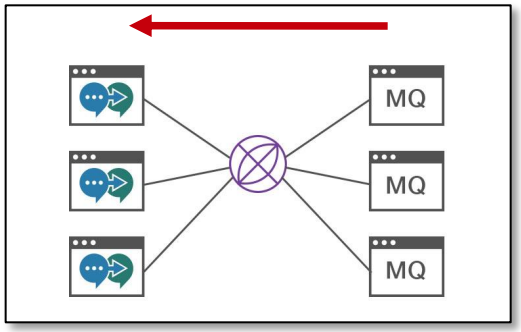
  - MQMD PutApplType always set to MQAT_AMQP

  - Some AMQP attributes → MQMD
  - Some AMQP attributes → MQ message properties
  - All AMQP application properties → MQ message properties

  - Simple AMQP binary payload → MQFMT_NONE message
  - Simple AMQP string payload → MQFMT_STRING message
  - All other AMQP payloads → MQFMT_AMQP

Interoperability (Beta)

- MQ publisher to AMQP consumer



  - Some MQMD fields → AMQP headers
  - Some MQMD fields → AMQP properties
  - All MQ message properties → AMQP application properties

  - MQFMT_NONE message → single AMQP binary data payload
  - MQFMT_STRING message → single AMQP string data payload
  - MQFMT_AMQP message → copy to payload section

Note: All MQ messages are got with MQGMO_CONVERT to convert string data to UTF8

Detail on Interoperability – AMQP to MQ (Beta)

Some AMQP **headers** are set as MQMD fields:

AMQP header.ttl	set on MQ message as MQMD.expiry (converted to 10ths of a second)
AMQP header.priority	set on MQ message as MQMD.priority (max value of 9)
AMQP properties.correlation-id	set on MQ message as MQMD.correlid

All AMQP **headers** are set as MQ message properties with a mapped name:

header.durable	set on MQ message as MQ propertyAMQPDurable
header.priority	set on MQ message as MQ propertyAMQPPriority
header.ttl	set on MQ message as MQ propertyAMQPTtl
header.first-acquirer	set on MQ message as MQ propertyAMQPFirstAcquirer
header.delivery-count	set on MQ message as MQ propertyAMQPDeliveryCount

All AMQP **properties** are also set as MQ message properties, e.g.

properties.user-id	set on MQ message as MQ propertyAMQPUserId
properties.to	set on MQ message as MQ propertyAMQPTo
properties.subject	set on MQ message as MQ propertyAMQPSubject
properties.reply-to	set on MQ message as MQ propertyAMQPReplyTo
properties.content-type	set on MQ message as MQ propertyAMQPContentType
properties.content-encoding	set on MQ message as MQ propertyAMQPContentEncoding
properties.creation-time	set on MQ message as MQ propertyAMQPCreationTime
properties.group-id	set on MQ message as MQ propertyAMQPGroupId
properties.message-id	set on MQ message as MQ propertyAMQPMessageId
Properties.group-sequence	set on MQ message as MQ propertyAMQPGroupSequence
Properties.absolute-expiry-time	set on MQ message as MQ propertyAMQPAbsoluteExpiryTime
Properties.reply-to-group-id	set on MQ message as MQ propertyAMQPReplyToGroupId

Finally, all AMQP **application-properties** are copied into the MQ message properties in the user space (usr.*) using similar naming conventions with some restrictions on the property length, characters used, and certain keyword restrictions e.g. “JMS”.

Detail on Interoperability – MQ to AMQP (Beta)

The following **MQMD fields** are set on the AMQP message as headers, **if and only if** the value in the MQ message is not the same as the AMQP default value for that property.

