How to Develop Responsive Applications with IBM MQ Light

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

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.
- 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.

Agenda

- Introduction & Background
- Use-case and features
- A new messaging API
- Deploying apps & IBM BlueMix
- Demo

IBM Messaging Focus Areas

Enable Developers to build more scalable, responsive applications

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

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

Capture Big Data from Mobile and Internet of Things

Focus on Internet-scale events, m2m device enablement, zero-admin, security and privacy, feed into real-time analytics, location-based notifications

What is “Application Messaging”?

Must use approved IT services

Share, re-use and save!

Jon
IT mgmt

Andy
Developer

Demand for Infrastructure services

Iain
Infrastructure Guy

Organisations moving in this direction

Beth
Business Sponsor

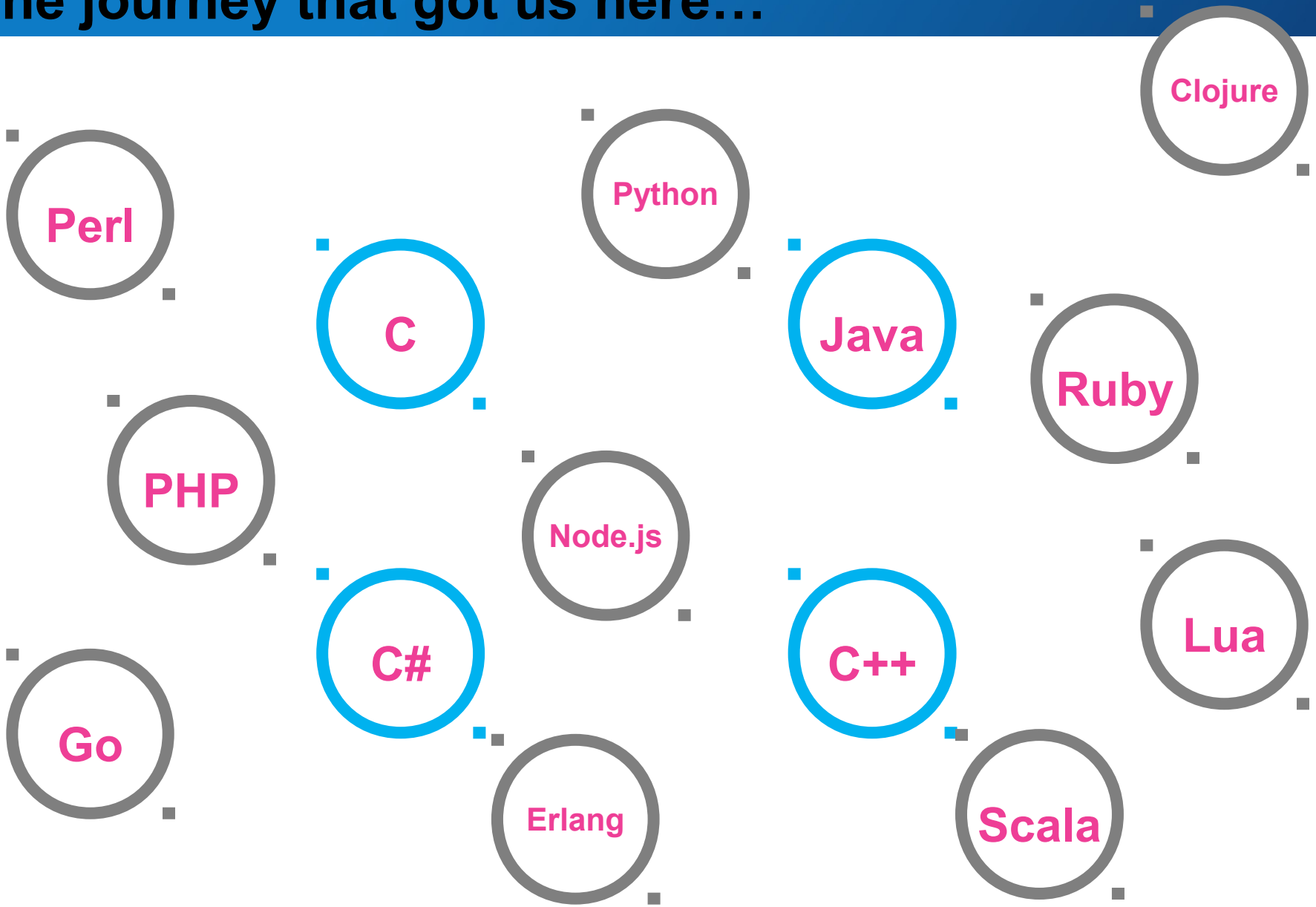
What handy tools can I grab?

Reduced pull for services

Andy
Developer

Iain
Infrastructure Guy

The journey that got us here…



What is 'Andy' the developer like?

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

What is MQ Light?

