



SHARE Live!:

What's the cloud going to do to my MQ network?

Chris Leonard - ChrisL@uk.ibm.com

Mayur Raja - mayur_raja@uk.ibm.com

IBM UK

Session 17820

Tuesday 11th August 2015



SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

Copyright (c) 2015 by SHARE Inc. Except where otherwise noted, this work is licensed under <http://creativecommons.org/licenses/by-nc-sa/3.0/>



Legal Disclaimer



The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

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.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

© IBM Corporation 2015. All Rights Reserved.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

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.

Material Content

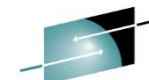
Mix of concepts and projects in design phase

This is a mix of topics that are conceptual in nature and projects in design phase. There are no plans in place at this point to deliver some of this capability. We simply want to discuss these, get your reaction and feedback in order to determine if/how to pursue.

We do not recommend basing any plans on this capability.

But, if you want to, you can create your own z/OSMF workflows to automate the provisioning and de-provisioning of resources.

MQ sessions this week



SHARE
Educate • Network • Influence

	Monday	Tuesday	Wednesday	Thursday	Friday
08:30			MQ for z/OS, Using and Abusing New Hardware and the New v8 Features	Nobody Uses Files Any More Do They? New Technologies for Old Technology, File Processing in MQ MFT and IIB	Monitoring and Auditing MQ Securing MQ Initiated CICS Workload
10:00	Introduction to MQ - Can MQ Really Make My Life Easier?	MQ for z/OS: The Insider Story	IBM Integration Bus MQ Flexibility	Common Problems and Problem Determination for MQ z/OS	IBM MQ and IBM Integration Bus - from Migration and Maintenance to Continuous Enhancements, How and Why to Stay Current
11:15	Introduction to IBM Integration Bus on z/OS	Introduction to the New MQ Appliance	MQ V8 Hands-on Labs! MQ V8 with CICS and COBOL! MQ SMF Labs!		
12:15					
1:45	What's New in the Messaging Family - MQ v8 and More		Getting Started with Performance of MQ on z/OS	IBM MQ: Are z/OS & Distributed Platforms Like Oil & Water?	
3:15	What's New in IBM Integration Bus	Live!: End to End Security of My Queue Manager on z/OS Application Programming with MQ Verbs	Digging into the MQ SMF Data	MQ Parallel Sysplex Exploitation, Getting the Best Availability from MQ on z/OS by Using Shared Queues	
4:30	MQ Security: New v8 Features Deep Dive	Live!: What's the Cloud Going to Do to My MQ Network? The Do's and Don'ts of IBM Integration Bus Performance	Giving It the Beans: Using IBM MQ as the Messaging Provider for JEE Applications in IBM WebSphere Application Server	Challenge the MQ & IIB Experts?	

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

IBM Hursley Park

Tues Aug 11th, 2015

SHARE
in Orlando **2015**



Agenda

- Does my traditional MQ network make sense in the cloud era?
- Client user growing pains
- Rapid development
- Scalability
- Provisioning MQ for z/OS

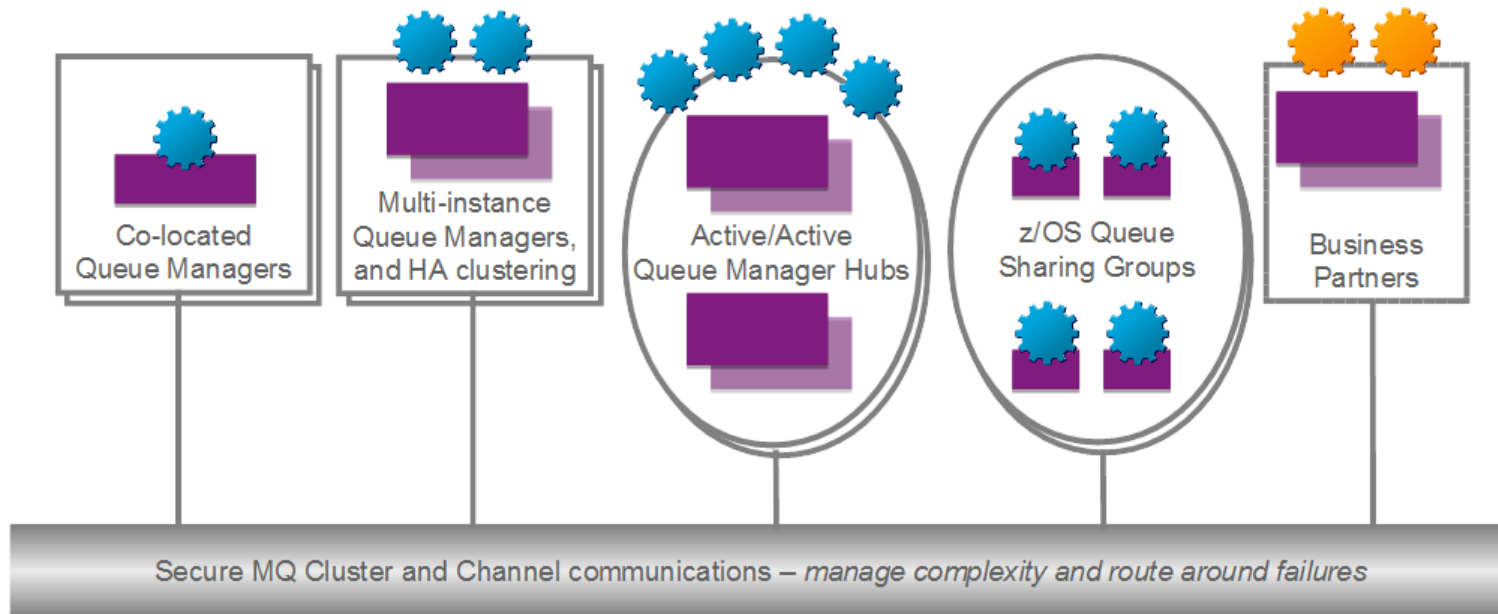
Does my traditional MQ infrastructure make sense in the cloud era?



The main challenges of the cloud era:

- Managing increasing numbers of concurrently connected users and applications, especially pervasive devices
- Rapid development and deployment to respond quickly to user requirements and maximise return on investment
- Provide dynamic capacity to manage demand effectively with efficient management of idle resources

Does my traditional MQ infrastructure make sense in the cloud era?



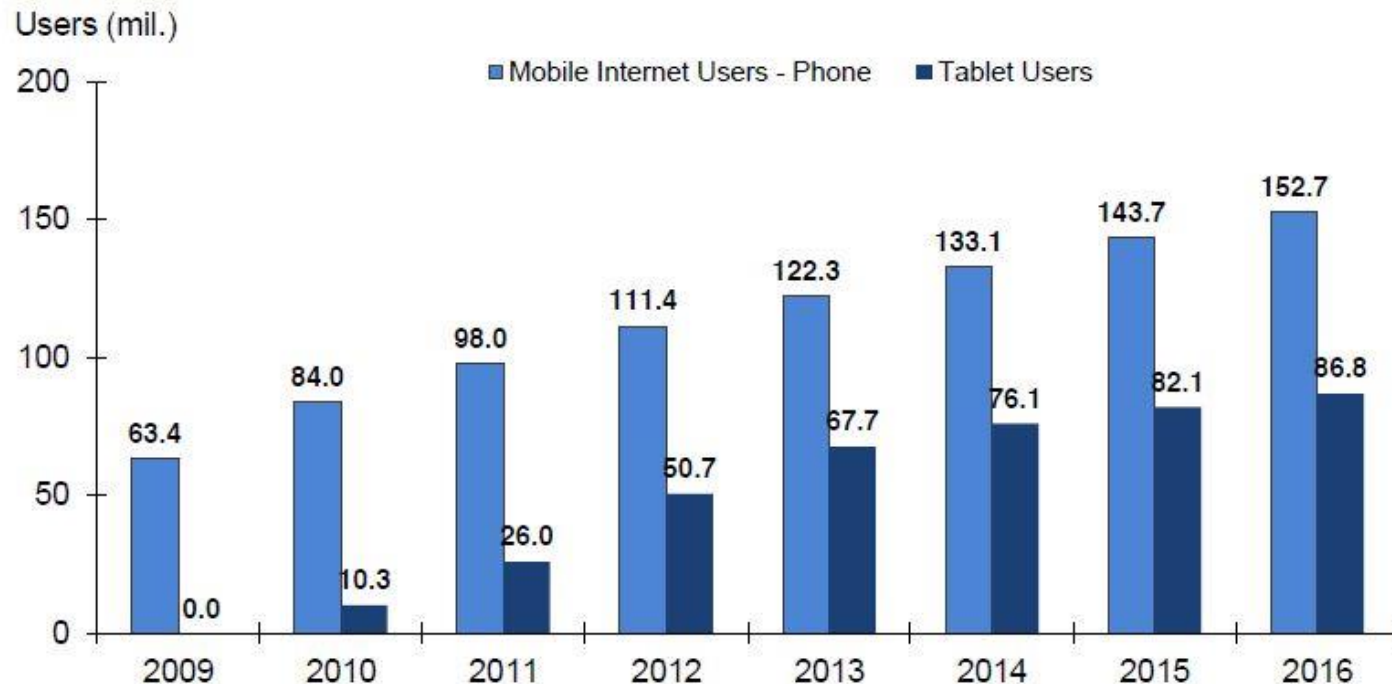
- Maybe.
 - MQ remains the market leader in enterprise messaging, designed to provide flexible, scalable solutions
 - Diverse platform and API coverage
 - Integration with other enterprise products such as application servers and databases

Agenda

- Does my traditional MQ network make sense in the cloud era?
- Client user growing pains
- Rapid development
- Scalability
- Provisioning MQ for z/OS

Client user growing pains

- Internet device adoption growth continues, for example in North America:

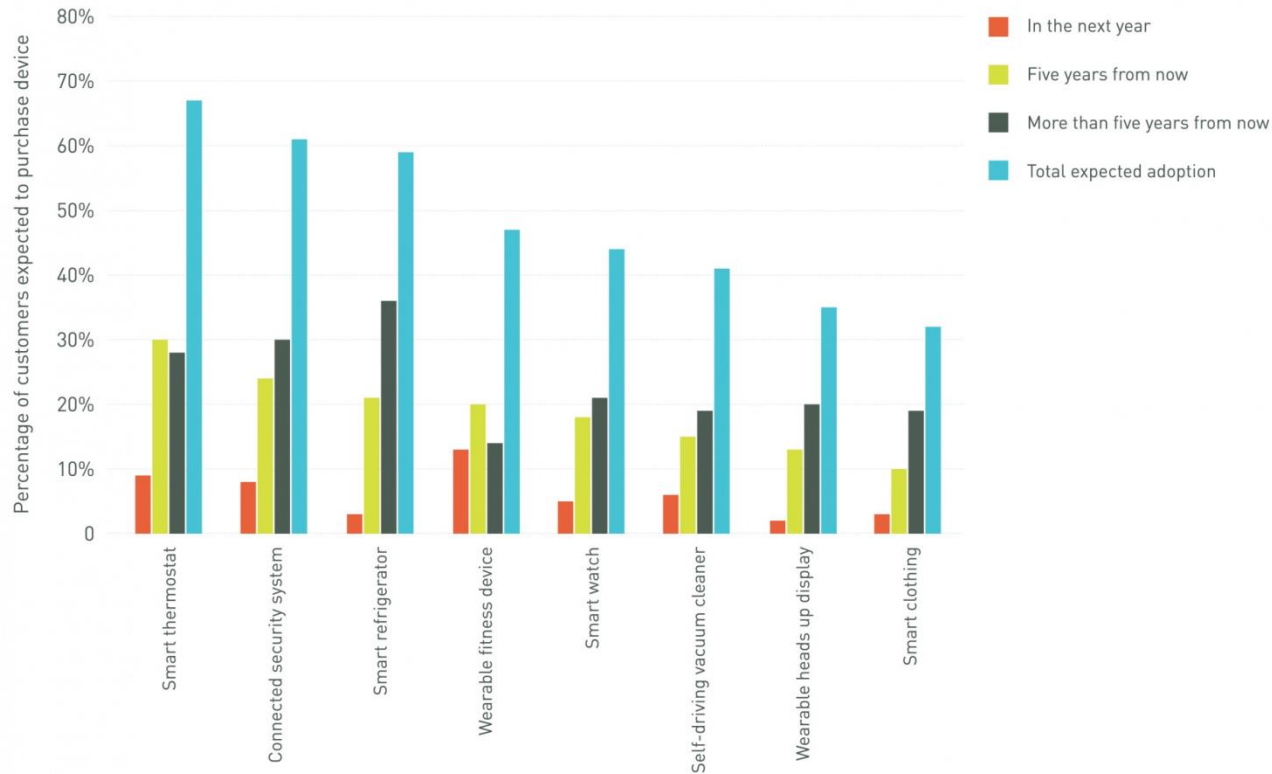


Source: Forrester Research Mobile Advertising Forecast, 2011-2016 (US)

Client user growing pains

- It's not just personal devices:

PROJECTED NEW ADOPTION OF CONNECTED TECHNOLOGY BY CONSUMERS



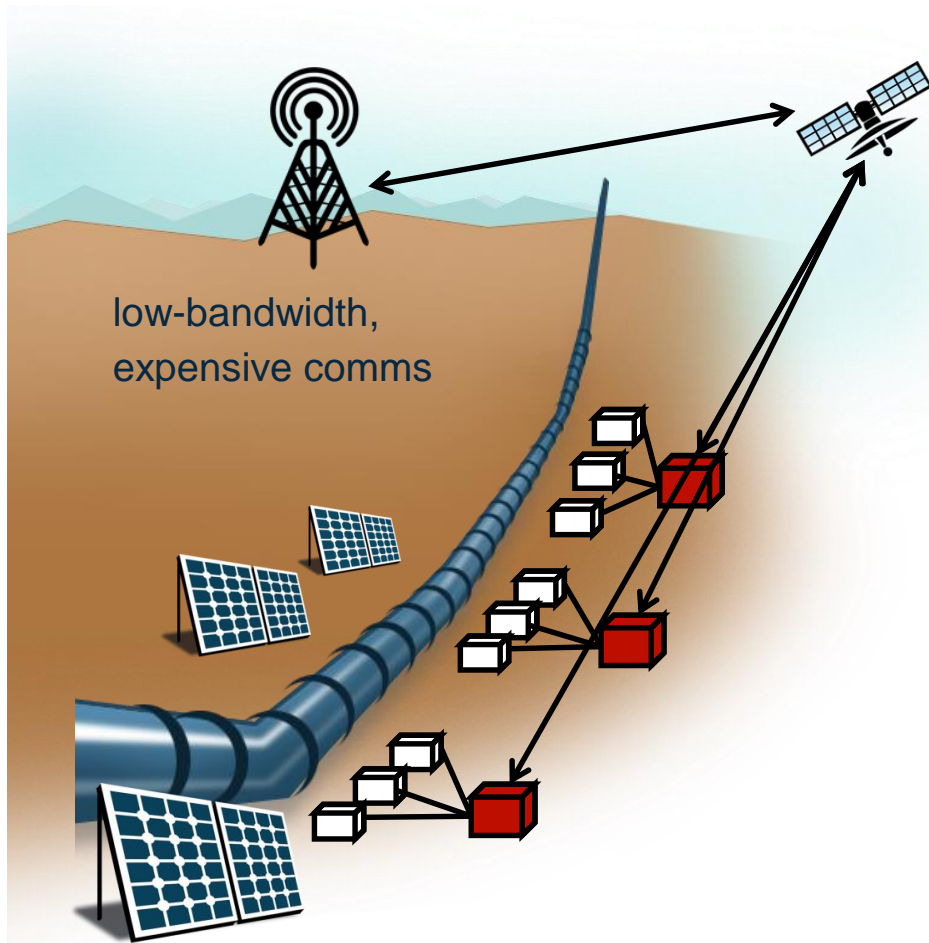
Source: Acuity Group 2014 Internet of Things Study via Forbes

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Lightweight messaging with MQTT

- Handling interconnected devices is not an unforeseen problem!
- To save inventing a new protocol every time a new embedded device came along, a common protocol was needed.
- MQ Telemetry Transport (MQTT) is that protocol.
 - It traces its roots back to 1999, where Dr Andy Stanford-Clark of IBM, and Arlen Nipper of Arcom (now Eurotech) devised the protocol.