MQMD.persistence	set on AMQP message as header.durable
MQMD.expiry	set on AMQP message as header.ttl
MQMD.priority	set on AMQP message as header.priority

Some **MQ message properties**, if they exist, are set as AMQP headers:

MQ message property AMQPFirstAcquirer	set on AMQP message as header.first-acquirer
MQ message property AMQPDeliveryCount	set on AMQP message as header.delivery-count

Some **MQ message properties** are set as AMQP properties:

AMQPUserId	set on the AMQP message as properties.user-id
AMQPTo	set on the AMQP message as properties.to
AMQPSubject	set on the AMQP message as properties.subject
AMQPReplyTo	set on the AMQP message as properties.reply-to
AMQPContentType	set on the AMQP message as properties.content-type
AMQPContentEncoding	set on the AMQP message as properties.content-encoding
AMQPCreationTime	set on the AMQP message as properties.creation-time
AMQPGroupId	set on the AMQP message as properties.group-id
AMQPMessageId	set on the AMQP message as properties.message-id
AMQPGroupSequence	set on the AMQP message as properties.group-sequence
AMQPAbsoluteExpiryTime	set on the AMQP message as properties.absolute-expiry-time
AMQPReplyToGroupId	set on the AMQP message as properties.reply-to-group-id

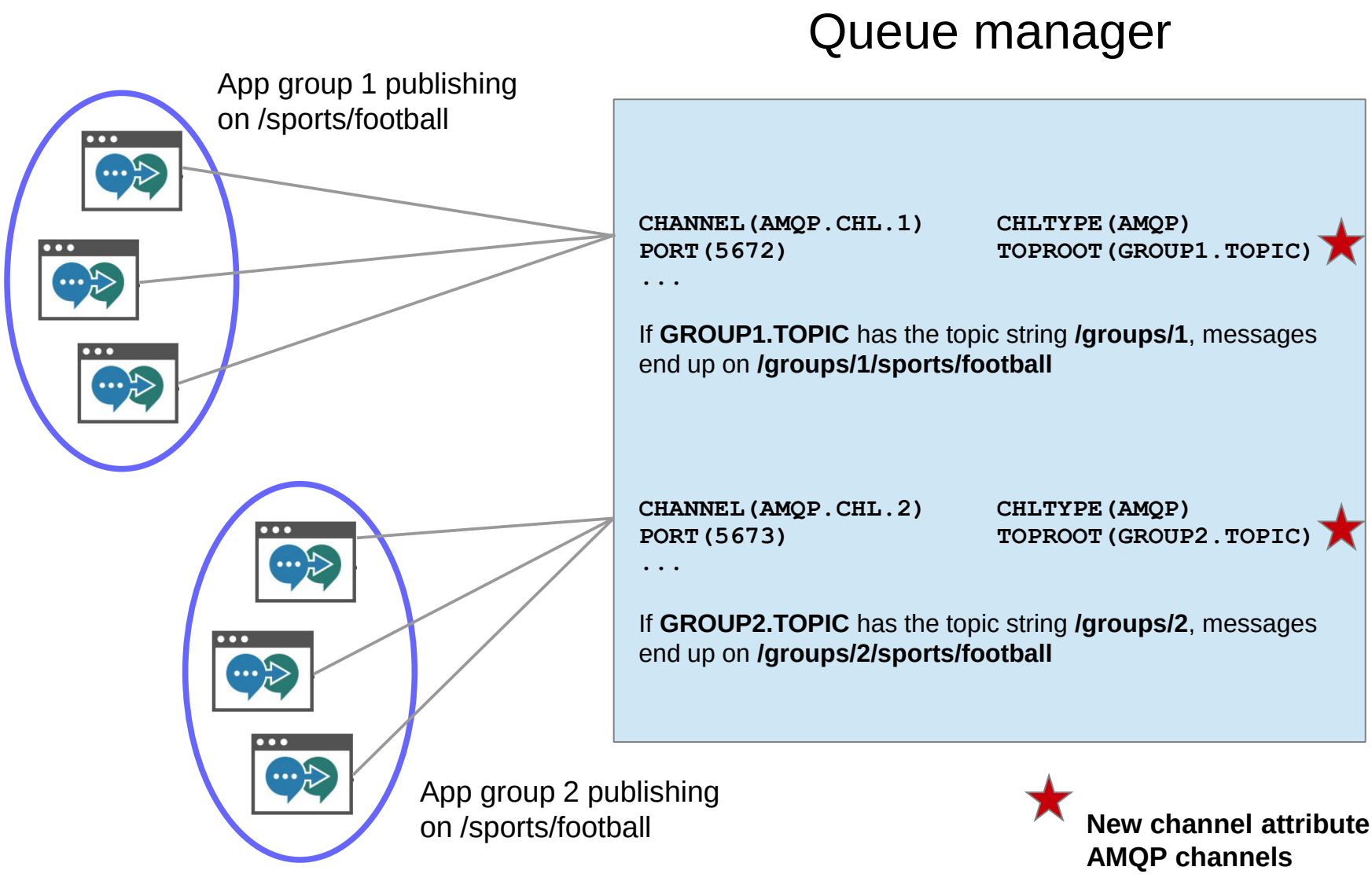
Finally, all **MQ message properties in the user space** (i.e. those which start usr.*) are copied into the AMQP message as application properties.