1. A new messaging API

2. A messaging runtime for on-premise development and deployment

3. A PaaS messaging runtime for admin-free cloud deployment (MQ Light Service in Bluemix)

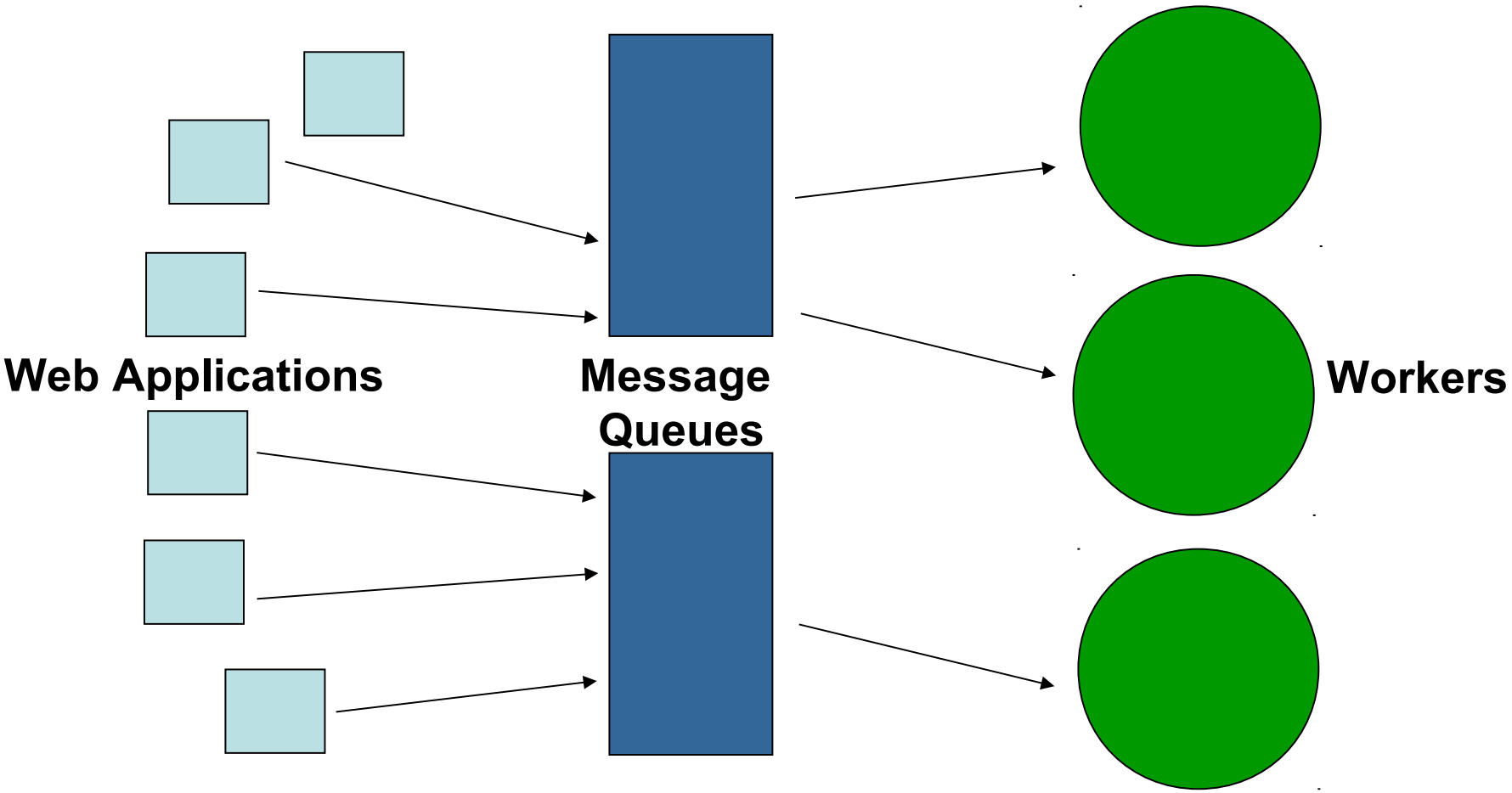
More on all of these throughout the slides...

Agenda

- Introduction & Background
- Use-case and features
- A new messaging API
- Deploying apps & IBM BlueMix
- Demo

A Typical Developer's Use-Case

I want to execute code without taxing my Web app processes



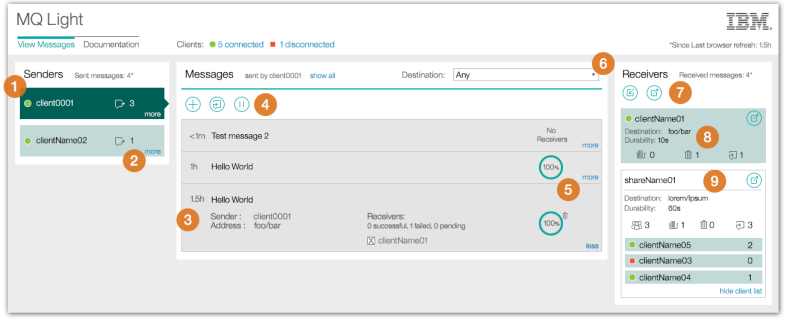
Application Messaging Features

Our new messaging offering will focus on the developer...

- A simple API

```
e.g. client.start()
client.send()
client.subscribe()
client.unsubscribe()
```

- Developer-oriented download & install process
- Web based tooling



- Open-source technologies

- Easy to deploy apps to a PaaS environment

Developer oriented download & install

We're addressing the inhibitors to developer approval

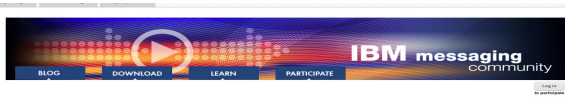
- Removing the sign-in step from the download process

- Removing the need to run an installer, allowing the product to be unzipped to any location

- Ensuring we don't require intrusive OS changes, e.g. altering the registry, setting up users

- Developer-centric platform support (see next slide)

- Providing a more developer-centric look & feel to the download site and support pages



Appropriate Platform Support

Developer-centric rather than enterprise-centric platforms

- We currently have support for:
  - Microsoft® Windows® 7
  - RedHat Enterprise Linux 6 64-bit
  - Ubuntu 12.04.4 LTS
  - Mac OS X
- We recognise the importance of platforms and we'll be looking at support for them during the beta, such as:
  - Ubuntu 14.04 LTS
  - Suse Linux
  - ...

z/OS, IBM i, Solaris®, AIX or HP-UX® are <u>not</u> our target platforms

Agenda

- Introduction & Background
- Use-case and features
- A new messaging API
- Deploying apps & IBM BlueMix
- Demo

Simple API

Better suited to rapid application development

Basic administration built into the API

- Creating endpoints, configuring connectivity to existing systems etc.

Based on AMQP 1.0

- Open protocol to encourage community development
- Intention is to open-source the clients (statement of direction)

- Language-specific libraries will be provided so the developer doesn't code directly to the AMQP spec

    - Node.js client currently available at npmjs.org

Designed to fit in with what developers are used to

- How quickly and easily can client libraries be added to your runtime of choice?