Lightweight messaging with MQTT



Design goals of MQTT:

- Works over unreliable communication networks
- Minimal data overhead (low bandwidth)
- Capable of supporting large numbers of devices
- Simple to interface the data with the traditional IT world
- Simple to developers to write applications to use

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Lightweight messaging with MQTT

Key capabilities:

- Expects and caters for frequent network disruption – built for low bandwidth, high latency, unreliable, high cost networks
- Expects that client applications may have very limited resources available.
- Publish/subscribe messaging paradigm as required by the majority of SCADA and sensor applications.
- Provides traditional messaging qualities of service where the environment allows.
- OASIS standard for ease of adoption by device vendors and third-party client software.



MQTT header

- MQTT header can be as little as 2 bytes:

bit	7	6	5	4	3	2	1	0
Byte 1	Message Type				DUP flag	QoS Level		RETAIN
Byte 2	Remaining Length (at least one byte)							

- Contrast with WebSphere MQ MQMD header structure:

```

struct tag MQMD {
    MQCHAR4   StrucId;           // Structure identifier
    MQLONG    Version;          // Structure version number
    MQLONG    Report;           // Options for report messages
    MQLONG    MsgType;          // Message type
    MQLONG    Expiry;           // Message lifetime
    MQLONG    Feedback;         // Feedback or reason code
    MQLONG    Encoding;         // Numeric encoding of message data
    MQLONG    CodedCharSetId;   // Character set identifier of message data
    MQCHAR8   Format;           // Format name of message data
    MQLONG    Priority;          // Message priority
    MQLONG    Persistence;      // Message persistence
    MQBYTE24  MsgId;            // Message identifier
    MQBYTE24  CorrelId;         // Correlation identifier
    MQLONG    BackoutCount;     // Backout counter
    MQCHAR48  ReplyToQ;         // Name of reply queue
    MQCHAR48  ReplyToQMGr;      // Name of reply queue manager
    MQCHAR12  UserIdentifier;    // User identifier
    MQBYTE32  AccountingToken;  // Accounting token
    MQCHAR32  ApplIdentityData; // Application data relating to identity
    MQLONG    PutApplType;      // Type of application that put the message
    MQCHAR28  PutApplName;      // Name of application that put the message
    MQCHAR8   PutDate;          // Date when message was put
    MQCHAR8   PutTime;          // Time when message was put
    MQCHAR4   ApplOriginData;   // Application data relating to origin
    MQBYTE24  GroupId;          // Group identifier
    MQLONG    MsgSeqNumber;     // Sequence number of logical message within group
    MQLONG    Offset;           // Offset of data in physical message from start of logical message
    MQLONG    MsgFlags;         // Message flags
    MQLONG    OriginalLength;   // Length of original message }
    
```

MQTT header

Message Types:

CONNECT	CONNACK
PUBLISH	PUBACK
PUBREC	PUBREL
PUBCOMP	SUBSCRIBE
SUBACK	UNSUBSCRIBE
UNSUBACK	PINGREQ
PINGRESP	DISCONNECT

DUP flag:

Used to indicate a redelivery message for one of the message types:

PUBLISH, PUBREL, SUBSCRIBE, UNSUBSCRIBE

bit	7	6	5	4	3	2	1	0
Byte 1	Message Type				DUP flag	QoS Level		RETAIN
Byte 2	Remaining Length (at least one byte)							
Byte 3	Remaining Length (msg up to 16KB)							
Byte 4	Remaining Length (msg up to 2MB)							
Byte 5	Remaining Length (msg up to 256MB)							

Variable **length** message (127 bytes maximum for the single byte length field), up to a maximum of 256MB for 4 length byte fields.

Indicates if a message should be **retained**, to be sent to new subscribers.

MQTT qualities of service

- QoS 0: At most once delivery (non-persistent)
 - No retry semantics are defined in the protocol.
 - The message arrives either once or not at all.
- QoS 1: At least once delivery (persistent, duplicate messages possible)
 - Client sends message with Message ID in the message header
 - Server acknowledges with a PUBACK control message
 - Message resent with a DUP bit set If the PUBACK message is not seen
- QoS 2: Exactly once delivery (persistent)
 - Uses additional flows to ensure that message is not duplicated
 - Server acknowledges with a PUBREC control message
 - Client releases message with a PUBREL control message
 - Server acknowledges completion with a PUBCOMP control message

QoS 2

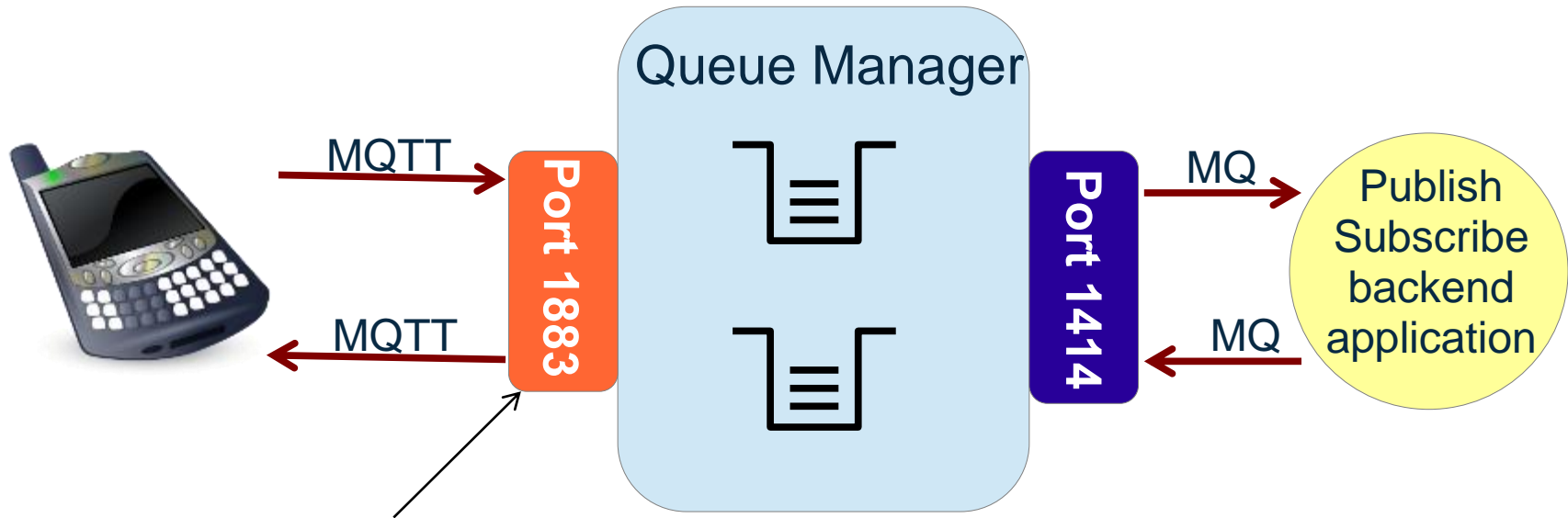
PubRel

PubComp

MQ Telemetry service (MQXR)

- Supplied as a component of MQ V7.1 and later releases on distributed platforms, under the component name “MQ Extended Reach” (or MQXR).
- MQXR brings MQTT protocol functionality to MQ
 - Highly scalable : tested with 200,000+ clients
 - Security : SSL channels, JAAS authentication, WMQ OAM
 - Ships with reference Java and C clients
 - Small footprint clients
 - Supports other APIs and implementations of MQTT clients available via 3rd parties

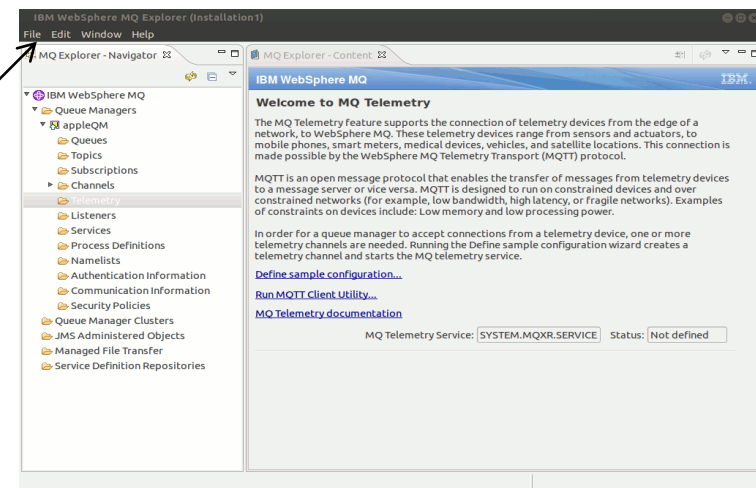
MQ Telemetry Service (MQXR)



WebSphere MQ MQTT Listener
IANA registered ports:
1883, 8883 for MQTT over SSH

Use WebSphere MQ Explorer to administer the WebSphere MQ Telemetry service – define Channels, start and stop the MQTT service.
Alternatively, it can be configured through 'runmqsc' commands.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



MQTT through Javascript

As of MQ 7.5.0.1, the MQXR component has support for MQTT v3.1 protocol over WebSockets.

This enables the use of MQTT through a WebSocket supporting web browser, meaning that MQTT can be used without preinstalling any software on a browser equipped device.

```
<script type="text/javascript" src="mqttws31.js"></script>
<script type="text/javascript">
  var clientId = "MyUniqueClientID";
  var client;
```

Reference the WMQ supplied MQTT javascript file

```
function publishMessage() {
  client = new Messaging.Client(location.hostname, Number(1883), clientId);
  client.onConnectionLost = onConnectionLost;
  client.onMessageArrived = onMessageArrived;
  client.connect({onSuccess:onConnect});
}
```

Connect to the MQTT server, and register callback functions

```
function onConnect() {
  // Once a connection has been made, make a subscription and send a message.
  console.log("onConnect");
  client.subscribe("/TopicLocation");
  message = new Messaging.Message("My publish text!");
  message.destinationName = "/TopicLocation";
  client.send(message);
}
```

Subscribe to the Topic, and publish a message.

Write callback functions here

```
</script>
<button type="button" onclick="publishMessage()" name="Connect">Publish a Message</button>
```

Invoke the Javascript function from HTML

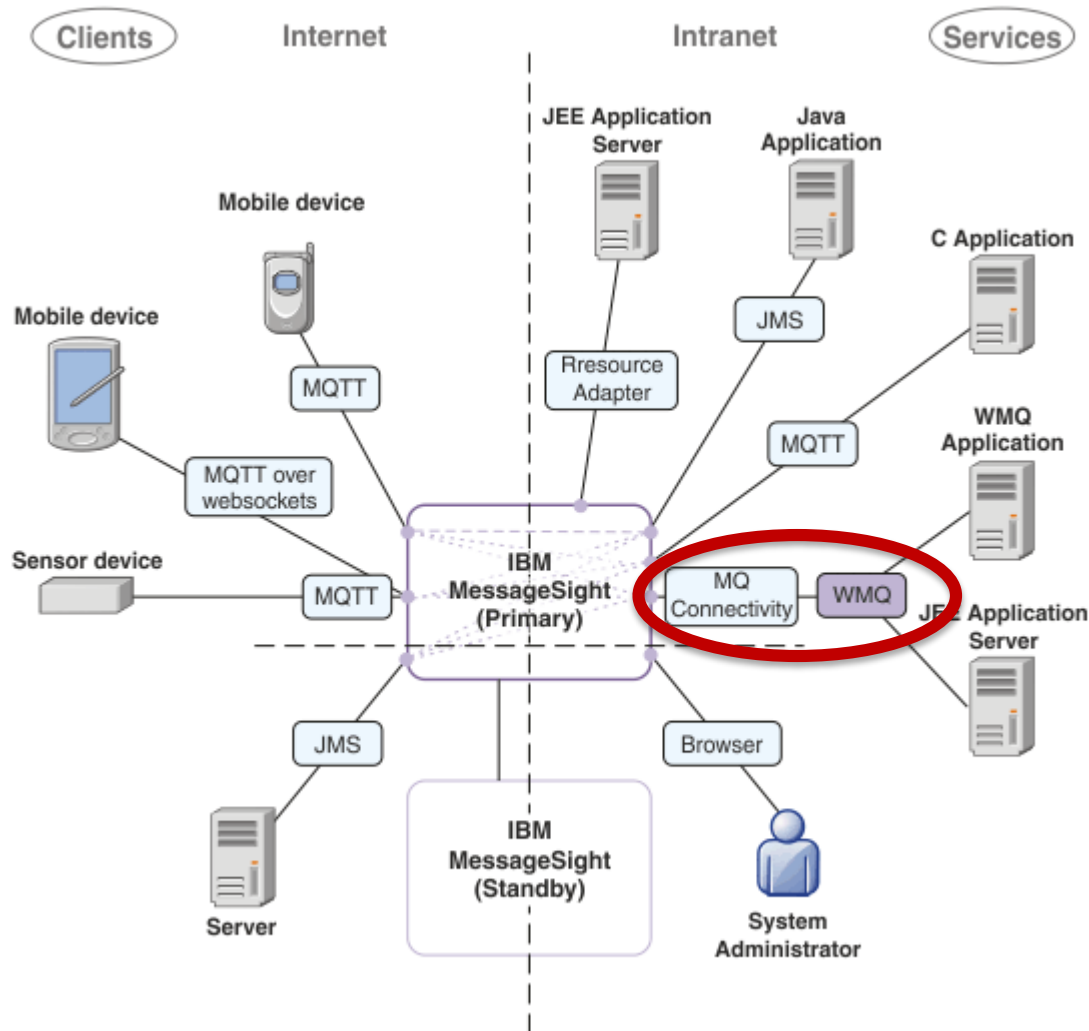
IBM MessageSight – big connectivity in a box



- A secure messaging server appliance optimised to meet the demands of massive scale messaging of machine-to-machine and mobile use cases.
- How massive? One appliance can achieve:
 - 1 million concurrent connections
 - 13 million non-persistent msg/sec
 - 400K persistent msg/sec

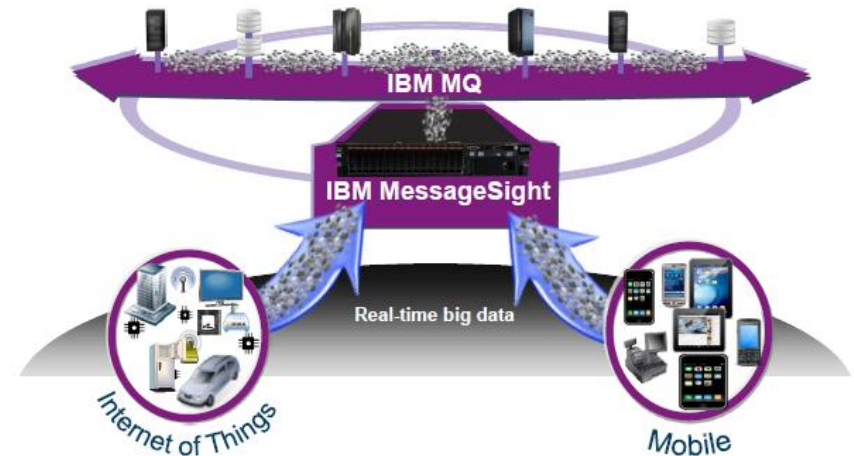
Complete your session evaluations online at www.SHARE.org/Orlando-Eval