Development Backlog (Beta)

These features aren't yet available in the beta, but they're on our backlog

- Parity with MQ Distributed Platform coverage
- Connection Authentication Rules
- Segregating AMQP/MQ Light messages from other MQ applications **(see next slide)**
- Enhanced Monitoring
- KnowledgeCenter documentation
- Administering shared destinations

Managing MQ Light in an MQ Environment (Beta)



Demo

Agenda

- Quick recap of MQ Light
- How MQ Light and AMQP applications work with IBM MQ (beta)
- Hybrid Messaging with MQ and Bluemix

IBM Bluemix

Bluemix is an **open-standards**, cloud-based platform for **building**, **running, and managing applications.**

Build your apps, your way

Use the most prominent compute technologies to power your app: Cloud Foundry, Docker, OpenStack.

Scale more than just instances

Development, monitoring, deployment, and logging tools allow the developer to run and manage the entire application.

Extend apps with services

A catalog of IBM, third party, and open source services allow the developer to stitch an application together quickly.

Deploy and manage hybrid apps seamlessly

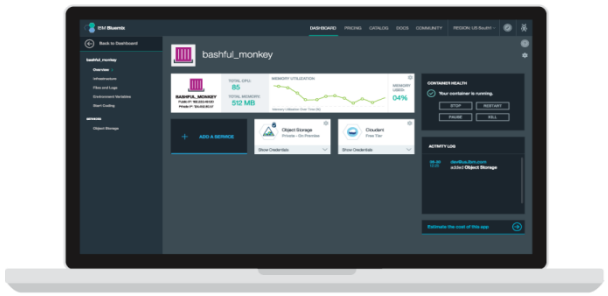
Get a seamless dev and management experience across a number of hybrid implementations options.

Layered Security

IBM secures the platform and infrastructure and provides you with the tools to secure your apps.