- Libraries available via appropriate repositories e.g. *npm install mqlight*

Current and considered API features

- Message content
  - **Text** and **Binary** message types
  - Message properties
- Message delivery
  - **TTL on messages**
  - **TTL on destinations**
  - Delivery delay
  - **Read-ahead buffer**
  - Require-subscriber (fail publish if no subscribers)
- Quality of service
  - **At most once** message delivery
  - **At least once** message delivery
  - **Acknowledgement of messages in client or application**

(**Bold** = available in MQ Light today!)

Node.js API design

- Uses callbacks on most API functions, e.g.

```
client.start(function(err) {
  If (err) {
    console.log(“Connection failed!”);
  }
});
```

- Event/state-change callbacks

```
client.on('started', function() {
  console.log("Connection made”);
  console.log("About to published our first message...”);
  …
});
```

- No plans to use Promises in the API
  - Better suited to callbacks which are only called once

Node.js API design

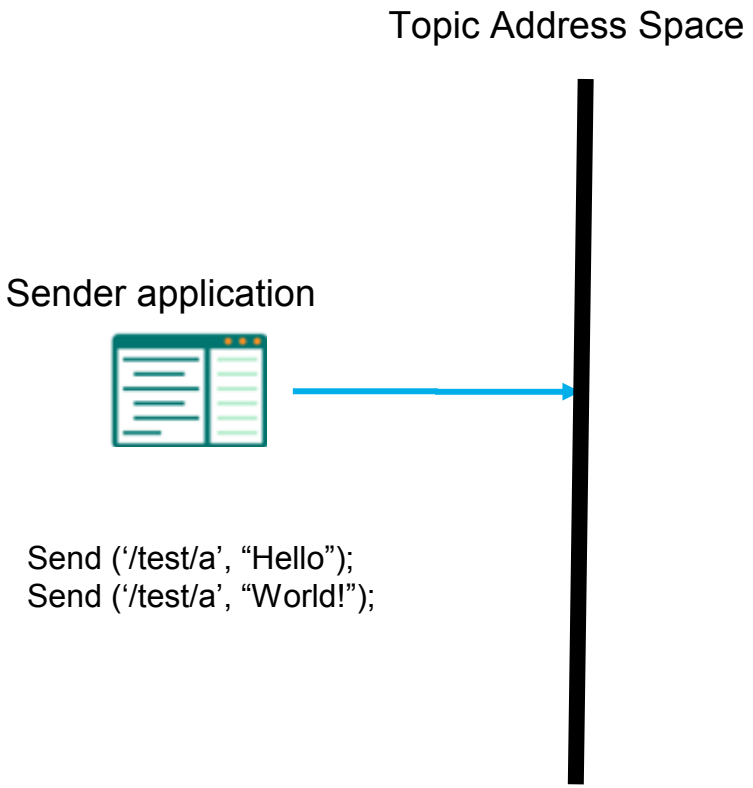
- Whatever object is passed into the send is received by the subscriber

```
client.send(topic, "Hello World");        // Receiver gets a string
client.send(topic, new Buffer([12,32]);  // Receiver gets a buffer
client.send(topic, {color:'red',size:'10'});  // Receiver gets JSON
```

- Many function parameters optional

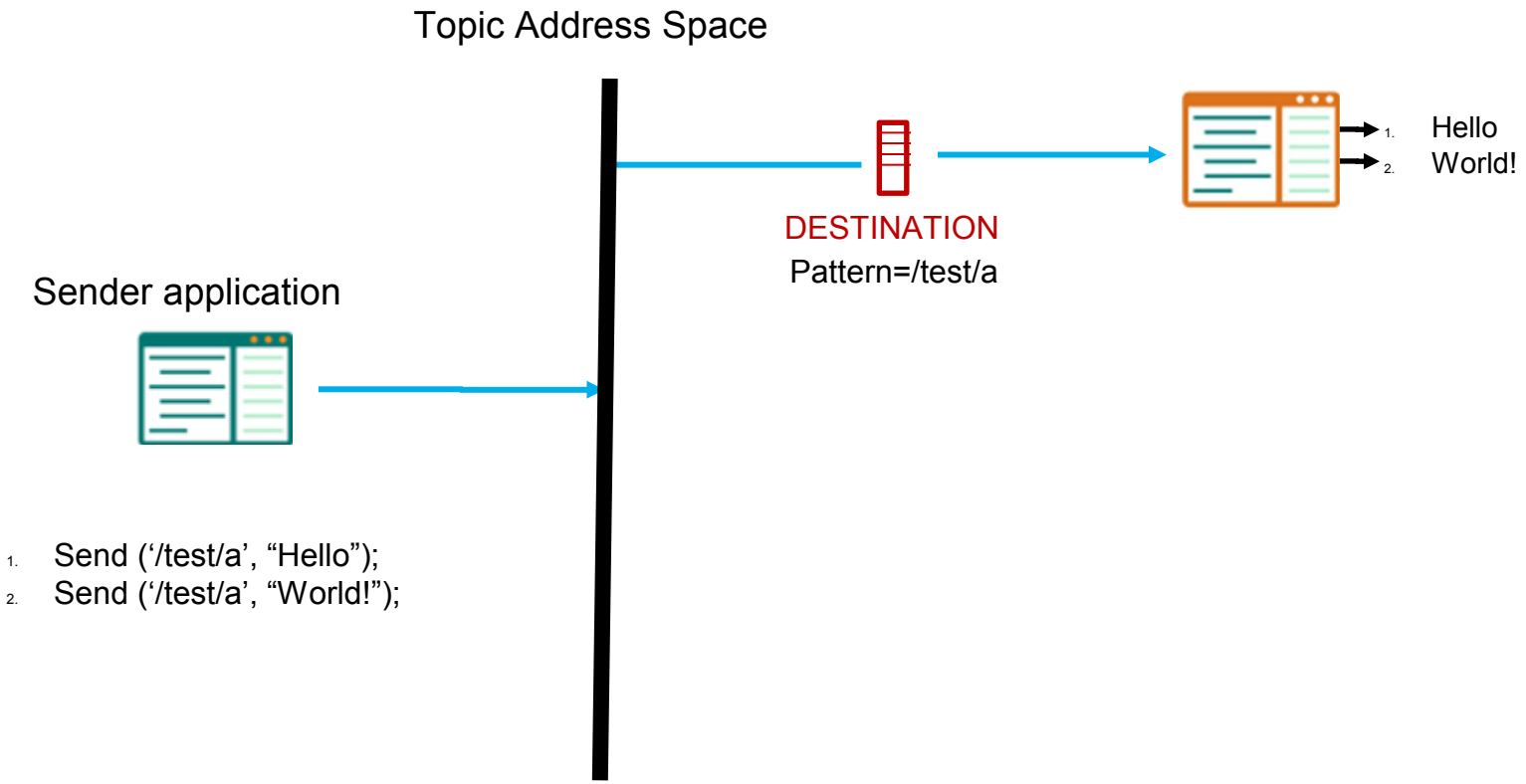
```
// Simple publish
client.send("sports/football", "Arsenal 1, ManU 0");

// Publish with some options
client.send("sports/football", "Arsenal 1, ManU 0",
           {deliveryDelay:1000, ttl:100});

// Publish with options and a callback
client.send("sports/football", "Arsenal 1, ManU 0",
           {deliveryDelay:1000, ttl:100},
           function(err) {  // Handle failed send });
```