IBM MessageSight and MQ



IBM MessageSight and MQ

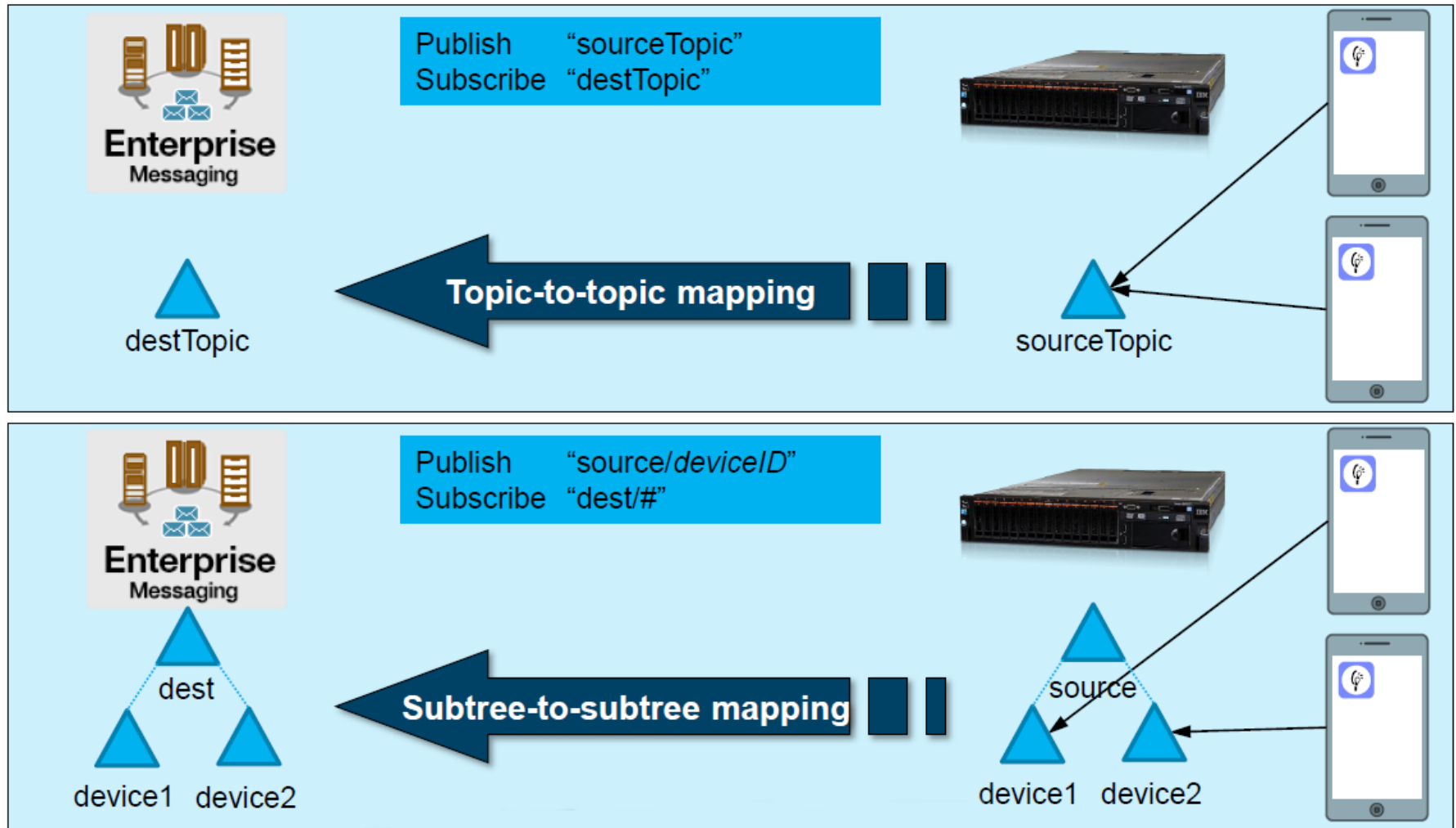
- Built-in MQ connectivity
- Securely extends existing enterprise messaging infrastructures
- Accelerate massive fan-out message delivery to huge numbers of devices
- Reliable bi-directional messaging enabling intelligent decisions based on real-time events



IBM MessageSight and MQ

- Very simple to set up
- Queue manager connection
 - Defines how to connect to a queue manager
 - Queue manager name
 - Connection name – host name, port
 - Channel name – server-connection channel
 - SSL cipher specification (optional)
- Destination mapping rule
 - Defines source and target of messages
 - Rule type – topic-to-topic, topic-to-queue, ...
 - Queue manager connection – one or more
 - Maximum messages to buffer for transmission
 - Retained messages?

IBM MessageSight and MQ



MQTT links

- MQTT homepage:
 - <http://mqtt.org>
- MQTT Specification
 - <http://www.ibm.com/developerworks/webservices/library/ws-mqtt/index.html>
- WebSphere MQ and MQ Telemetry
 - <http://www-01.ibm.com/software/integration/wmq/>
- Mobile Messaging & M2M Client Pack
 - <http://www.ibm.com/developerworks/mydeveloperworks/blogs/c565c720-fe84-4f63-873f-607d87787327/entry/download>
- MQTT: the Smarter Planet Protocol
 - <http://andypiper.co.uk/2010/08/05/mqtt-the-smarter-planet-protocol/>
- Lotus Expeditor (MQTT microbroker)
 - <http://www.ibm.com/software/lotus/products/expeditor/>
- IBM MessageSight
 - www.ibm.com/software/products/gb/en/messagesight/

Agenda

- Does my traditional MQ network make sense in the cloud era?
- Client user growing pains
- **Rapid development**
- Scalability
- Provisioning MQ for z/OS

Rapid development of messaging applications



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

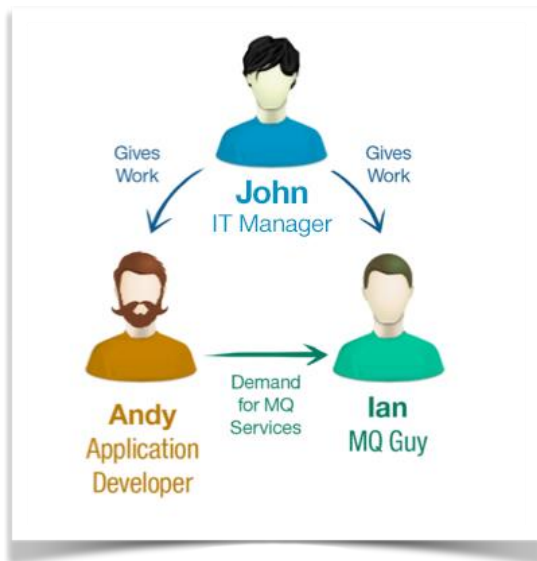
Meet Andy – Messaging application developer

- Wants to product applications that can be field tested in the minimum time possible
- Discovers technologies that are prevalent in his communities
- Uses the best tool for the job
- Intolerant of process / imposed technologies that do not obviously and immediately benefit his application



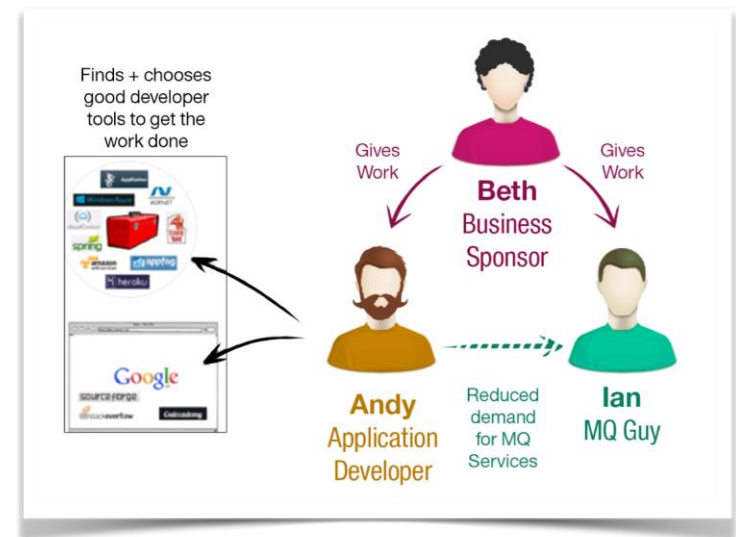
Organisational changes

From:
Centrally planned IT Architecture



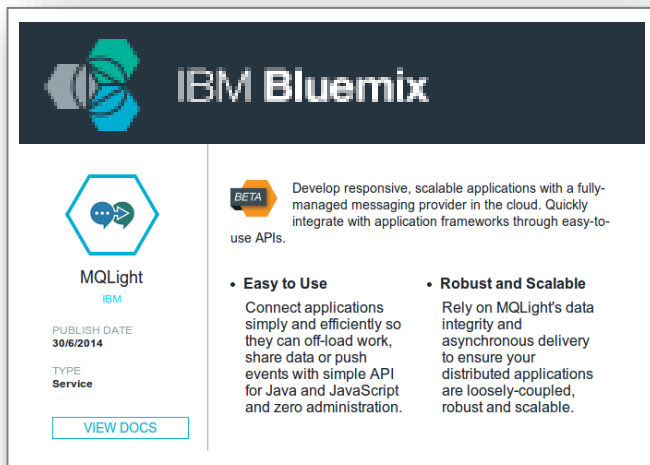
Centrally controlled common standards
Planned projects delivering core systems
Focused on skills and investment reuse

Emerges:
Business sponsor driven
Developer led architecture



Developers select tools to get the job done
Focused on trying new apps and concepts
in the market

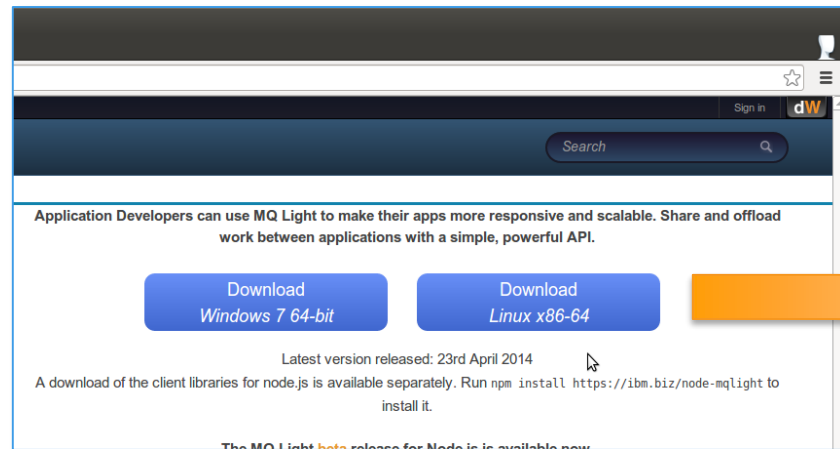
MQ Light : Software and Cloud



- Messaging that application developers will love to use, helping them make responsive applications that scale easily
- 3 ways to get it:
 - MQ Light software download
 - Bluemix service
 - *Statement of Direction for support in MQ Version 8.*
- Open APIs crafted to feel natural in a growing range of popular languages
- Tooling that makes modular app development easy

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

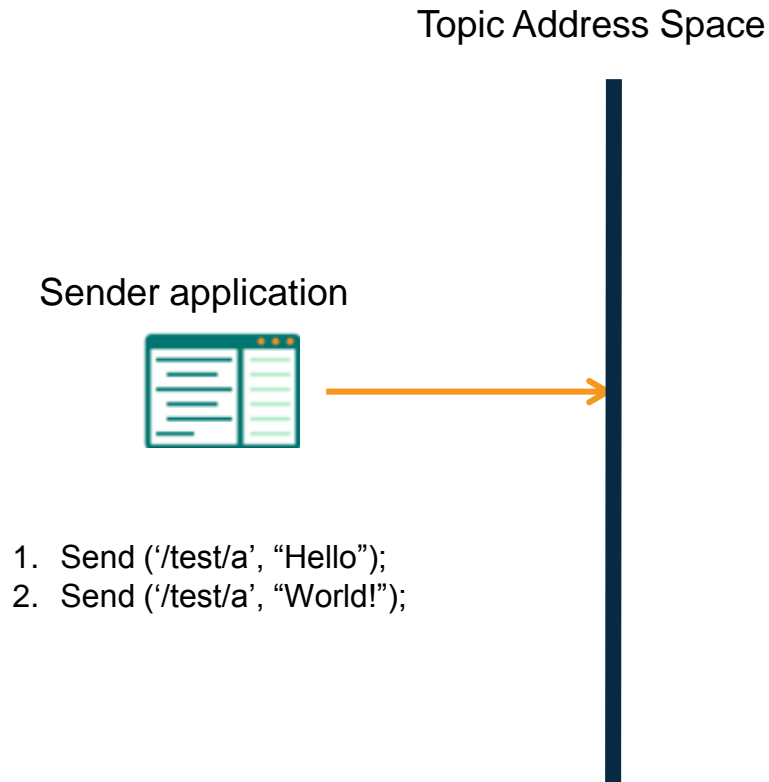
Software download - First five minute experience



- Download and get coding within 5 minutes
 - Linux-x86-64, Windows7 64 bit, Mac OSX
 - Unzip install
 - Unlimited time developer license (unsupported).
- No administration; just code and go
- Node API client libraries installed using npm package manager
- Tutorials and examples in multiple languages, relevant to actual use

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

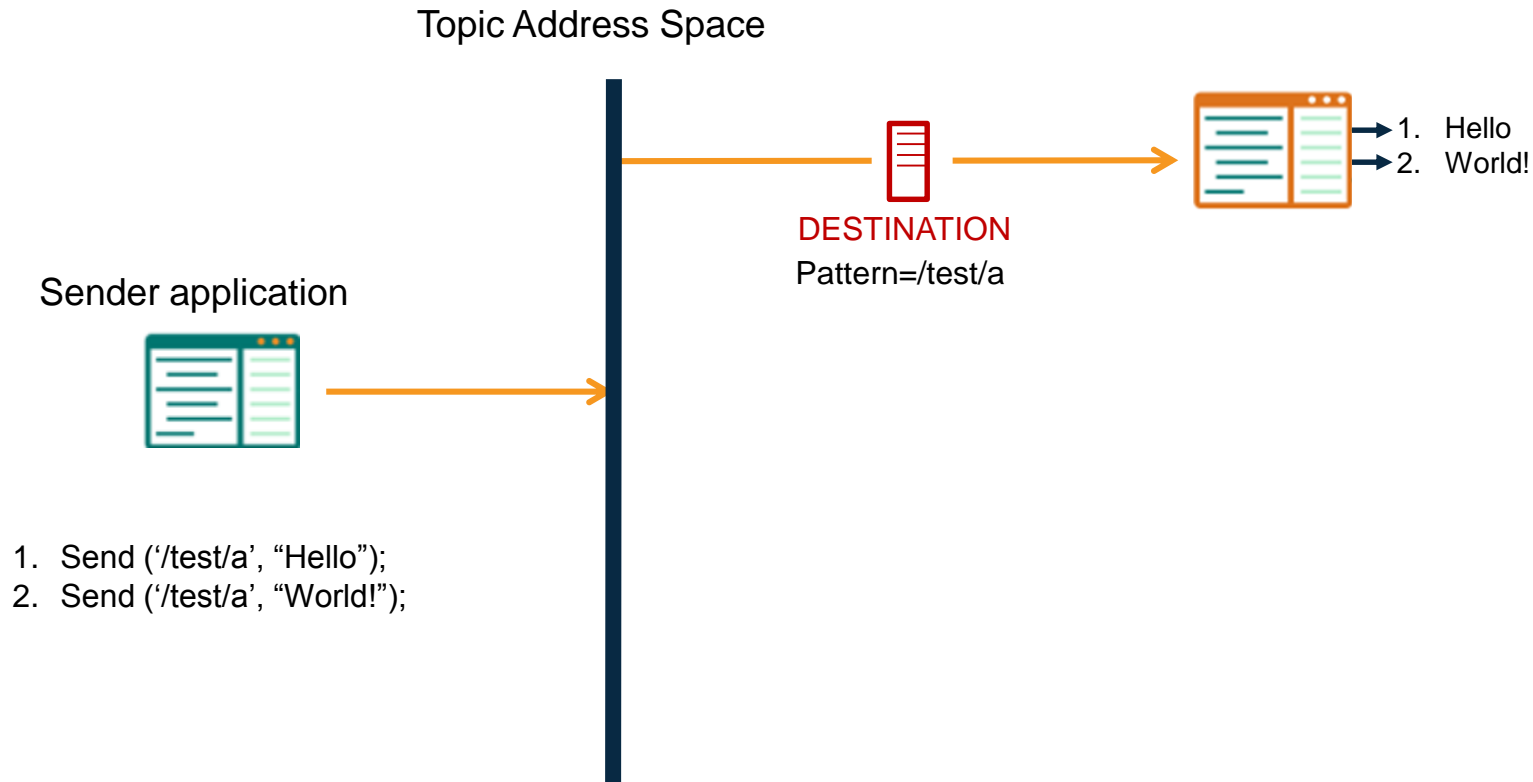
MQ Light Messaging Model – Send Messages



- Applications send messages to a topic.
- A topic is an address in the topic space
 - Either flat or arranged hierarchically.