Flexible Pricing

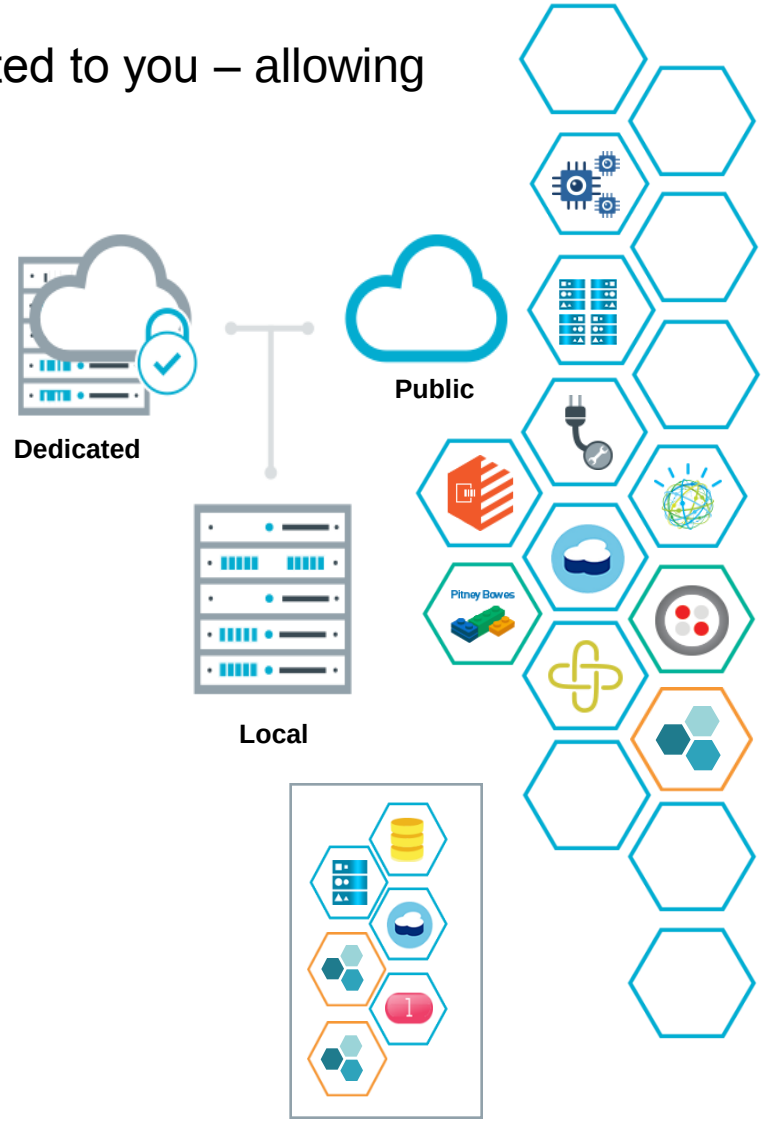
Try compute options and services for free and, when you're ready, pay only for what you use. Pay as you go and subscription models offer choice and flexibility.



Has a “Dedicated to You” option

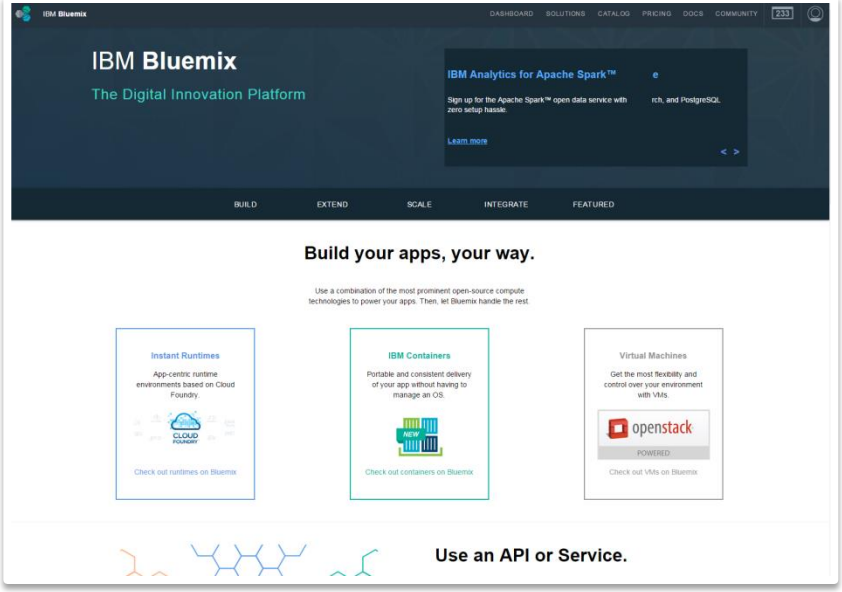
Single tenant hardware that’s completely dedicated to you – allowing you to satisfy regulatory & legal compliance.