MQ Light Messaging Model – Send Messages



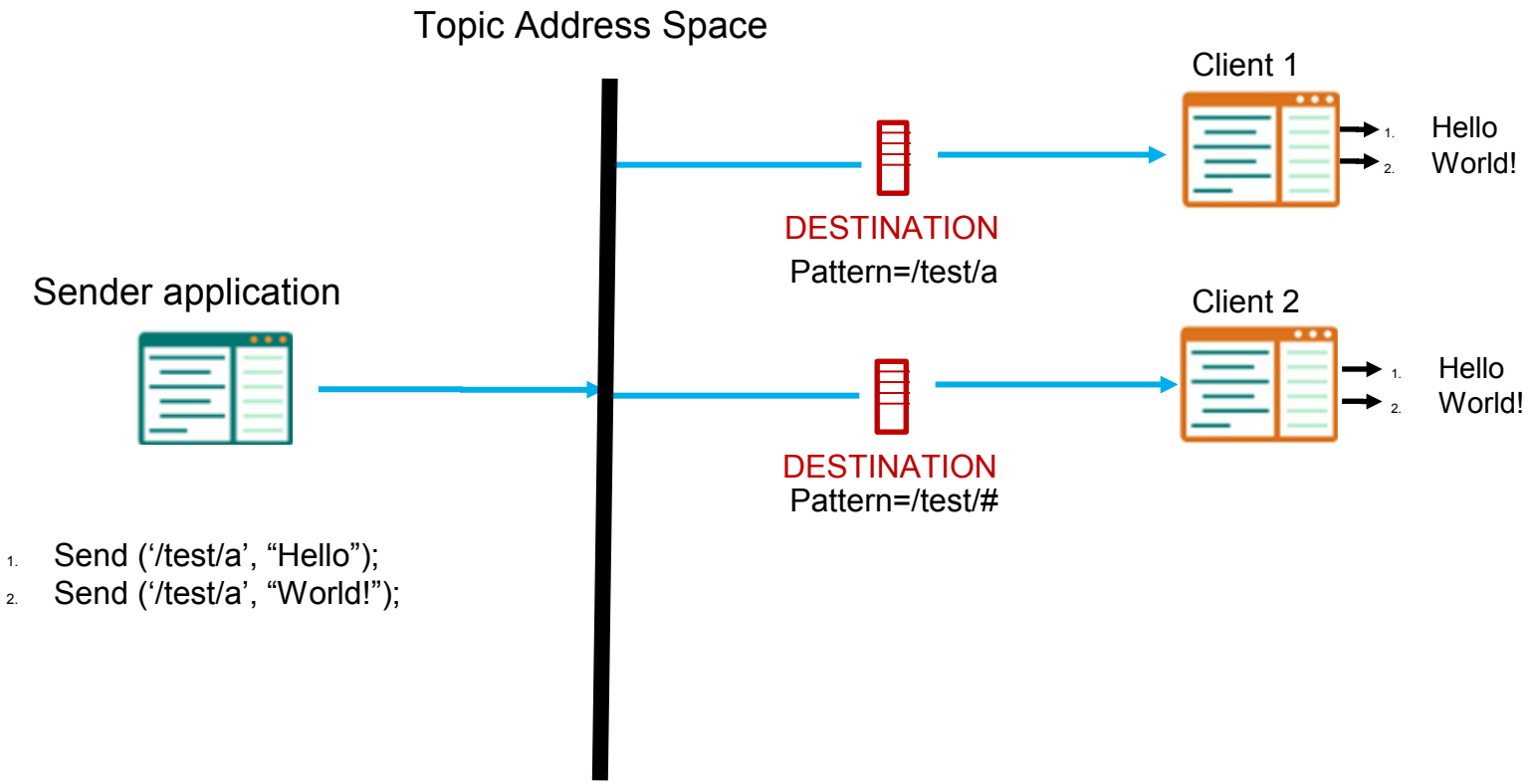
Applications send messages to a ***topic***.
A topic is an address in the topic space
either flat or arranged hierarchically.