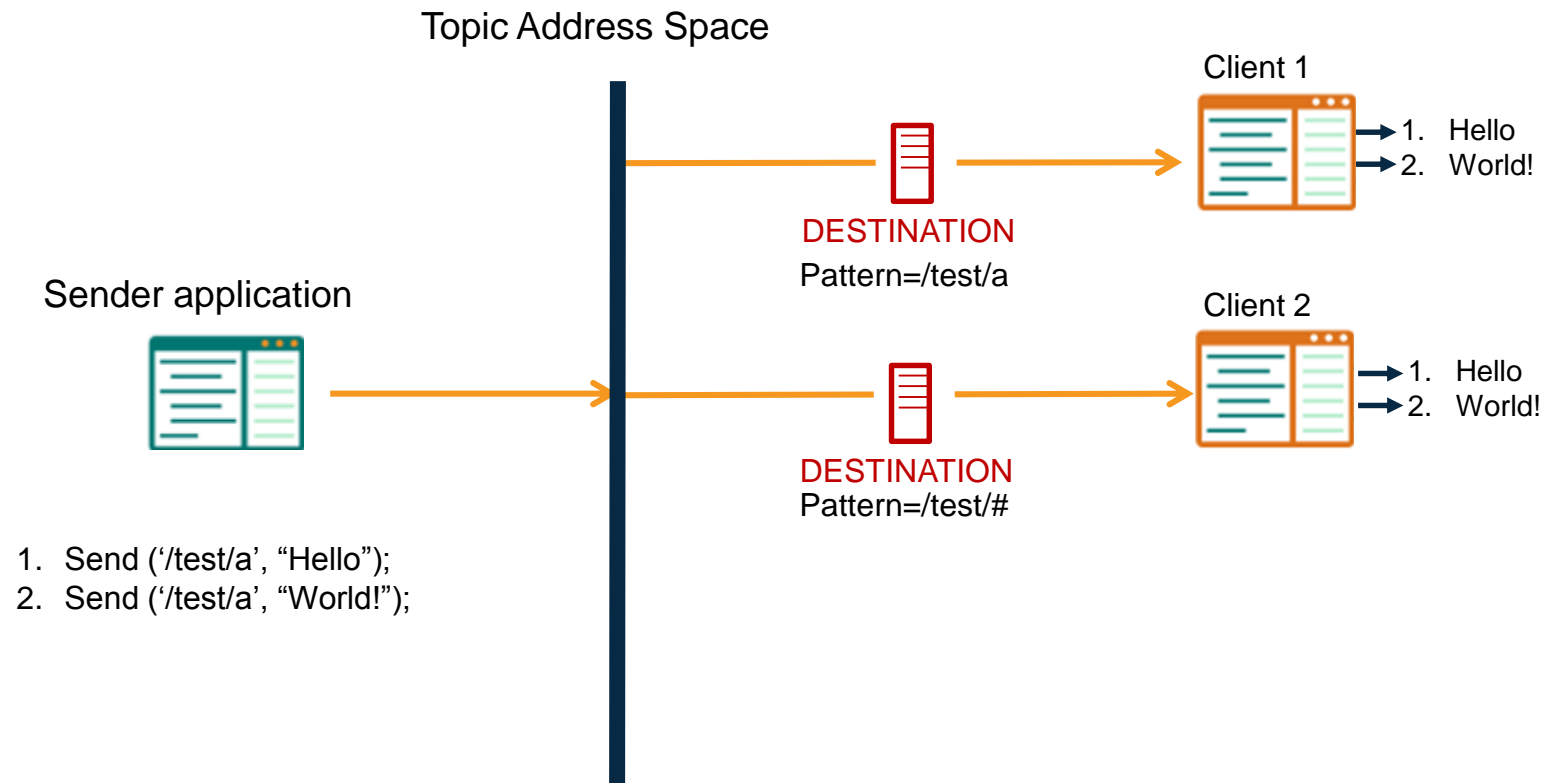
Complete your session evaluations online at www.SHARE.org/Orlando-Eval

MQ Light Messaging Model – Simple Receive



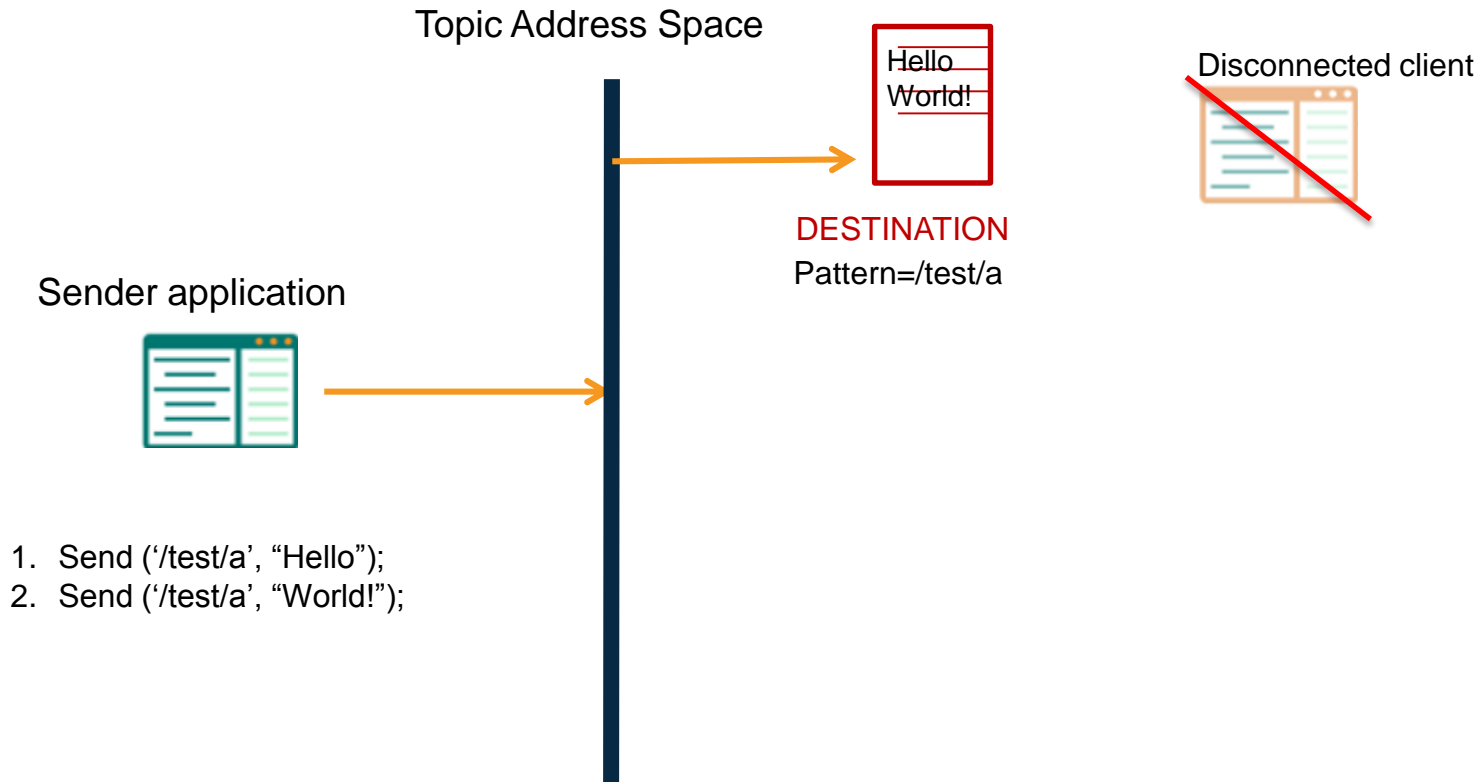
- Applications receive messages by creating a destination with a pattern which matches the topics they are interested in.
- Pattern matching scheme based on MQ.

MQ Light Messaging Model – Pub/Sub



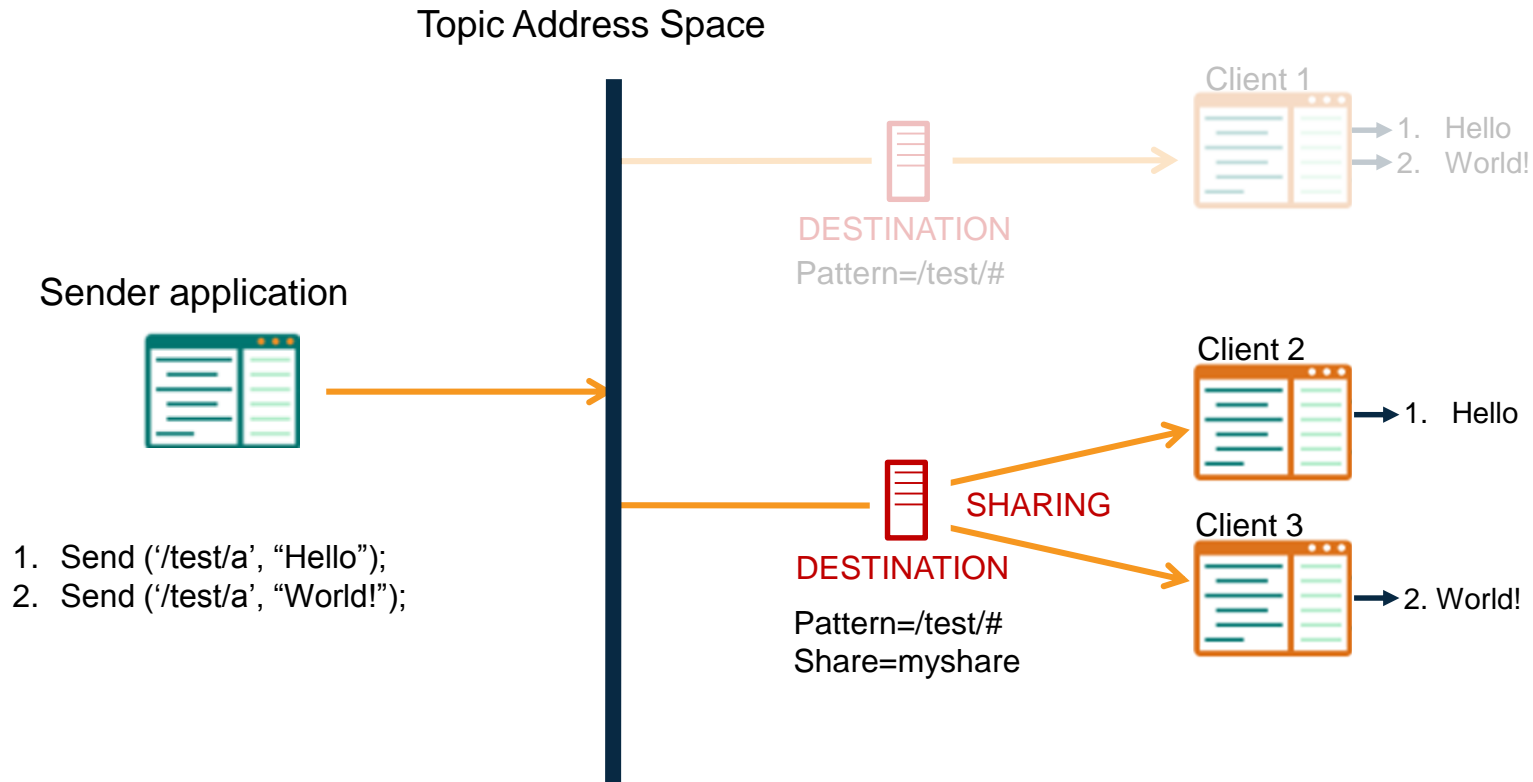
- Multiple destinations can be created which match the same topic
 - Pub/Sub style.

MQ Light Messaging Model – Persistent destinations



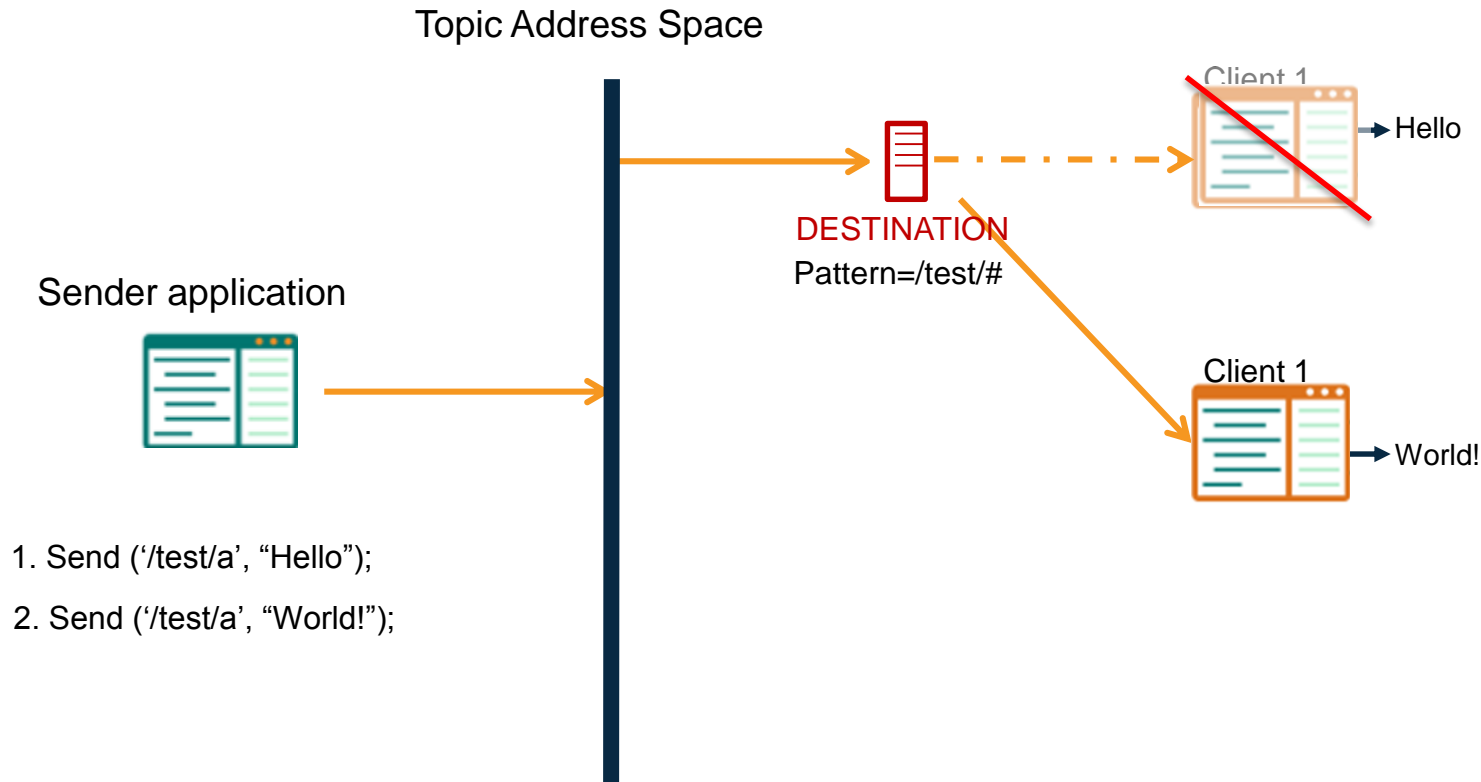
- Destinations persist for a defined “time to live” after receiver detaches.

MQ Light Messaging Model – Sharing



- Clients attaching to the same topic pattern and share name attach to the same shared destination.

MQ Light Messaging Model – Client takeover



- Applications connect to MQ Light service specifying (optional) client ID.
- Re-using the same client ID takes over the original connection.
 - Ideal for worker takeover in the cloud.