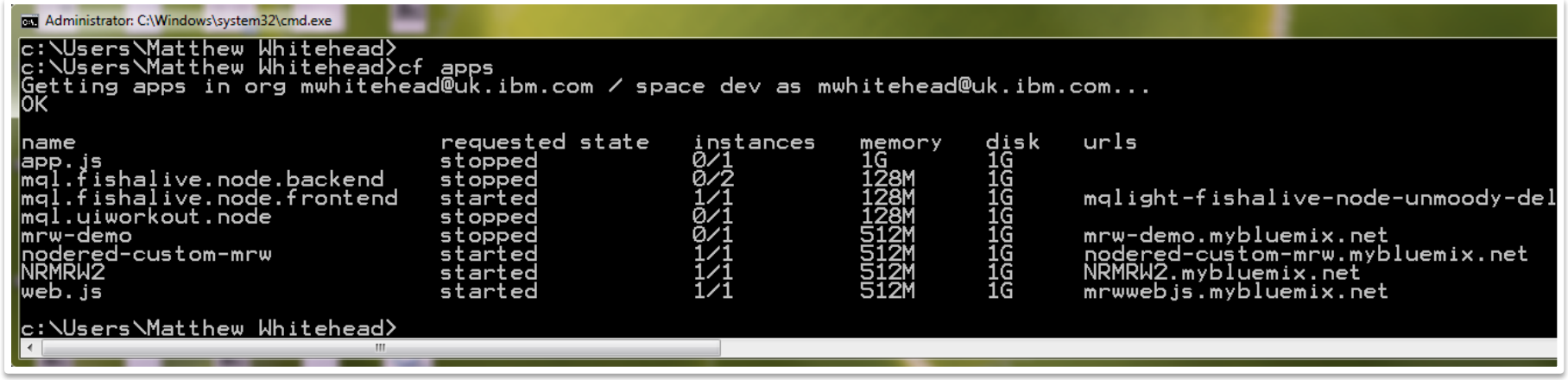
- The Bluemix platform and dedicated runtimes and services sit on SoftLayer hardware that is dedicated to you
- You still have the ability to connect to all multi-tenant services in the “public” catalog
- Integrated to your LDAP for developer authentication
- Elastic capacity based on your demands.



IBM Bluemix

- Based on Pivotal ® Cloud Foundry ®
- Uses standard cf commands, e.g.
    - cf push <myapp>
    - cf start <myapp>
    - cf stop <myapp>
    - cf apps
    - cf create-service …
    - cf bind-service …