MQ Light Messaging Model – Simple Receive



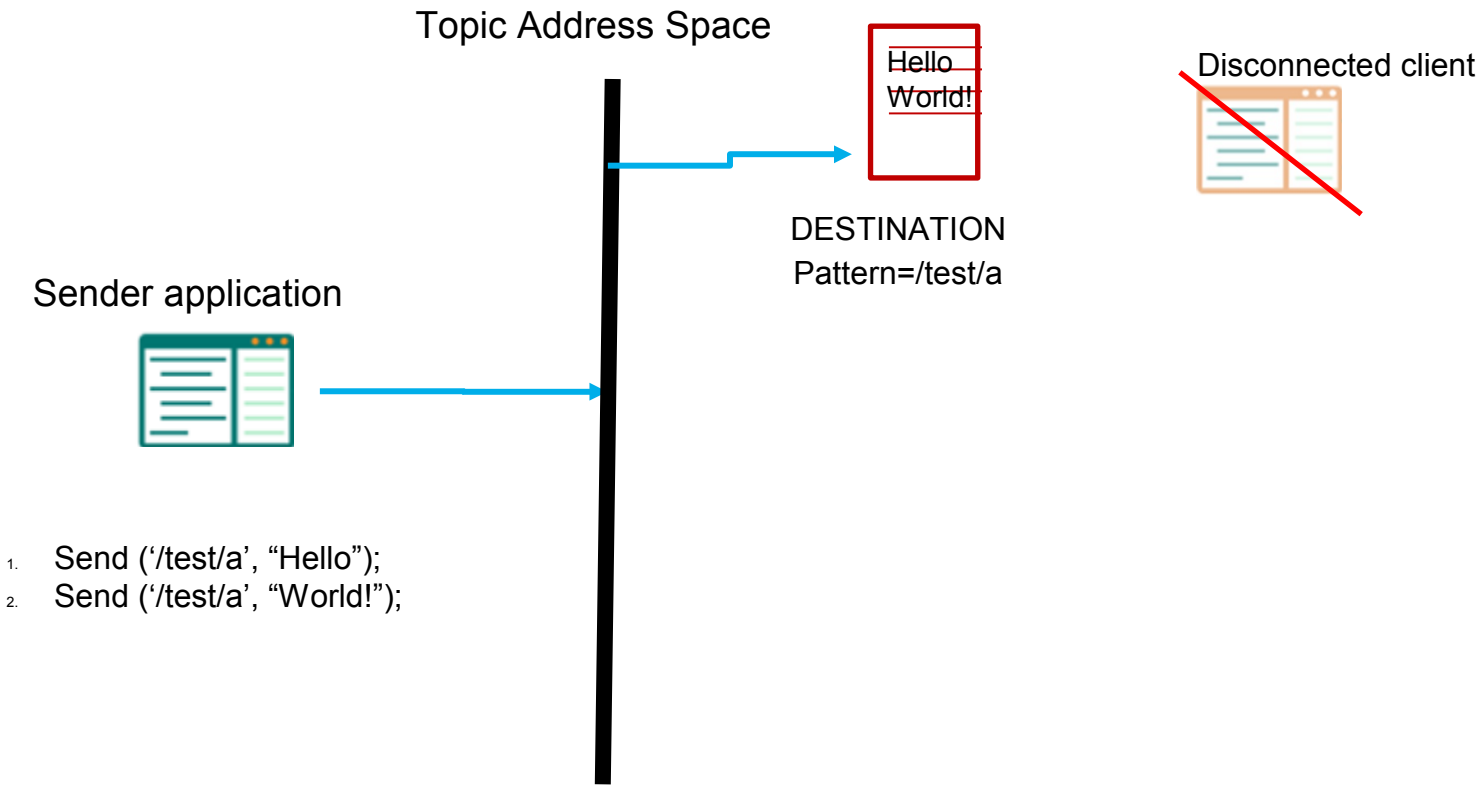
- Applications receive messages by creating a **destination** with a pattern which matches the topics they are interested in.
- Pattern matching scheme based on WMQ.

MQ Light Messaging Model – Pub/Sub



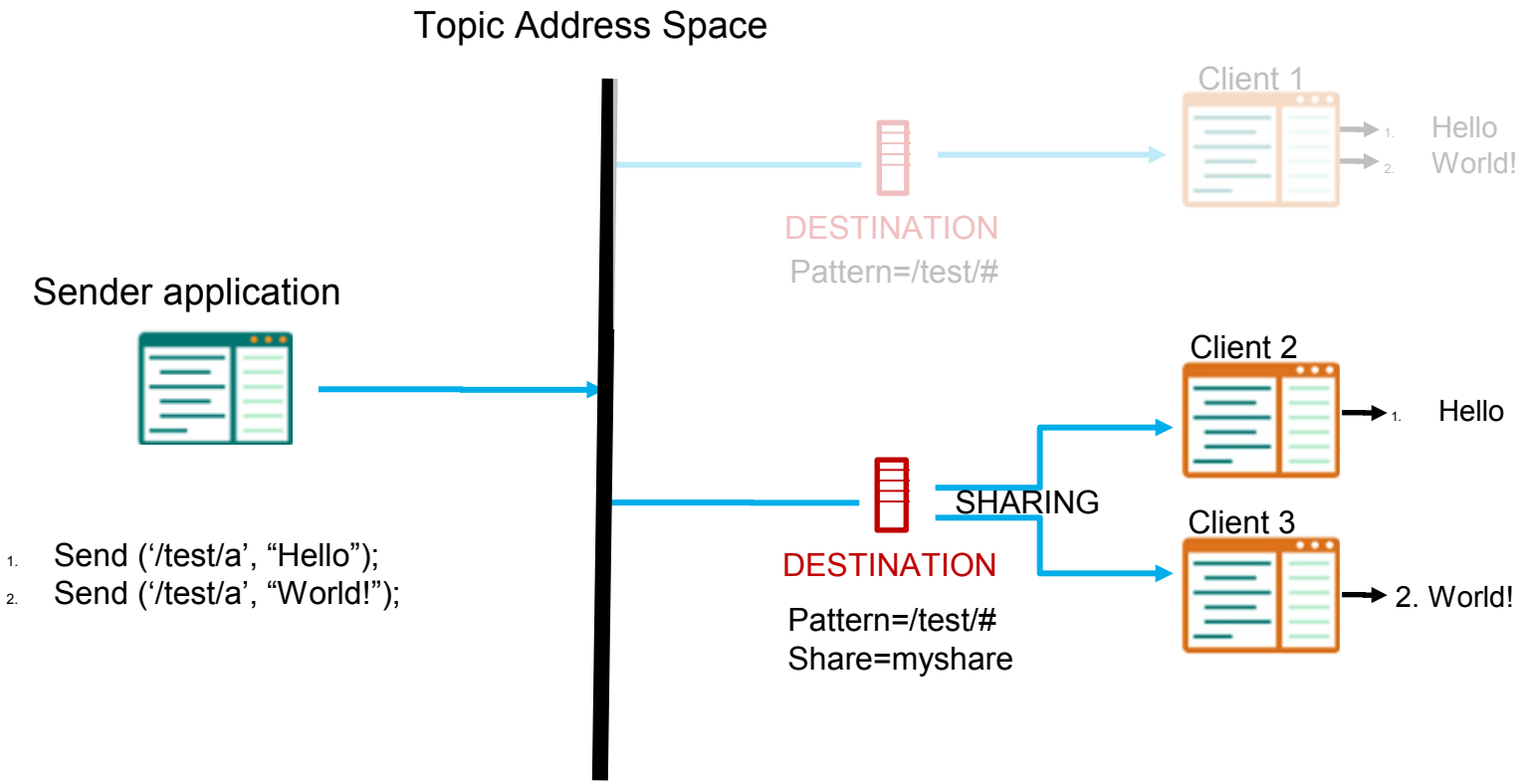
- Multiple destinations can be created which match the same topic
  - Pub/Sub style.