MQ Light Messaging Model

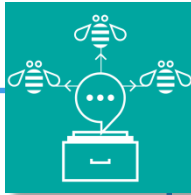
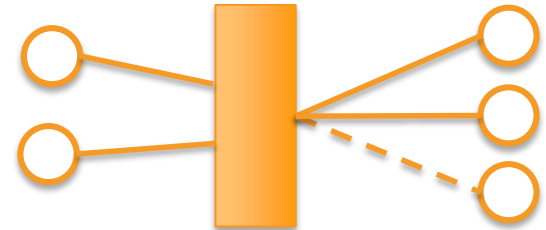
- Messages
 - Payload is either Text or Binary.
 - Content-type is used by clients to transfer JSON
 - Per message time to live.
- Message delivery model
 - At most once delivery (QoS 0)
 - At least once delivery (QoS 1)
 - Acknowledge & Reject messages
 - Control over the number of unacknowledged messages delivered. (link credit)

Use Cases

Worker Offload

Intensive work offloaded and distributed amongst worker processes to be performed asynchronously

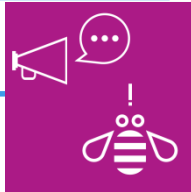
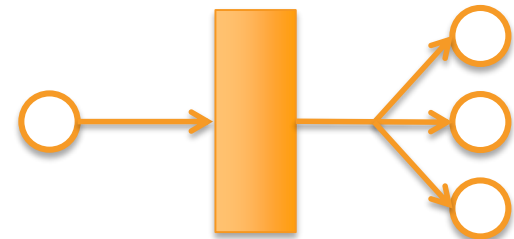
- Processing images or videos
- Performing text analytics



Event Driven

Take one or more actions when something interesting happens

- Email logs and update dashboards when build finishes
- Upload videos once finished transcoding

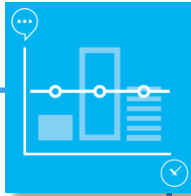
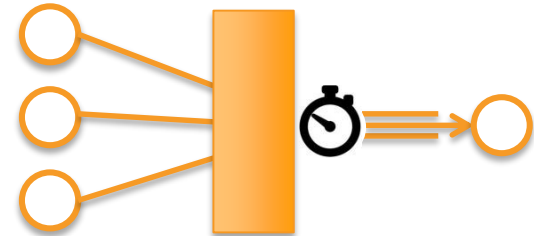


Use Cases

Delayed Processing

Schedule a task to happen at a specific point in time

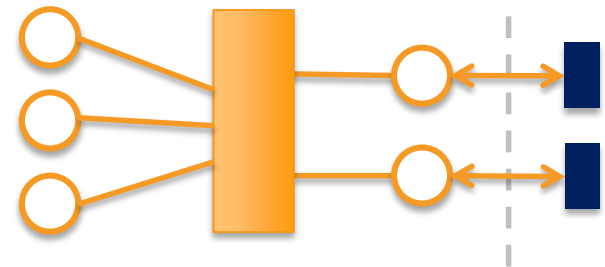
- Run in detailed reports when app use is low
- Generate end of day summary



3rd Party Integration

Ensure applications remain responsive even when 3rd party system are not available or responding fast enough

- Updating existing CRM system
- Booking appointment



MQ Light - WebUI

IBM MQ Light

[View Messages](#)[Documentation](#)

Clients: ● 2 connected ■ 3 disconnected

Since last clear history: 2 mins



Sending

Sent messages: 3

■ sender_1 1

[Topic List](#)

■ sender_2 1

[Topic List](#)

■ sender_3 1

[Topic List](#)

Messages

All messages

Topic pattern: Any



<1 min test message 3

1 / 1

[Details](#)

Sender: sender_3
Topic: sample/alternate/topic
TTL: 1 week

Receivers:
1 successful, 0 pending

☒ receiver_2

<1 min test message 2

1 / 1

[Details](#)

<1 min test message 1

1 / 1

[Details](#)

Receiving

Received messages: 3

● receiver_1

Pattern: sample/topic

1 2

● receiver_2

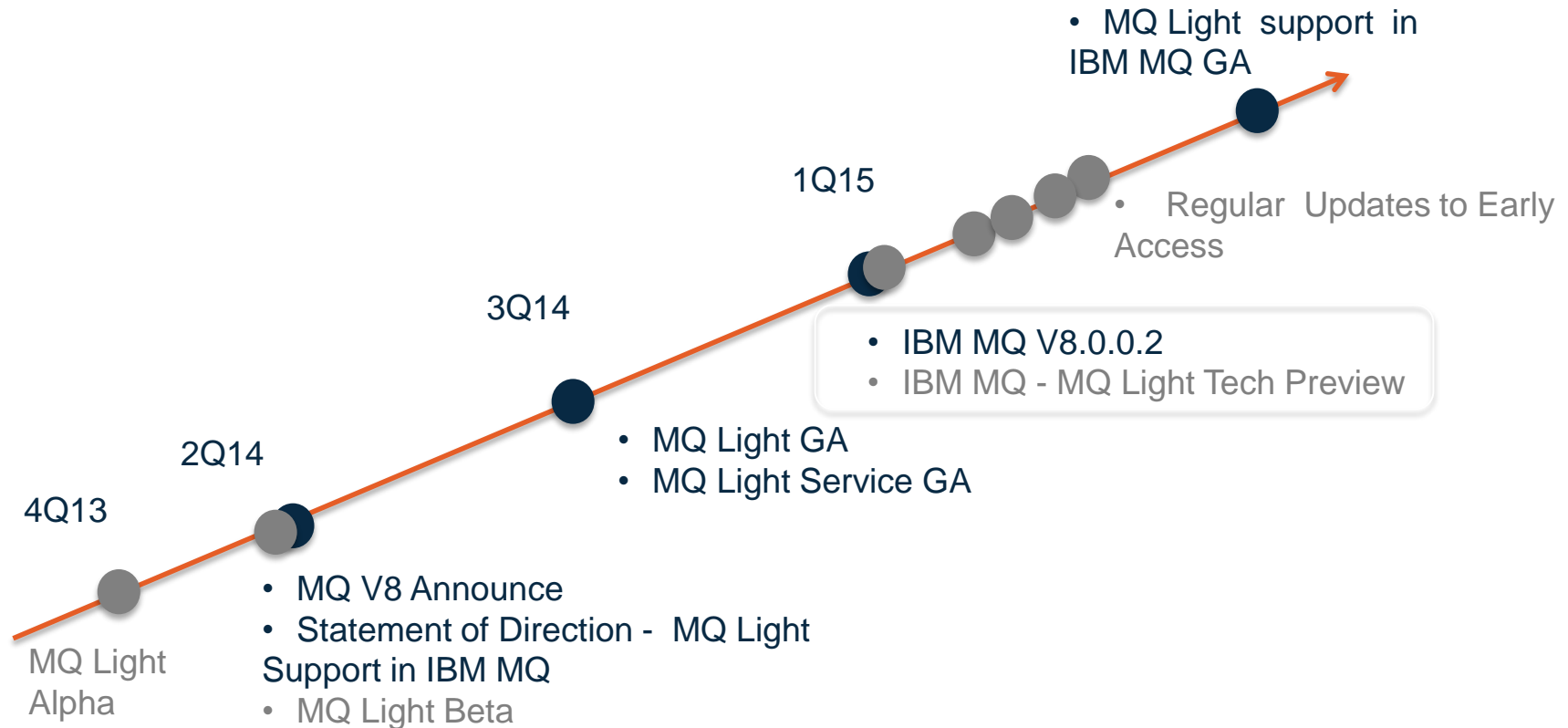
Pattern: sample/alternate/
topic

1 1

Agenda

- Does my traditional MQ network make sense in the cloud era?
- Client user growing pains
- Rapid development
- **Scalability**
- Provisioning MQ for z/OS

MQ Light Support in IBM MQ



IBM MQ – MQ Light Tech Preview

★ Open Beta - Available now ★



- Platforms

- Windows 64 Bit
- Linux x86_64

- Beta Installation

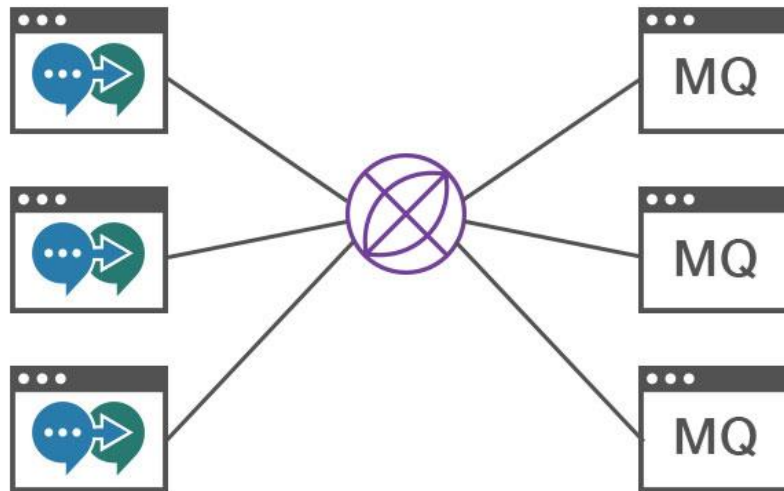
- Prereq is IBM MQ v8.0.0.2
- Add Tech Preview install media
 - Linux – RPM which is installed along side the other MQ RPMs
 - Windows – Zip which is manually extracted to an MQ installation

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



New AMQP channel type

- Adds a channel type of “AMQP”
- Support a subset of the AMQP 1.0 Oasis specification
- Interoperable with MQ FAP and MQTT applications



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

AMQP channels

★ Open Beta - Available now ★



```
mwhitehead@ubuntu-vm: /opt/mqmAMQP
:
:
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)
```

DIS CONN command

Note the new client ID attribute set on the MQ connection

AMQP channels

★ Open Beta - Available now ★



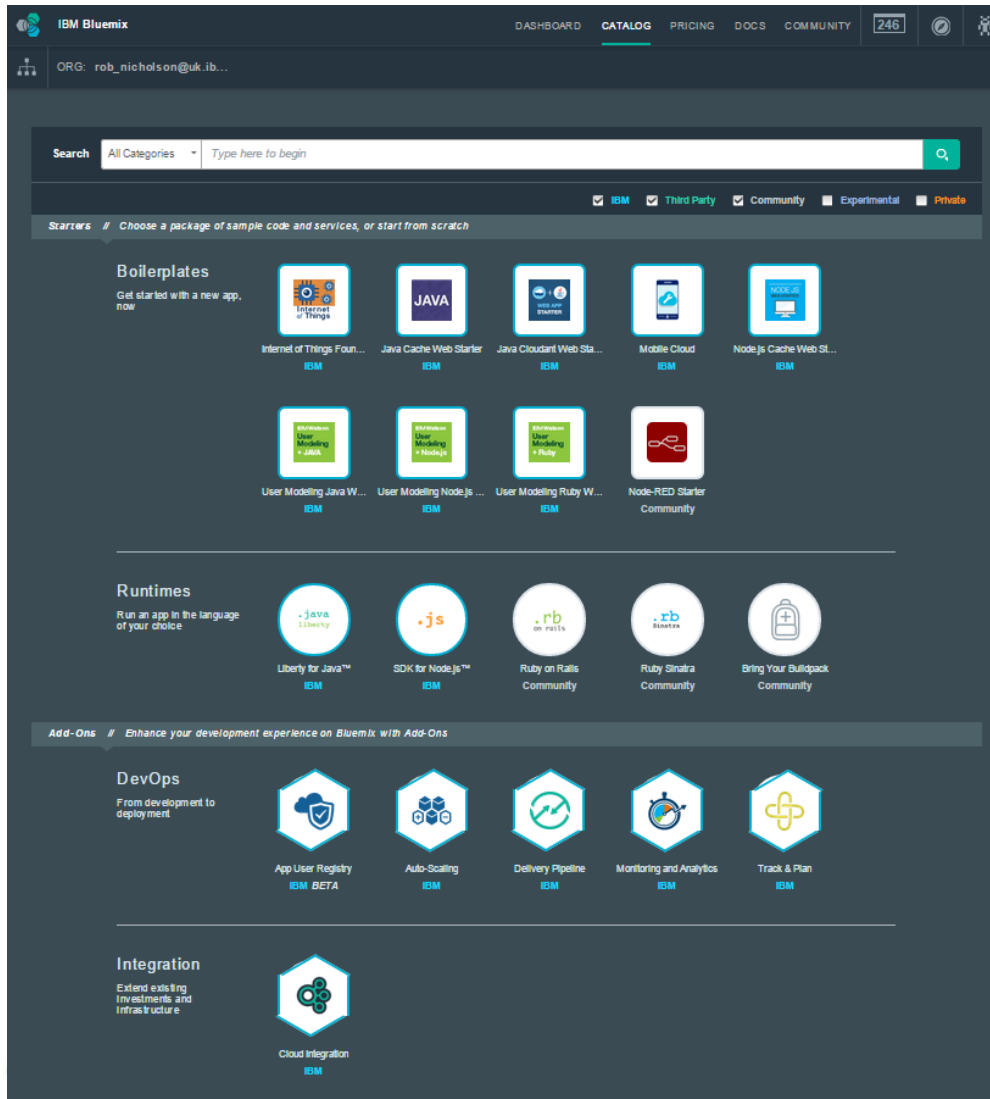
```
mwhitehead@ubuntu-vm: /opt/mqmAMQP

:
:
display chstatus(*) chltype(AMQP) clientid(*) all
2 : display chstatus(*) chltype(AMQP) clientid(*) all
AMQ8417: Display Channel Status details.
CHANNEL(MYAMQP)
STATUS(RUNNING)
KAINT(0)
CLNTUSER( )
MSGRCVD(0)
INDOUBTOUT(0)
LMSGDATE(2015-01-25)
CHLSDATE(2015-01-25)
PROTOCOL(AMQP)
CLIENTID(recv_fcce6d9)
CONNAME(192.168.56.1)
MCAUSER(mwhitehead)
MSGSENT(3)
INDOUBTIN(0)
PENDING(0)
LMSGTIME(03.29.31)
CHLSTIME(02.46.16)
```

DIS CHSTATUS command



MQ Light on IBM Bluemix



Run Your Apps

The developer can choose any language runtime or bring their own. Just upload your code and go.

DevOps

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

APIs and Services

A catalog of open source, IBM and third party APIs services allow a developer to stitch together an application in minutes.

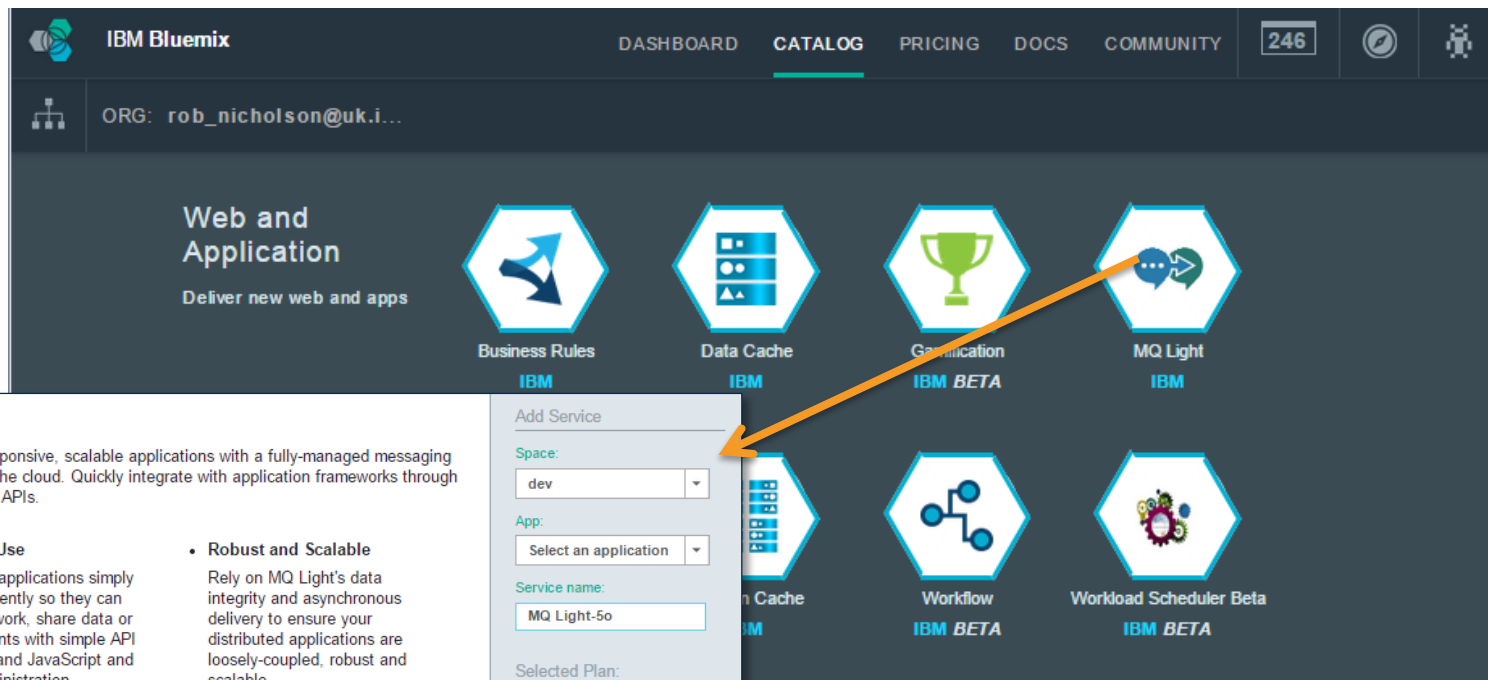
Cloud Integration


Build hybrid environments. Connect to on-premises systems of record plus other public and private clouds. Expose your own APIs to your developers.

Built on IBM SoftLayer

Runs automatically on top of IBM's leading infrastructure as a service. No need to worry about provisioning or managing infrastructure.

Introduction to MQ Light Service





MQ Light
IBM

PUBLISH DATE
9/26/2014

TYPE
Service

[VIEW DOCS](#)

Develop responsive, scalable applications with a fully-managed messaging provider in the cloud. Quickly integrate with application frameworks through easy-to-use APIs.

- Easy to Use**
Connect applications simply and efficiently so they can off-load work, share data or push events with simple API for Java and JavaScript and zero administration.
- Robust and Scalable**
Rely on MQ Light's data integrity and asynchronous delivery to ensure your distributed applications are loosely-coupled, robust and scalable.