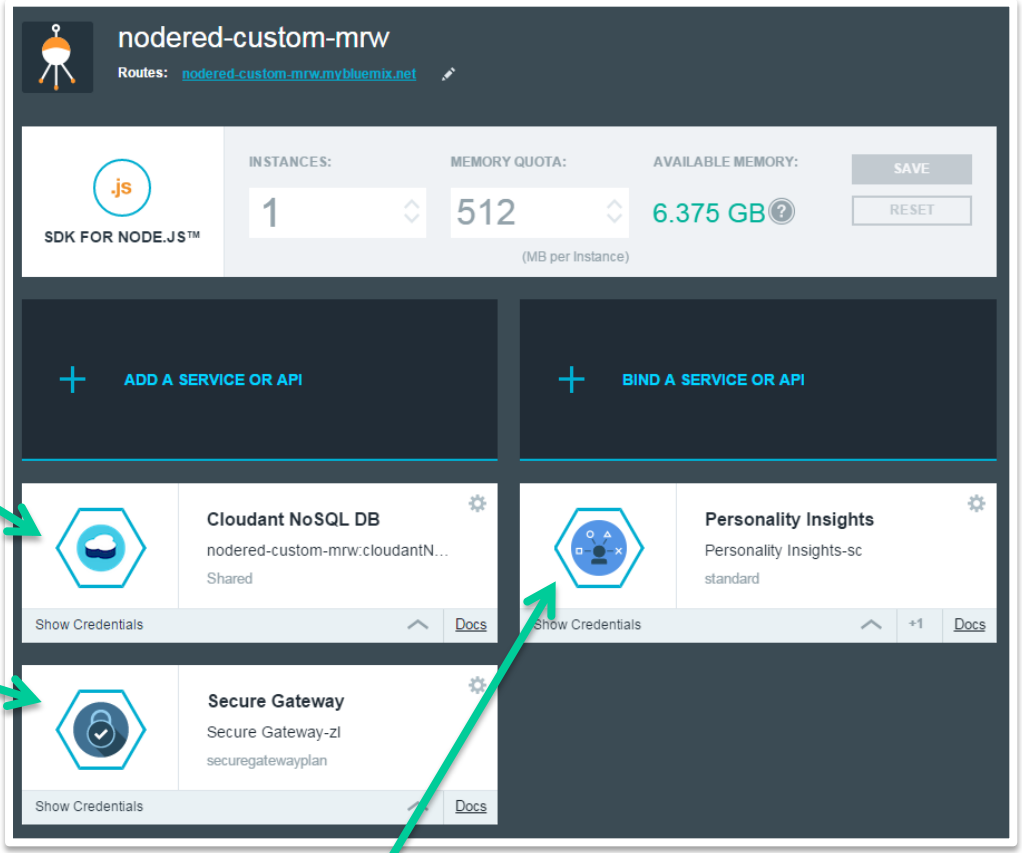
```
Administrator: C:\Windows\system32\cmd.exe
c:\Users\Matthew Whitehead>
c:\Users\Matthew Whitehead>cf apps
Getting apps in org mwhitehead@uk.ibm.com / space dev as mwhitehead@uk.ibm.com...
OK

name                          requested state   instances   memory   disk   urls
app.js                        stopped           0/1         1G       1G
mql.fishalive.node.backend    stopped           0/2         128M     1G
mql.fishalive.node.frontend   started           1/1         128M     1G     mqlight-fishalive-node-unmoody-del
mql.uiworkout.node            stopped           0/1         128M     1G
mrw-demo                      stopped           0/1         512M     1G     mrw-demo.mybluemix.net
nodered-custom-mrw            started           1/1         512M     1G     nodered-custom-mrw.mybluemix.net
NRMRW2                        started           1/1         512M     1G     NRMRW2.mybluemix.net
web.js                        started           1/1         512M     1G     mrwwebjs.mybluemix.net

c:\Users\Matthew Whitehead>
```

IBM Bluemix – Deployment Options

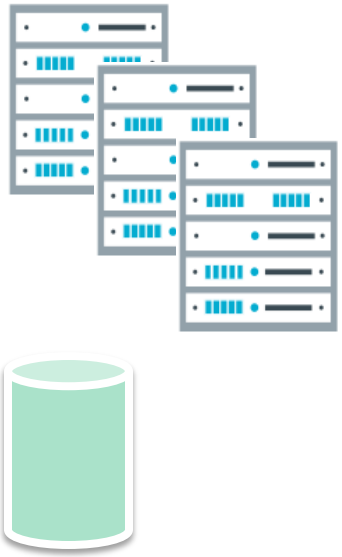
Applications are bound to the services they require

This is the dashboard for one of my applications. It's bound to several Bluemix services:

- A Cloudant NoSQL database service, for storing data
- A Bluemix Secure Gateway service, for Hybrid Messaging
- A Watson Personality Insights service, for text analysis