MQ Light Messaging Model – Persistent destinations



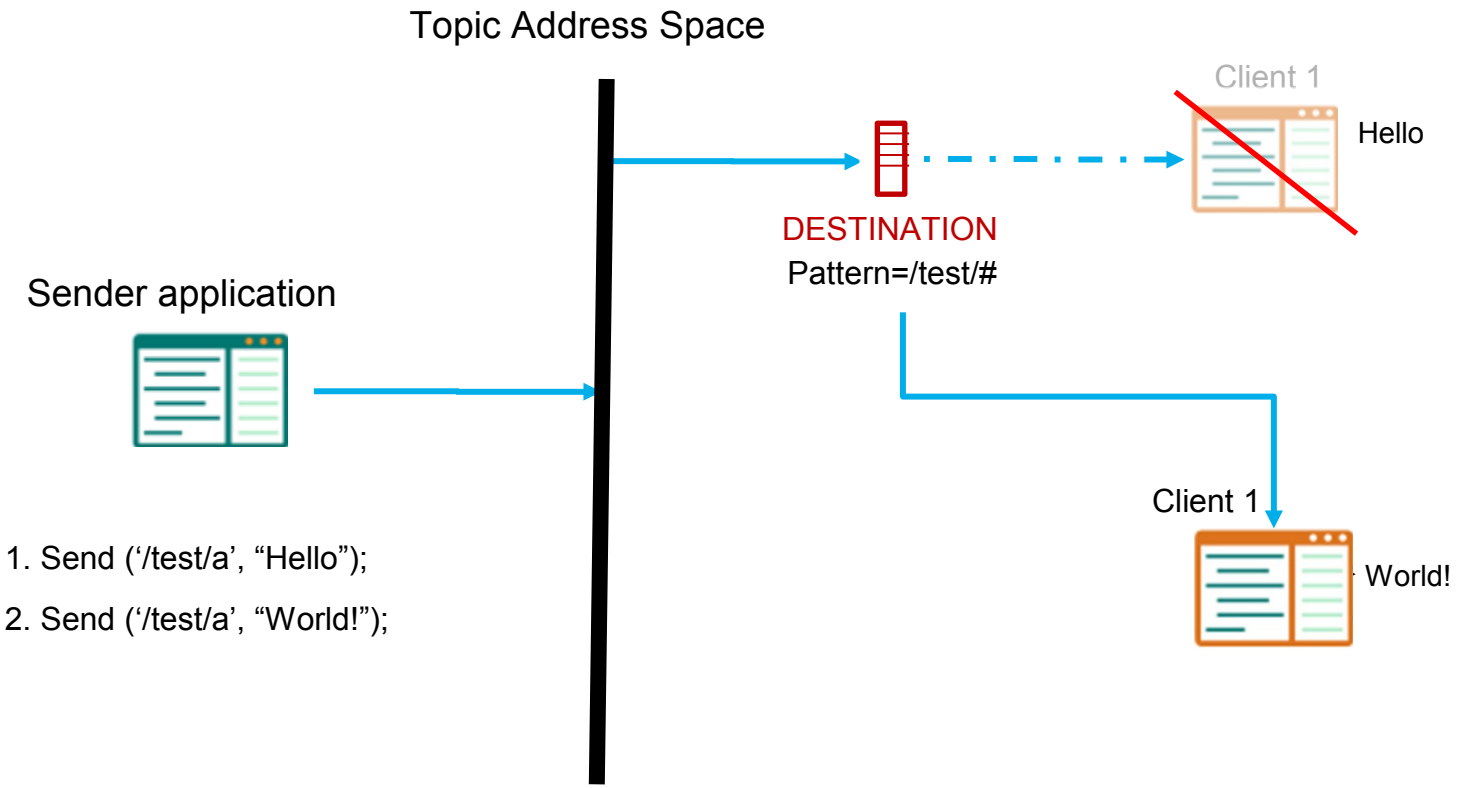
- Destinations persist for a defined "time to live" after receiver detaches.

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 specify (optional) client ID.
- Re-using the same client ID pre-empts the original connection.
  - Ideal for worker takeover in the cloud.