Pick a plan Monthly prices shown are for country or region: [United Kingdom](#)

Plan	Features	Price
✓ MQ Light Standard Plan	Free allowance of 10,000 messages per month	£3.02 GBP/Million digital messages

This is the standard service plan for MQ Light charged in units of millions of messages per month.

Add Service

Space:
dev

App:
Select an application

Service name:
MQ Light-50

Selected Plan:
MQ Light Standard...

CREATE

Plan	Features	Price
✓ MQ Light Standard Plan	Free allowance of 10,000 messages per month	£3.02 GBP/Million digital messages
<p><i>This is the standard service plan for MQ Light charged in units of millions of messages per month.</i></p>		

Some other example cloud deployment patterns for MQ...



- Hypervisor editions
- IBM PureSystems patterns
 - MQ support for PureSystems virtual patterns (vsys.next) just released in 8.0.0.3
- MQ on Azure

MQ Hypervisor edition

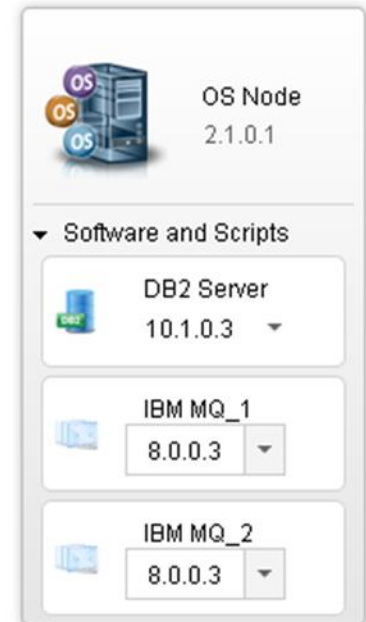


- MQ Hypervisor Editions allow automation and standardisation of the traditional approach to provisioning messaging systems, which combined with IBM Workload Deployer/PureApp gives many benefits:
 - Standardization of software images reduces risk and uncertainty
 - Automated provisioning reduces errors and speeds time to value
 - Repeatable configuration across sets of machines is quicker and less error-prone
 - Applying software maintenance is simpler and quicker using IWD/IPAS GUI or CLI
 - Comprehensive history/audit is maintained
 - License tracking is integrated

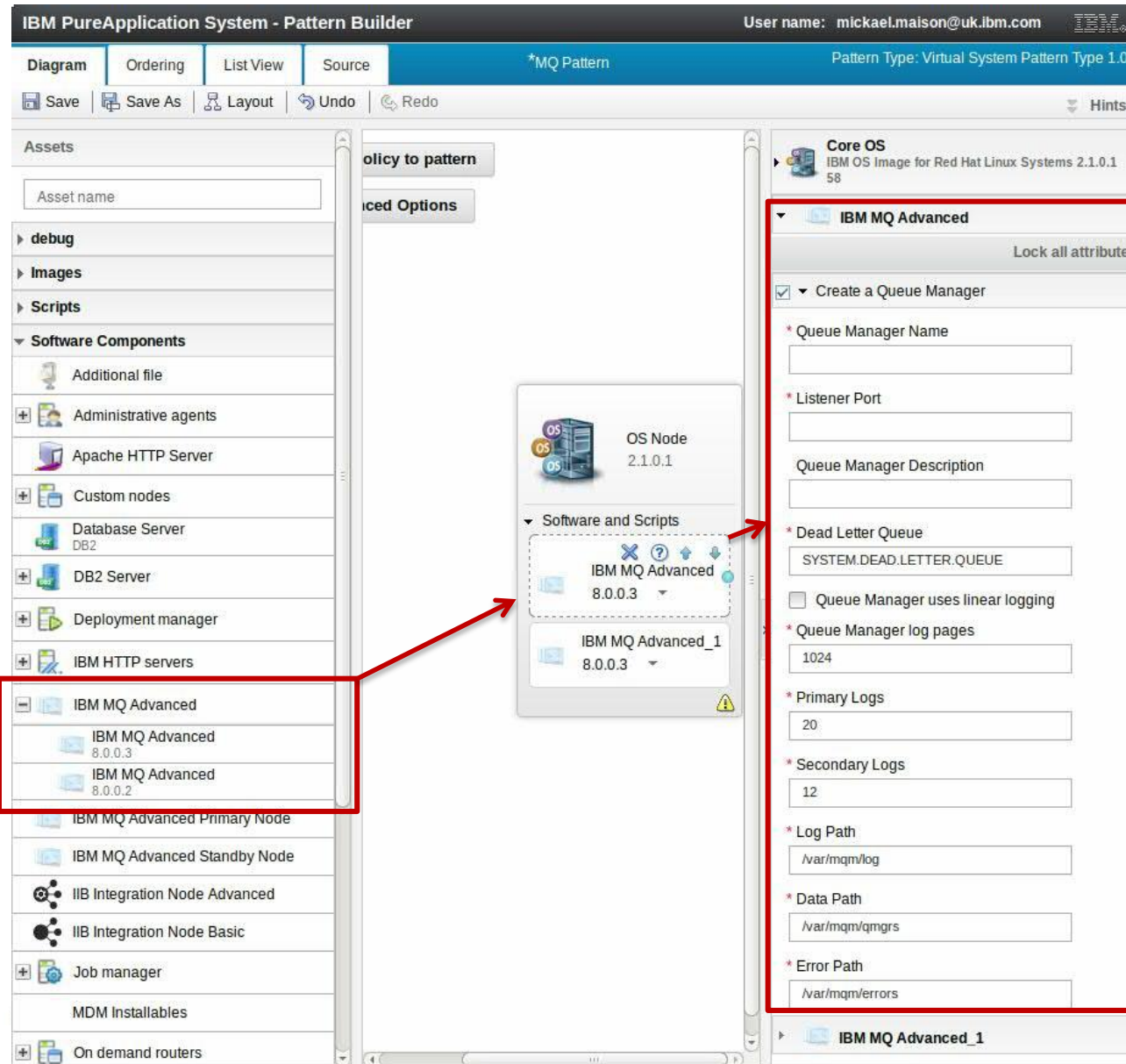
Complete your session evaluations online at www.SHARE.org/Orlando-Eval

MQ support for IBM PureSystems virtual patterns

- Conceptually similar to hypervisor but separates MQ package from OS
- Drag and drop MQ onto Core OS image to compose VM configuration – e.g.
 - DB2 / IIB / WAS and Multiple QMs on a VM
- Support for HA/DR with multi-instance Queue Managers
- Management and maintenance of MQ via PureApp console
- Delivered as core component of MQ from 8.0.0.3 onwards
- Available in:
 - Pure Application System (on-premise)
 - Pure Application Service (on Softlayer)
 - Pure Application Software (bring your own hardware)



MQ support for IBM PureSystems virtual patterns



IBM PureApplication System - Pattern Builder

User name: mickael.maison@uk.ibm.com

Diagram Ordering List View Source *MQ Pattern Pattern Type: Virtual System Pattern Type 1.0

Save Save As Layout Undo Redo Hints

Assets

Asset name

debug

Images

Scripts

Software Components

Additional file

Administrative agents

Apache HTTP Server

Custom nodes

Database Server DB2

DB2 Server

Deployment manager

IBM HTTP servers

IBM MQ Advanced

IBM MQ Advanced 8.0.0.3

IBM MQ Advanced 8.0.0.2

IBM MQ Advanced Primary Node

IBM MQ Advanced Standby Node

IIb Integration Node Advanced

IIb Integration Node Basic

Job manager

MDM Installables

On demand routers

policy to pattern

Advanced Options

OS Node 2.1.0.1

Software and Scripts

IBM MQ Advanced 8.0.0.3

IBM MQ Advanced_1 8.0.0.3

Core OS

IBM OS Image for Red Hat Linux Systems 2.1.0.1 58

IBM MQ Advanced

Lock all attributes

☒ Create a Queue Manager

* Queue Manager Name

* Listener Port

Queue Manager Description

* Dead Letter Queue

SYSTEM.DEAD.LETTER.QUEUE

☐ Queue Manager uses linear logging

* Queue Manager log pages

1024

* Primary Logs

20

* Secondary Logs

12

* Log Path

/var/mqm/log

* Data Path

/var/mqm/qmgrs

* Error Path

/var/mqm/errors

IBM MQ Advanced_1

- Selecting to deploy an instance of the MQ package allows automatic creation and configuration of an MQ queue manager
- Once deployed, the system runs as a standard MQ Advanced queue manager, with the same capabilities as a natively installed queue manager on that platform

MQ on Azure

- Recently announced offering allows enterprises to deploy existing IBM licences into managed systems in the Azure cloud
- MQ is now available as a deployable pattern
- Enables scalable MQ deployments using Azure's platform- and infrastructure-as-service capabilities



Links

- MQ Light
 - <https://developer.ibm.com/messaging/mq-light/>
- MQ Light for Bluemix
 - <http://www.bluemix.net>
- MQ Hypervisor edition for Red Hat
 - <http://www.ibm.com/software/products/en/wmq-hyper-redhat/>
- MQ Hypervisor edition for AIX
 - <http://www.ibm.com/software/products/en/wmq-hypervisor-aix>
- What's new for virtual system patterns in Pure Application System 2.0
 - http://www.ibm.com/developerworks/websphere/library/techarticles/1412_apte/1412_apte.html
- IBM MQ 8.0.0.3 Virtual System Pattern
 - https://www.ibm.com/developerworks/community/blogs/explorepatterns/entry/IBM_MQ_8_0_0_3_Virtual_System_Pattern?lang=en
- MQ on Microsoft Azure
 - <https://msopentech.com/blog/2014/11/04/ibm-websphere-mq-db2-now-microsoft-azure/>
- Extending IBM WebSphere MQ WebSphere Message Broker to the Clouds
 - <https://ibm.biz/BdENPZ>

Agenda

- Does my traditional MQ network make sense in the cloud era?
- Client user growing pains
- Rapid development
- Scalability
- Provisioning MQ for z/OS

mqaas – Using z/OSMF to provision MQ for z/OS

Agenda

- z/OS Cloud Services Objectives
 - Rapidly stand up/down z/OS subsystems
- mqaas POC Objectives
- mqaas POC Approach
 - Queue Manager Configuration Tasks
 - Tasks to provision/de-provision Queue Managers
 - z/OSMF Workflows
 - z/OSMF MQ Workflows
 - REST APIs to invoke z/OSMF Workflows
- Benefits of using such Services
- mqaas POC - future work
- Questions

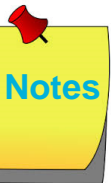
Complete your session evaluations online at www.SHARE.org/Orlando-Eval

z/OS Cloud Services Objectives

- **A zBLC Initiative**
 - **Proof of Concept to provide zCloud services**
- **Provision/De-provision z/OS Middleware Subsystems**
 - MQ, CICS, DB2, IMS and Liberty Profile Server
 - For use by Application Developers/Testers (***Service Consumers***)
- **Self-Service, Rapid, On Demand and With Ease**
 - At a click of a button from a Service Portal
 - Automated
 - Minimum/No input parameters
 - No z/OS/JCL skills required
- **System Administrators (*Service Providers*) to have overall control**
 - Pre-allocate infrastructure
 - Control usage via pre-set properties and customizable scripts/workflows
 - Enforce corporate/site resource naming standards

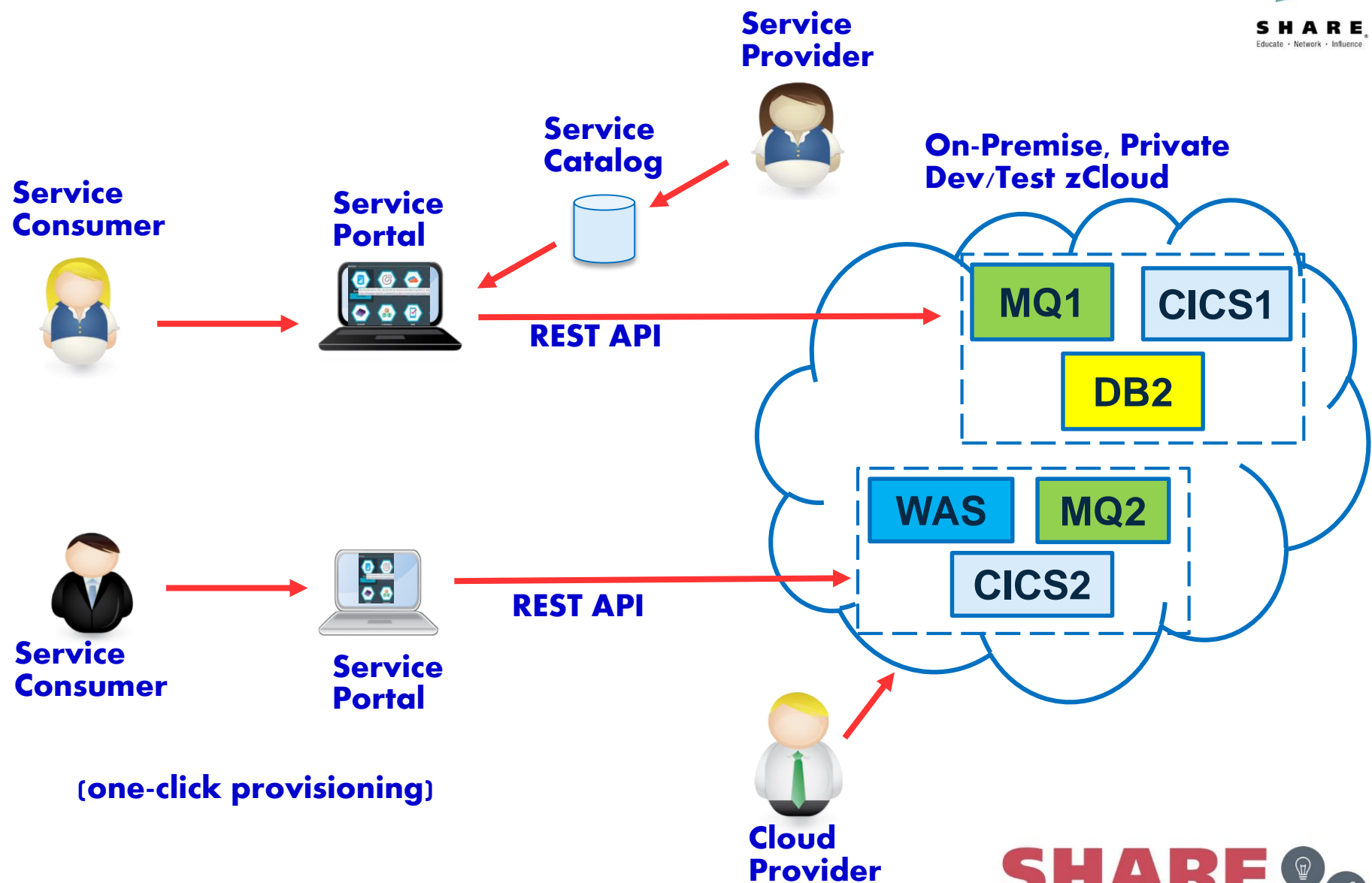
z/OS Cloud Services Objectives

- Following discussions with numerous z/OS customers, a proof of concept (POC) was kicked off in 1Q 2015 to provide services to provision and de-provision MQ, CICS, DB2, IMS and Liberty Profile subsystems (in this presentation, we focus on the provisioning/de-provisioning of MQ for z/OS subsystems (this includes both the Queue Manager and Channel Initiator address spaces)).
- Provision/de-provision should be:
 - Self-service and on demand. It should be possible for application developers and testers (i.e. the Service Consumers) to provision/de-provision subsystems on demand at a click of a button from a graphical user interface.
 - Rapid and with ease, with little or no z/OS skill required. It should be possible to invoke services at the push of a button from the user interface with no/minimum parameters. Services should drive automated and customizable scripts/workflows to provision/de-provision subsystems. Subsystems are for DEV/TEST use but maybe also for QA/PROD. But, typically expect to provision in DEV/TEST and promote through to QA and PROD.
- System programmers/administrators (i.e. the Service Providers) should be able to maintain overall control. They should be able to define an infrastructure onto which subsystems can be provisioned via automated scripts. By using scripts, system programmers/administrators can pre-set values for properties and define corporate naming standards.



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Rapidly stand-up/down z/OS Subsystems



(one-click provisioning)

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

mqaas POC - Objectives

- Provide System Services to:
 - Provision/De-provision a standalone Queue Manager
 - Define/Display Queues
- Invoke Services via a REST API

mqaas POC - Objectives

- In Phase 1 of the POC, the objectives for MQ for z/OS as a service (mqaas) were to provide services to:
 - Provision/de-provision a standalone Queue Manager.