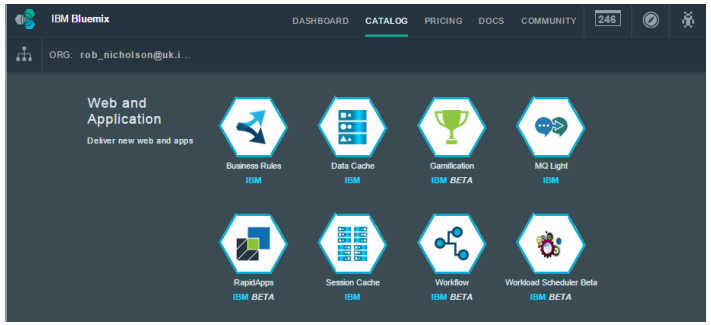


Hybrid Messaging – Joining the 2 worlds together



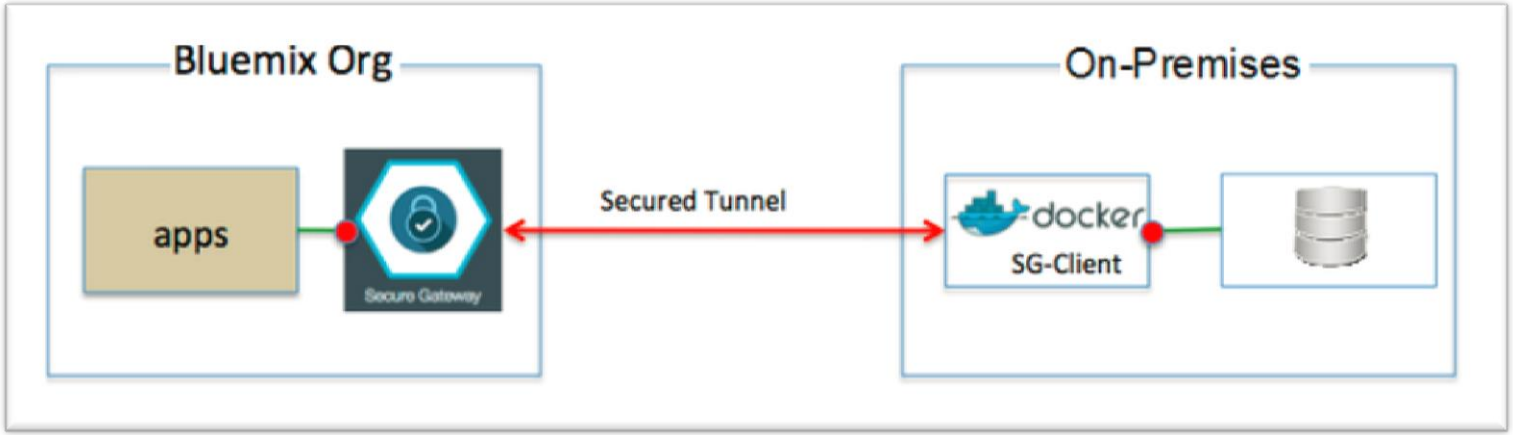
- Systems of record
- Enterprise data
- 24 x 7 x 365 applications

- Systems of engagement
- Mobile
- Social
- Analytics & Watson
- Rapid development



Hybrid Messaging

Connecting the cloud with the enterprise



- **Bluemix Secure Gateway** provides a tunnel between Bluemix and your enterprise
- Gives your Bluemix apps an IP and port that routes through to a server in the enterprise
- Requires Docker in the enterprise to act as the reverse-tunnel agent

Demo

Thank You - Questions?

Related sessions:

- Introduction to MQ Light & Bluemix
  - Tuesday 1.00pm
- Repeat of this session
  - Wednesday 9.50am

Please Note

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Trademark Statement

- IBM and the IBM logo are trademarks of International Business Machines Corporation, registered in many jurisdictions. Other marks may be trademarks or registered trademarks of their respective owners.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
- Red Hat Enterprise Linux is a registered trademark of Red Hat, Inc. in the United States and other countries.
- Ubuntu and Canonical are registered trademarks of Canonical Ltd.
- SUSE and SLES are registered trademarks of SUSE LLC in the United States and other countries
- Mac and OS X are trademarks of Apple Inc., registered in the U.S. and other countries
- Other company, product and service names may be trademarks, registered marks or service marks of their respective owners.
- References in this publication to IBM products and services do not imply that IBM intends to make them available in all countries in which IBM operates.