Agenda

- Introduction & Background
- Use-case and features
- A new messaging API
- Deploying apps & IBM BlueMix
- Demo

Application Messaging Deployment Options

IBM MQ Light Runtime

Tooling for developer platforms

Write apps in node.js

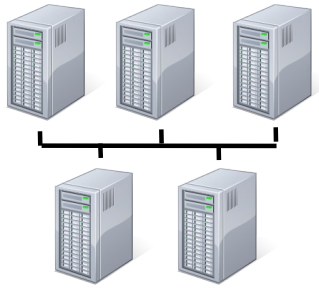
Deploys

Deploys

Deploys

MQ Light Runtime

On-premise application messaging



IBM WebSphere MQ
Enterprise messaging
(statement of direction)

MQ Light Service in
IBM BlueMix

Public cloud messaging

Application Messaging Deployment Options

IBM MQ Light Runtime

Tooling for developer platforms

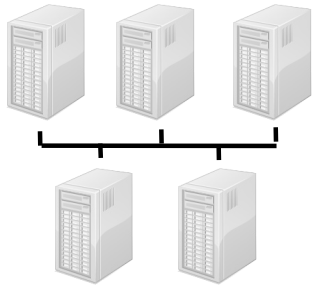
Write apps in node.js

Deploys

MQ Light Runtime

On-premise application messaging

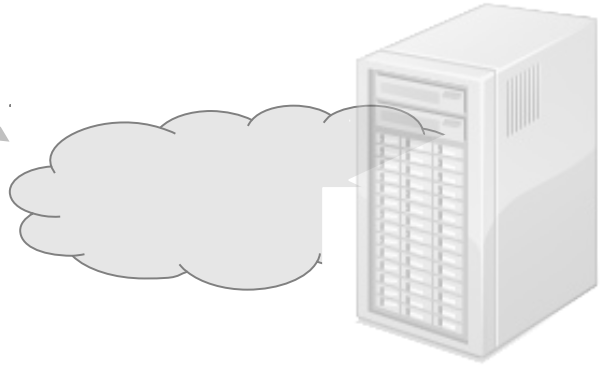
Deploys



IBM WebSphere MQ

Enterprise messaging
(Statement of direction)

Deploys



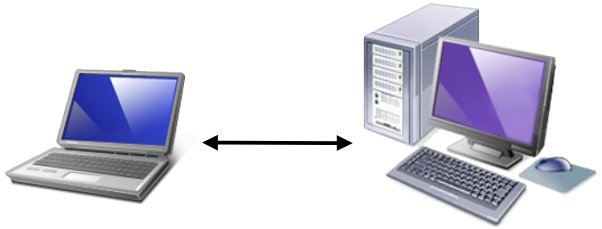
MQ Light Service in IBM BlueMix

Public cloud messaging

IBM MQ Light – for on-premise deployment

Some developers will like the new APIs but won't want to deploy to a PaaS

- Small line-of-business projects, e.g. queueing up builds
- No mission critical features
  - No HA
  - No horizontal scaling



MQ Light is available to license as the production runtime for messaging applications

- Specific licensing plan not yet finalised
- Looking at licensing models that work for development teams

Fully compatible with the IBM MQ Light service in BlueMix

- Easy to migrate applications to a PaaS at a later point

Application Messaging Deployment Options

IBM MQ Light Runtime

Tooling for developer platforms

Write apps in node.js

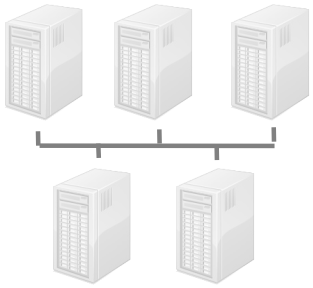
Deploys

Deploys

Deploys

MQ Light Runtime

On-premise application messaging



IBM WebSphere MQ

Enterprise messaging
(statement of direction)

MQ Light Service in
IBM BlueMix

Public cloud messaging

IBM MQ Light Service - Messaging in a PaaS

As well as the standalone MQ Light product we also have a version of MQ Light running in the IBM BlueMix PaaS

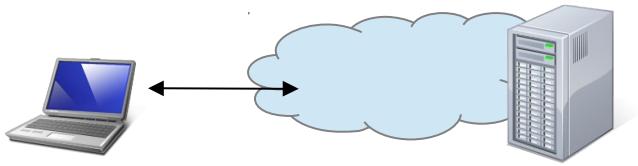
- Called the MQ Light Service
- Supports the MQ Light API and JMS
- Admin-free experience
- MQ Light environment automatically provisioned for each user when they request an instance of it
- Applications are linked or 'bound' to the provisioned environment automatically when they are deployed to BlueMix
- **Not** a full MQ experience in the cloud

How does MQ Light fit in PaaS development?

The intention is for MQ Light to provide a development environment for applications that will deploy against IBM MQ Light in BlueMix

Gives the developer 2 choices for development:

1. Develop directly in BlueMix. As you make changes to your code you push the new version straight to BlueMix and check your changes are OK