(Queue Managers in a Queue Sharing Group (QSG) may be considered later).
- Define/Display Queues



Notes

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

mqaas POC - Approach

- Looked at:
 - Current Queue Manager Configuration tasks
 - Tasks for provisioning/de-provisioning Queue Managers
 - z/OSMF Workflows – the Building Blocks
 - MQ z/OSMF Workflows
 - REST APIs to invoke MQ z/OSMF workflows

Queue Manager Configuration Tasks

Task
Task 1: Identify the z/OS system parameters
Task 2: APF authorize the WebSphere MQ load libraries
Task 3: Update the z/OS link list and LPA
Task 4: Update the z/OS program properties table
Task 5: Define the WebSphere MQ subsystem to z/OS
Task 6: Create procedures for the WebSphere MQ queue manager
Task 7: Create procedures for the channel initiator
Task 8: Define the WebSphere MQ subsystem to a z/OS WLM service class
Task 9: Select and set up your coupling facility offload storage environment
Task 10: Set up the coupling facility
Task 11: Implement your ESM security controls
Task 12: Update SYS1.PARMLIB members
Task 13: Customize the initialization input data sets
Task 14: Create the bootstrap and log data sets
Task 15: Define your page sets
Task 16: Add the WebSphere MQ entries to the DB2 data-sharing group
Task 17: Tailor your system parameter module
Task 18: Tailor the channel initiator parameters

For POC, assume SMPE install of MQ plus some tasks already done by System Programmer

For POC, stand-alone QMgr, QSG to be considered later

For POC, stand-alone Qmgr, QSG to be considered later

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Queue Manager Configuration Tasks ..

Task
Task 19: Set up Batch, TSO, and RRS adapters
Task 20: Set up the operations and control panels
Task 21: Include the WebSphere MQ dump formatting member
Task 22: Suppress information messages
Task 23: Update your system DIAG member for AMS
Task 24: Create procedures for Advanced Message Security
Task 25: Set up the started task user Advanced Message Security
Task 26: Grant RACDCERT permissions to the security administrator for Advanced Message Security
Task 27: Grant users resource permissions for WebSphere MQ Advanced Message Security

All to be
considered
later

Queue Manager Configuration Tasks

- For the POC, we tackle the tasks in turquoise in the provision.xml workflow (see next slide).
- The POC does not do much in the Channel Initiator yet
 - Services to configure ports, listeners and channels will be considered in the future
- We have yet to consider how some of the tasks in white (i.e SMPE installs, setting program properties, setting link pack areas, etc) should be handled. These system operations may be handled by new function that is yet to be designed and written in z/OSMF.
- We are only creating a standalone Queue Manager for the POC so tasks to do with configuring a Queue Manager to be a member of a Queue Sharing Group (QSG) will be addressed at a later date.
- We will also consider automating tasks to configure adapters, dump formatters, etc. at a later date. Though some of these may be handled by new function yet to be designed and written in z/OSMF.
- In the future, we hope to identify those steps in a workflow that are common to provisioning/de-provisioning z/OS subsystems and code them only once. Currently each z/OS subsystem has its own implementation of the steps in a workflow.
- Today, TCPIP ports have to be manually configured
 - Port assignment for specific jobs in z/OS Communication Server
 - MQ uses port 1414 by default but users can change and/or define additional ports
- In future, we will use z/OS Communication Server RESTful Cloud Services to configure TCP/IP stacks and ports



Notes

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Steps to provision a z/OS Queue Manager

Step	Description	JCL
1	Specify Queue Manager Criteria	-
2	Define APF Load Library	CSQ4ZNAP.jcl
3	APF authorise the Load Library	CSQ4ZAAU.jcl
4	Generate MQ Initialization Parms (ZPARms)	CSQ4ZZPM.jcl
5	Define BSDS and Logs	CSQ4ZNBL.jcl
6	Define Pagesets	CSQ4ZNPG.jcl
7	Construct QMGR startup JCL	CSQ4ZMST.jcl
8	Use SETSSI to define subsystem to z/OS	CSQ4ZSSI.jcl
9	Construct CHINIT Startup JCL	CSQ4ZCHN.jcl
10	Construct DISX (CHINIT startup command)	CSQ4ZDSX.jcl
11	Disable subsystem security	CSQ4ZNSE.jcl
12	Start QMGR	CSQ4ZSQM.jcl

Further potential to merge and reduce steps
Or add installation specific steps (e.g. perform IVP, backups, monitoring, etc.)

Steps to de-provision a z/OS Queue Manager

Step	Description	JCL
1	Stop Channel Initiator	CSQ4ZTCN.jcl
2	Stop Queue Manager	CSQ4ZTQM.jcl
3	Wait for Subsystems to stop	CSQ4ZWSS.rex
4	Delete subsystem security permissions	CSQ4ZDSE.JCL
5	Delete MQ specific Datasets	CSQ4ZDDS.JCL

De-provision everything that was provisioned (but undo of SETSSI not supported yet)

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

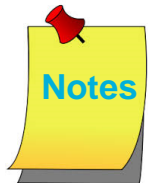
Steps to provision a z/OS Queue Manager

- Notice that we have introduced a few additional steps.
 - For example, 'Specify Queue Manager Criteria'. If the workflow is run (performed) manually from within z/OSMF, criteria like:
 - Environment
 - Subsystem ID
 - User ID
 - Enable/Disable Security

can be specified but, if the workflow is run 'automatically' from the http client, these values are currently hardcoded overrides on the POST REST API call.

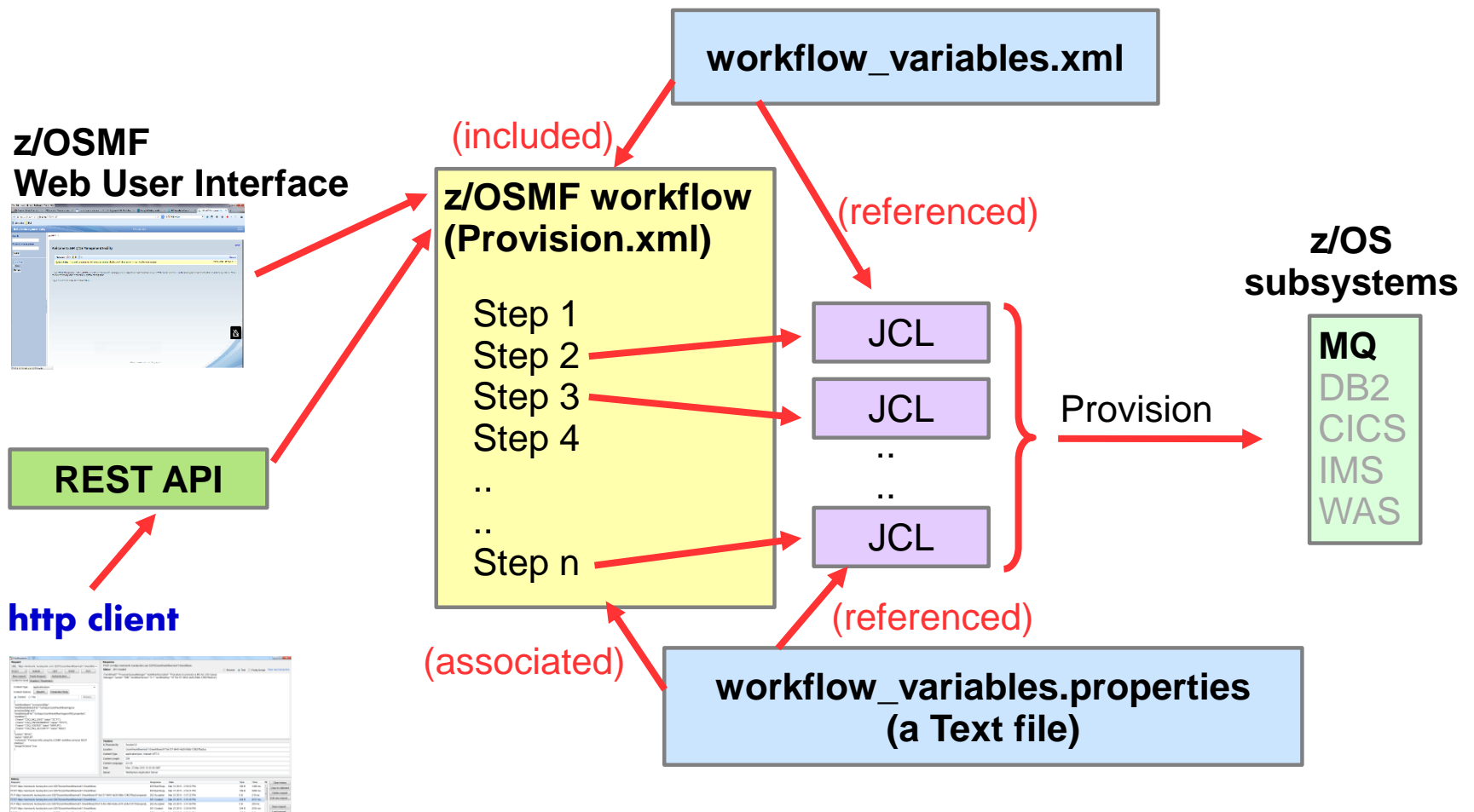
Emphasise that we are performing a set of steps here and not just running a 'crtmqm' command. Though, the overall provision.xml workflow can be viewed as being equivalent to 'crtmqm'.

- The next slides highlight some points about workflows and their associated files.



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

z/OSMF Workflows – the Building Blocks



Note: Steps in a workflow can a) just gather data, b) submit JCL, c) invoke a REXX EXEC/shell script, d) call another workflow.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

z/OSMF Workflows – the Building Blocks

- A z/OSMF workflow consists of:
 - XML file(s) that Define a set of steps to be performed, typically in sequence
 - A Variable File (optional) that lists the variables in use by a workflow
 - A properties file (optional) that defines the initial values for the variables
- Workflow instances are:
 - Created from workflow XML files
- A step in a workflow can:
 - Just gather input
 - Submit an associated JCL job
 - Invoke an associated REXX exec/shell script
 - Call another workflow
- Workflows are driven:
 - By a REST API
 - From a z/OSMF Web User Interface
- The picture on the previous slide shows the structure of a z/OSMF workflow.
- Variables can be defined inline (within a workflow XML file) but, if they are defined in a separate variable XML file they can be included in more than one workflow. For example, it is likely that variables referenced by a provision workflow XML file may also be referenced by a de-provision workflow XML file.
- The variables can also be assigned default values (properties) inline (within a workflow XML file) but, if they are defined in a separate properties file (which is simply a flat text file) they can be customized with ease. For example, once the XML workflow and variable files are defined, they don't need to be touched again. Only the properties file need be updated to customize the values of the variables.



Notes

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

mqaas POC – Provisioning the QMgr

http Client (frontend)

- Firefox plugin http requester REST API client (could also use GOOGLE's POSTMAN plugin)
 - Issues REST APIs to drive Services on the backend z/OS system
 - POST is issued to create a z/OSMF workflow
 - PUT is issued to start (run) a workflow
- z/OSMF Web User Interface

Services (development)

- Implemented as z/OSMF workflows (xml, properties and JCL files)
 - Apache Velocity scripting used to simplify JCL (see later example)
- Created in an XML editor
- FTPed to backend
- Held in a directory in Unix System Services
- z/OSMF instance can also be created and run via the z/OSMF Web User Interface

Services (backend)

- z/OSMF workflow instances:
 - Created as a result of POST requests from the http client
 - Started as a result of PUT requests from the http client
 - Provision/de-provision Queue Managers (and Channel Initiators)

Host (backend)

- z/OS System with the MQ load libraries pre-installed
- Target for running provisioned Queue Managers (and Channel Initiators)



Notes

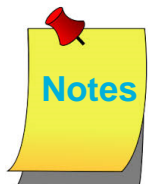
Complete your session evaluations online at www.SHARE.org/Orlando-Eval

z/OSMF MQ Workflows

- The MQ **provision.xml** workflow file consists of steps to:
 - Customize and Configure a Queue Manager
 - Start the Queue Manager (and Channel Initiator)
- The MQ **deprovision.xml** workflow file consists of steps to:
 - Stop the Queue Manager (and Channel Initiator)
 - Delete all resources that were created when provisioning the Queue Manager
- The queueServices.xml workflow file consists of steps to:
 - Define a Queue
 - Display a Queue

z/OSMF MQ Workflows

- We discuss the current customization tasks required to configure a Queue Manager in the next two slides.
- We then discuss:
 - A) The customization tasks that we perform in the provision.xml workflow to provision a Queue Manager.
 - B) The tasks that we perform in the deprovision.xml workflow to de-provision a Queue Manager.



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

provision.xml – steps to define datasets

```

C:\zsmf\mqprov\provision.xml - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

provision.xml workflow_variables.xml workflow_variables.properties CSQ4ZNAP.jcl CSQ4ZAAU.jcl CSQ4ZZPM.jcl

98 <!-- *****
99 Step - Define the BSDS and LOG DATASETS
100 ***** -->
101 <step name="defBSDSLogs" optional="false">
102 <title>Define the BSDS and Log Datasets</title>
103 <description>Job CSQ4ZNBL.jcl defines the BSDS and Log Datasets.</description>
104 <prereqStep name="genInitParms"/>
105 <instructions>Job CSQ4ZNBL.jcl defines the BSDS and Log Datasets.</instructions>
106 <weight>1</weight>
107 <skills>MQ administration</skills>
108 <autoEnable>true</autoEnable>
109 <template>
110 <fileTemplate substitution="true">CSQ4ZNBL.jcl</fileTemplate>
111 <submitAs>JCL</submitAs>
112 <maxLrecl>80</maxLrecl>
113 </template>
114 </step>
115
116 <!-- *****
117 Step - Define the PAGESETS
118 ***** -->
119 <step name="defPAGE" optional="false">
120 <title>Define the PAGESETS</title>
121 <description>Job CSQ4ZNPG.jcl defines the PAGESETs.</description>
122 <prereqStep name="defBSDSLogs"/>
123 <instructions>Job CSQ4ZNPG.jcl defines the PAGESETs.</instructions>
124 <weight>1</weight>
125 <skills>MQ administration</skills>
126 <autoEnable>true</autoEnable>
127 <template>
128 <fileTemplate substitution="true">CSQ4ZNPG.jcl</fileTemplate>
129 <submitAs>JCL</submitAs>
130 <maxLrecl>80</maxLrecl>
131 </template>
132 </step>

```

Note:

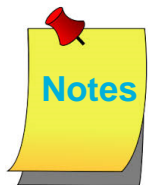
Step to Define BSDS and LOGS.
Submits JCL job CSQ4ZNBL.

Note:

Step to Define PAGESETs.
Submits JCL job CSQ4ZNPG.

provision.xml – steps to define datasets

- This slide shows a snapshot of the provision.xml file.
- Two steps are shown:
 - defBSDSLogs – submits JCL job CSQ4ZNBL.JCL to define the BSDS and Logs
 - defPAGE – submits JCL job CSQ4ZNPG.JCL to define the PAGESETs.
- Notice that the defBSDSLogs step is defined as a pre-requisite step to the defPAGE step
- Also note that both steps are defined to be performed automatically.



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

workflow_variables.xml

```
C:\zosmf\mqprov\workflow_variables.xml - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
provision.xml workflow_variables.xml workflow_variables.properties CSQ4ZNAP.jcl CSQ4ZAAU.jcl

11 <variable name="CSQ_ENVIRONMENT" scope="instance">
12   <label>ENVIRONMENT</label>
13   <abstract>Environment for which the queue manager is to be provisioned</abstract>
14   <description>The environment for which the queue manager is to be provisioned</description>
15   <category>MQ</category>
16   <string valueMustBeChoice="true">
17     <choice>DEV</choice>
18     <choice>TEST</choice>
19     <choice>QA</choice>
20     <choice>PROD</choice>
21   </string>
22 </variable>
23 <variable name="CSQ_DEV_DUAL_BSDS" scope="instance">
24   <label>DEV_DUAL_BSDS</label>
25   <abstract>Dual BSDS for DEV</abstract>
26   <description>Are dual BSDSs to be used for DEV ?</description>
27   <category>MQ</category>
28   <boolean>
29   </boolean>
30 </variable>
31 <variable name="CSQ_DEV_DUAL_ACTV_LOGS" scope="instance">
32   <label>DEV_DUAL_ACTV_LOGS</label>
33   <abstract>Dual Active Logs for DEV</abstract>
34   <description>Are dual Active Logs to be used for DEV ?</description>
35   <category>MQ</category>
36   <boolean>
37   </boolean>
38 </variable>
```

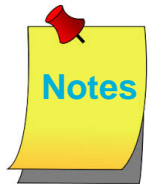
Note:

Variables to control properties for DEV environment:

- CSQ_DEV_DUAL_BSDS
- CSQ_DEV_DUAL_ACTV_LOGS
- etc.

workflow_variables.xml

- This slide shows a snapshot of the workflow_variables.xml file. It highlights the set of variables that are used to define the values of properties used when provisioning a Queue Manager for DEvelopment purposes.
- The workflow_variables.xml file similarly contains sets of variables that are used to define the values of properties used when provisioning a Queue Manager for TEST, QA or PROD.



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

workflow_variables.properties

```
C:\zosmf\mqprov\workflow_variables.properties - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
provision.xml workflow_variables.xml workflow_variables.properties CSQ4ZNAP.jcl CSQ4ZAAU.jcl

5  ## Environment - DEV=Development, TEST=Test, QA=Quality Assurance, PROD
6  CSQ_ENVIRONMENT = TEST
7  ##
8  ## Parameters for DEV environment
9  CSQ_DEV_DUAL_BSDS = false
10 CSQ_DEV_DUAL_ACTV_LOGS = false
11 CSQ_DEV_NO_OF_ACTV_LOGS = 1
12 CSQ_DEV_ACTV_LOG_OFFLOAD = false
13 CSQ_DEV_DUAL_ARCV_LOGS = false
14 CSQ_DEV_NO_OF_PAGESETS = 5
15 ##
16 ## Parameters for TEST environment
17 CSQ_TEST_DUAL_BSDS = false
18 CSQ_TEST_DUAL_ACTV_LOGS = false
19 CSQ_TEST_NO_OF_ACTV_LOGS = 2
20 CSQ_TEST_ACTV_LOG_OFFLOAD = false
21 CSQ_TEST_DUAL_ARCV_LOGS = false
22 CSQ_TEST_NO_OF_PAGESETS = 5
23 ##
24 ## Parameters for QA environment
25 CSQ_QA_DUAL_BSDS = true
26 CSQ_QA_DUAL_ACTV_LOGS = true
27 CSQ_QA_NO_OF_ACTV_LOGS = 4
28 CSQ_QA_ACTV_LOG_OFFLOAD = true
29 CSQ_QA_DUAL_ARCV_LOGS = false
30 CSQ_QA_NO_OF_PAGESETS = 8
```

DEV

TEST

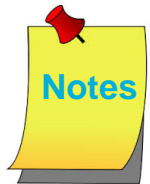
QA

Note:

Can define different characteristics for Queue Managers based on the environment they are being created for !

workflow_variables.properties

- This slide shows a snapshot of the workflow_variables.properties file. It highlights how different properties can be set for environment specific variables to define different characteristics for the Queue Manager being provisioned (e.g. 1 active log for DEV, 2 active logs for TEST, 4 active logs for QA, etc.)



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Automatic variable substitution

```
C:\zosmf\mqprov\CSQ4ZNB.jcl - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

provision.xml workflow_variables.xml workflow_variables.properties CSQ4ZNAP.jcl CSQ4ZAAU.jcl CSQ4ZZPM.jcl CSQ4ZNB

161      RECORDS(5 5)
162      CONTROLINTERVALSIZE(1024))
163 #end
164 ##
165 ## Allocate active logs
166 ##
167 ## TODO: Use CSQ_LOG1B_VOLUME and CSQ_LOG2B_VOLUME
168 ##
169 #foreach ($logcopy in [1..$noofcopiesofactvlogs])
170 #foreach ($log in [1..$noofactvlogs])
171     DEFINE CLUSTER -
172         (NAME ($instance-CSQ_BLP_HLQ).$instance-CSQ_MQ_SSTID).LOGCOPY.$logcopy.DS00$log)
173         LINEAR
174 #if ($logcopy == 1)
175     VOLUMES($instance-CSQ_LOG1A_VOLUME) -
176 #else ($logcopy == 2)
177     VOLUMES($instance-CSQ_LOG2A_VOLUME) -
178 #end
```

Note:

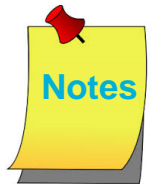
`${instance-*}` variables are automatically substituted at runtime. No more `++var++` values in JCL !

Note:

`${*}` local variables are also automatically resolved at runtime.

Automatic variable substitution

- This slide shows a snapshot of CSQ4ZNBL.JCL (JCL to define the BSDS and Logs). It highlights the `${instance-*}` z/OSMF variables which are automatically substituted at runtime with values defined in the properties file or with overriding values specified at workflow creation time (if creating the workflow via the POST REST API) or at workflow runtime (if running the workflow via the z/OSMF Web User Interface).
- In the current MQ configuration task sample JCL, System Administrators need to manually replace `++var++` values with values that are suitable for their environment.



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Apache Velocity Scripting to limit JCL

```
C:\zosmf\mqprov\CSQ4ZNB1.jcl - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
provision.xml workflow_variables.xml workflow_variables.properties CSQ4ZNAP.jcl CSQ4Z
161 RECORDS(5 5) -
162 CONTROLINTERVALSIZE(1024))
163 #end
164 ##
165 ## Allocate active logs
166 ##
167 ## TODO: Use CSQ_LOG1B_VOLUME and CSQ_LOG2B_VOLUME
168 ##
169 #foreach ($logcopy in [1..$noofcopiesofactvlogs])
170 #foreach ($log in [1..$noofactvlogs])
171     DEFINE CLUSTER -
172         (NAME (${instance-CSQ_BLP_HLQ}.${instance-CSQ_MQ_SSID}).LOG
173         LINEAR -
174 #if ($logcopy == 1)
175         VOLUMES(${instance-CSQ_LOG1A_VOLUME}) -
176 #else ($logcopy == 2)
177         VOLUMES(${instance-CSQ_LOG2A_VOLUME}) -
178 #end
179     SHAREOPTIONS(2 3) -
180     RECORDS(180000) )
181     DATA -
182     (NAME (${instance-CSQ_BLP_HLQ}.${instance-CSQ_MQ_SSID}).LOGC
```

Note:

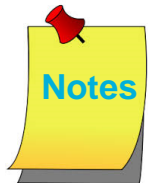
Can use Apache Velocity 'foreach' loops to avoid repetitive JCL !

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Apache Velocity Scripting to limit JCL

- This slide shows a snapshot of the CSQ4ZNBL (define BSDS and Logs) JCL to highlight how z/OSMF allows the use of Apache Velocity Scripting Language statements to limit JCL statements. Notice how the number of active logs can be defined within a foreach loop. This means that the DEFINE CLUSTER command to define the active log need not be coded repeatedly for each active log (as is the case with the current MQ sample JCL for customizing the Queue Manager).

Apache Velocity Scripting Language statements clearly need to be added to the JCL but this example shows just how powerful they can be.



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Step to invoke a REXX exec

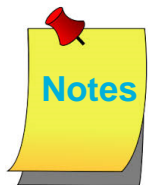
```
C:\zosmf\mqprov\deprovision.xml - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
CSQ4ZNBL.jcl CSQ4ZNPG.jcl CSQ4ZMST.jcl CSQ4ZSSL.jcl CSQ4ZCHN.jcl CSQ4ZDS>

62 <autoEnable>true</autoEnable>
63 <template>
64   <fileTemplate substitution="true">CSQ4ZTQM.jcl</fileTemplate>
65   <submitAs maxRc="0">JCL</submitAs>
66   <maxLrecl>80</maxLrecl>
67 </template>
68 </step>
69 <!-- *****
70      Step - Wait for Channel Initiator and Queue Manager to STOP
71      *****
72 <step name="waitForSSstop" optional="false">
73   <title>Wait for Subsystems to Stop</title>
74   <description>REXX Exec CSQ4ZWSS.rex waits for the Channel Initi
75   <prereqStep name="stopQMGR"/>
76   <instructions>Wait for Channel Initiator and Queue Manager to s
77   <weight>1</weight>
78   <skills>MQ administration</skills>
79   <autoEnable>true</autoEnable>
80   <template>
81     <fileTemplate substitution="true">CSQ4ZWSS.rex</fileTemplate>
82     <submitAs maxRc="0">TSO-REXX-JCL</submitAs>
83     <maxLrecl>1024</maxLrecl>
84   </template>
```

Note:
REXX exec waits for
QMGR to stop before
proceeding to next step.

Step to invoke a REXX exec

- This slide shows a snapshot of the deprovision.xml file which has a step to invoke a REXX exec. The exec determines if the Queue Manager and Channel Initiator subsystems have terminated or not. If not, it waits for a few minutes to allow them to terminate before it attempts to CANCEL (and FORCE, if required) them so that resources used by the Queue Manager and Channel Initiator can be deleted during de-provision.
- z/OSMF actually builds some JCL to wrapper around the exec before submitting the JCL to invoke the exec.
- Note, when invoking REXX execs in this way, it is important that the maxLrecl value is set to 1024 and not to 80 (as is the case when submitting JCL).



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Define QLOCAL Workflow

```
CSQ4ZMST.jcl  CSQ4ZSSI.jcl  CSQ4ZCHN.jcl  CSQ4ZDSX.jcl  CSQ4ZNSE.jcl  CSQ4ZSQM.jcl  depr
87      <!-- This is a step to create a queue -->
88      <step name="CreateQ">
89          <title>Create Queue </title>
90          <description>
91              A DEFINE QLOCAL MQSC command is issued based on user input
92          </description>
93          <variableValue name="qm_name" required="true"/>
94          <variableValue name="q_name" required="true"/>
95          <variableValue name="index" required="true"/>
96          <variableValue name="MaxDepth" required="true"/>
97
98          <instructions substitution="true">
99              The following MQSC command can be used to define your queue:
100              When you have reviewed the queue definition parameters, press <s
101              <br/>
102              <code>DEFINE QLOCAL( $instance-q_name ) INDXTYPE($instance-index
103          </instructions>
104          <weight>5</weight>
105          <skills>User</skills>
106          <autoEnable>true</autoEnable>
107          <template>
108              <fileTemplate substitution="true">./define.jcl</fileTemplate>
109              <submitAs maxRc="0">JCL</submitAs>
110              <maxLrecl>80</maxLrecl>
111          </template>
```

- **define.jcl** currently calls CSQUTIL to issue the DEFINE QLOCAL MQSC command

Example POST (create workflow) request

HttpRequester

Request

URL

Content to Send

Content Type:

Content Options:

☒ Content ☐ File

```
{
  "workflowName": "provisionQMGr",
  "workflowDefinitionFile": "/u/mayur/zosmf/workflow/mqprov/provision.xml",
  "variableInputFile": "/u/mayur/zosmf/workflow/mqprov/workflow_variables.properties",
  "variables": [
    { "name": "CSQ_MQ_SSID", "value": "ZCT1" },
    { "name": "CSQ_ENVIRONMENT", "value": "TEST" },
    { "name": "CSQ_USERID", "value": "MAYUR" },
    { "name": "CSQ_MQ_SECURITY", "value": "false" }
  ],
  "system": "MV4C",
  "owner": "MAYUR",
  "comments": "Provision MQ using the zOSMF workflow services REST interface",
  "assignToOwner": true
}
```

Response

POST on https://winmvs4c.hursley.ibm.com:32070/zosmf/workflow/rest/1.0/workflows

Status: 201 Created

```
{ "workflowID": "ProvisionQueueManager", "workflowDescription": "Procedure to provision a MQ for zOS Queue Manager", "vendor": "IBM", "workflowVersion": "0.1", "workflowKey": "9715e137-9943-4a38-89bb-12f627fba3ca" }
```

Headers

X-Powered-By	Servlet/3.0
Location	/zosmf/workflow/rest/1.0/workflows/9715e137-9943-4a38-89bb-12f627fba3ca
Content-Type	application/json; charset=UTF-8
Content-Length	204
Content-Language	en-US
Date	Mon, 23 Mar 2015 15:35:48 GMT
Server	WebSphere Application Server

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Example PUT (start workflow) request

HttpRequester

Request

URL: <https://winmvs4c.hursley.ibm.com:32070/zosmf/workflow/rest/1.0/workflow>

PUT GET POST PUT

New request Paste Request Authentication...

Content to Send Headers Parameters

Content Type: application/json

Content Options: Base64 Parameter Body

☒ Content ☐ File Browse...

```
{
  "resolveConflictByUsing": "existingValue",
  "notificationUrl": "https://winmvs4c.hursley.ibm.com"
}
```

Response

PUT on <https://winmvs4c.hursley.ibm.com:32070/zosmf/workflow/rest/1.0/workflows/9715e137-9943-4a38-89bb-12f627fba3ca/operations/start>

Status: 202 Accepted

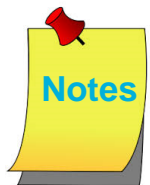
☐ Browser ☒ Text

Headers

X-Powered-By	Servlet/3.0
Content-Type	application/json; charset=UTF-8
Content-Language	en-US
Content-Length	0
Date	Mon, 23 Mar 2015 15:37:22 GMT
Server	WebSphere Application Server

POST/PUT REST APIs

- There is no need to spend too long on these slides. Just mention that the POST and PUT REST API calls can be used to create and run a z/OSMF workflow, respectively. This means that users can define their own portals to drive services if required.
- These two slides show how to issue the POST (create a workflow) and PUT (start a workflow) REST APIs. Here we are using the http requester REST client that comes with Firefox Mozilla but you can use any REST client (Google Chrome has a plugin for a REST client called POSTMAN).
- Notice the variable overrides on the POST request. These variables define the Queue Manager criteria.
- Being able to invoke workflows via the REST APIs means users can write their own, customized portals if required.



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Benefits of using mqaas

- **Useful in a cloud environment**
 - Rapidly provision Queue Managers to deploy applications to
 - Can introduce charging based on usage
 - No z/OS specific skills required
- **Useful in Development/Test (beta) environments**
 - Can share system resources to provision Queue Managers in a pool
 - Application Developers can provision/de-provision Queue Managers
 - No need to wait for System Administrators to action
 - Can stand-up Queue Managers (and Channel Initiators) when needed
 - Define resources required by applications
 - Can stand-down Queue Managers (and Channel Initiators) when finished
 - Delete all resources that were created
- **Useful for Automation**
 - Can automate Queue Manager configuration tasks
 - Administrator can still maintain overall control
 - Administrator can enforce resource naming conventions
 - Reduces need for z/OS skills
 - Can address future z/OS skills shortage issue
- **Note:** You can use z/OSMF today (without this POC) to automate subsystem configuration tasks !

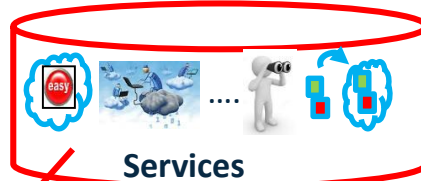
mqaas POC – future work

- Ship Provision/De-provision workflows as sample templates
 - We'd like to:
 - Turn more values into properties
 - Run an Installation Verification Test as a step in the MQ Provision workflow
- Working with customers to define requirements for zCloud
 - A managed beta is being run for SELECTED zBLC customers (i.e. those customers who are serious about helping us define the requirements for zCloud)
- Looking to see what templates to provide to allow customers to create services with ease
- Some potential Cloud Services are suggested on the next slide
- Hybrid Clouds
 - On-premise Cloud
 - SoftLayer Cloud (for additional capacity .. Elastic Scale)
- Patterns
 - CICS + MQ + DB2, WAS + MQ + DB2
- Provisioning of middleware on Clouds that span multiple platforms

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Potential Cloud Services

Service Registry/Catalog



Potential Users



Service Consumer



Service Provider

Service Portal



Resource Registry

Could hold:

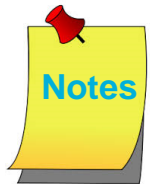
- Configuration/State of provisioned middleware Systems/Resources
- Can be used for backup or as models for cloning
- Historic monitoring and usage (audit) data

Note: Have corresponding de-provisioning services where required.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

mqaas POC – future work

- We are starting POC Phase 2 in Aug, 2015 and will probably continue with a few more phases before we decide if/when/how we will ship some of this function. Right now we are looking to work with more customers to better understand their requirements so that we can implement services that will be useful to our customers.



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Any Questions ?



Please complete your session evaluation .. Thank You !



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

IBM Hursley Park

Tues Aug 11th, 2015

Copyright and Trademarks

© IBM Corporation 2015. All Rights Reserved.

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.