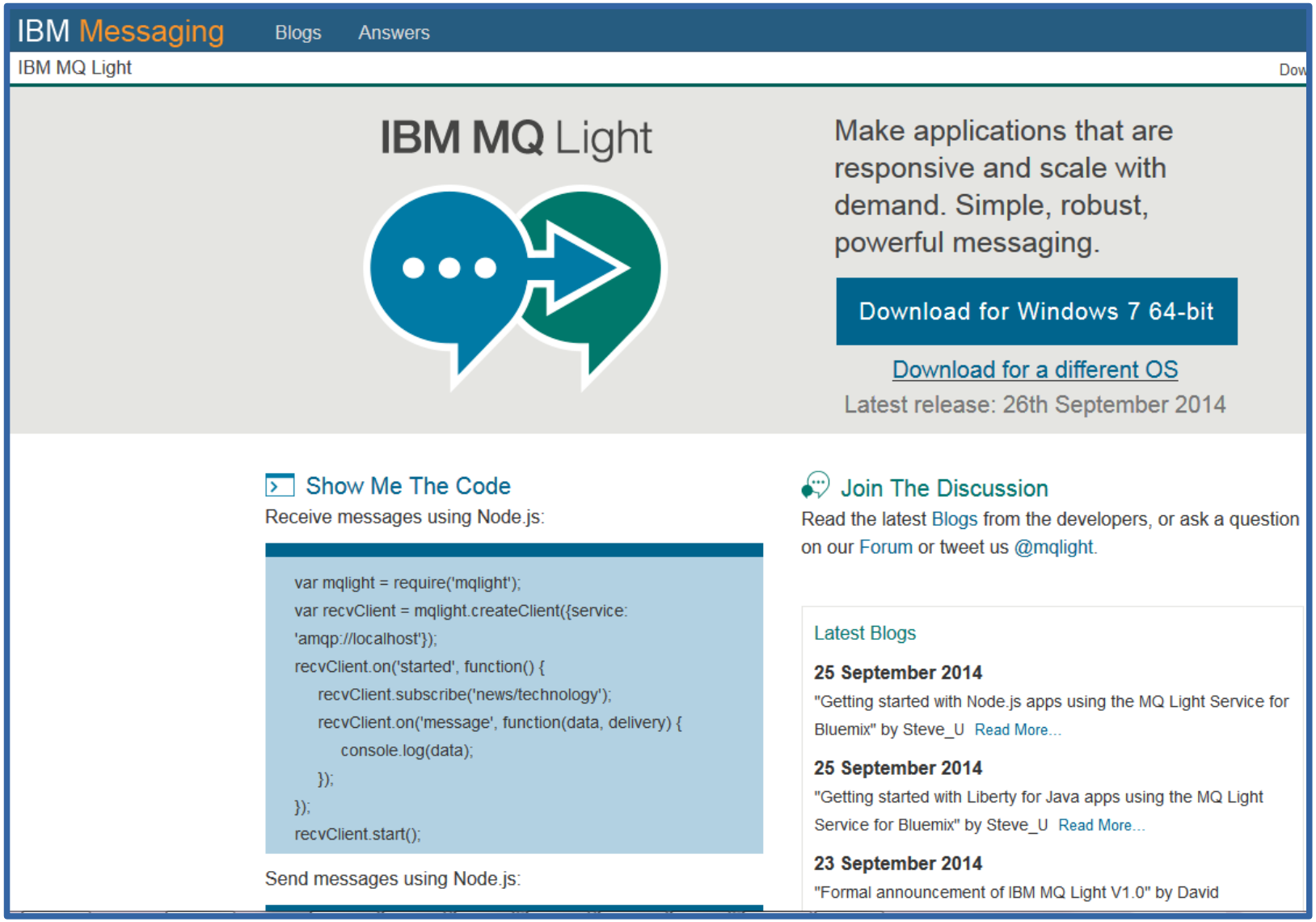


2. Develop on your laptop using MQ Light. Once you're 90% complete, start pushing your app to BlueMix to try it out in the cloud and iron out any remaining bugs.

MQ Light offers an offline development environment for the MQ Light API (not JMS)

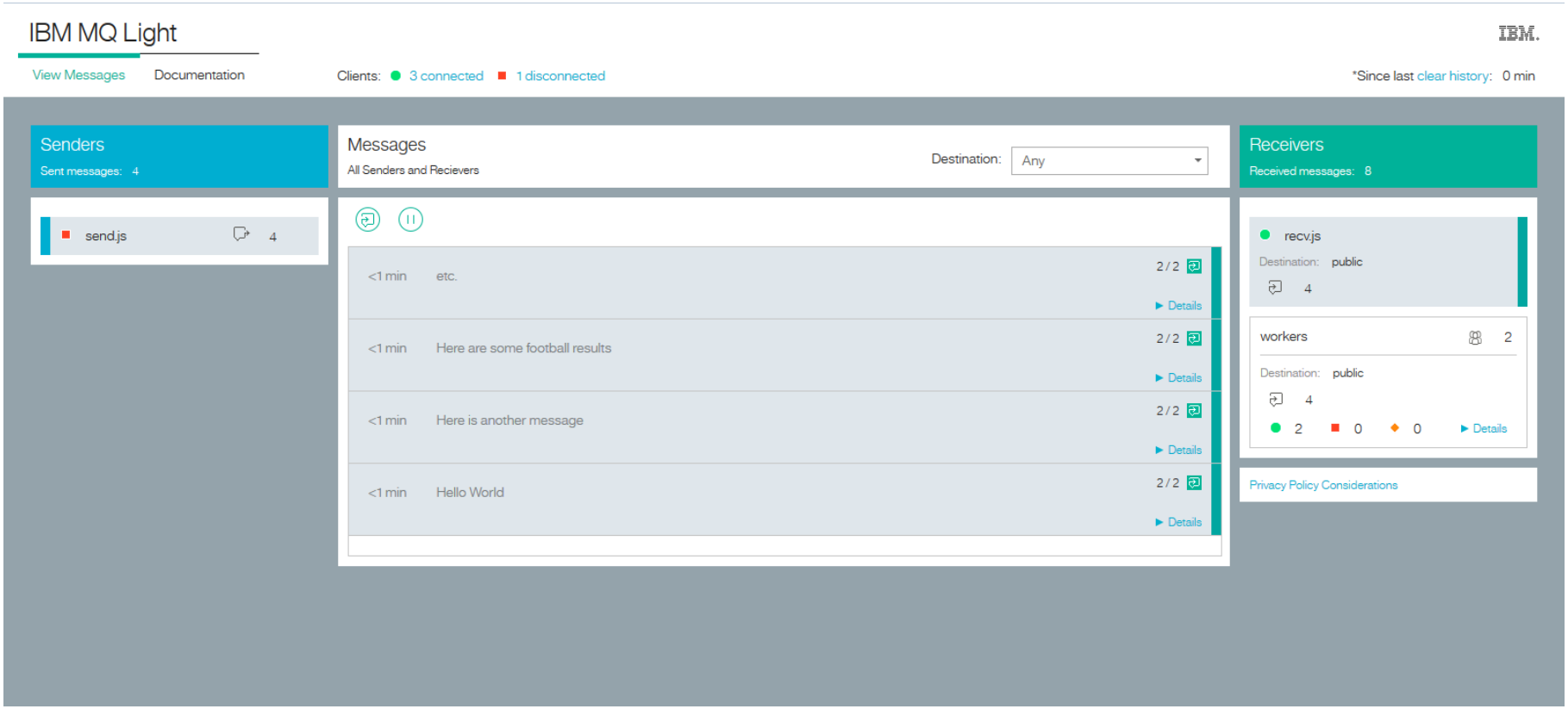
- Provides a development version of the messaging service that will be offered in the cloud
- Makes offline application development possible when connectivity to the cloud is not available
- Comparatively small download & install footprint to get you started quickly & easily

Download & Install Experience

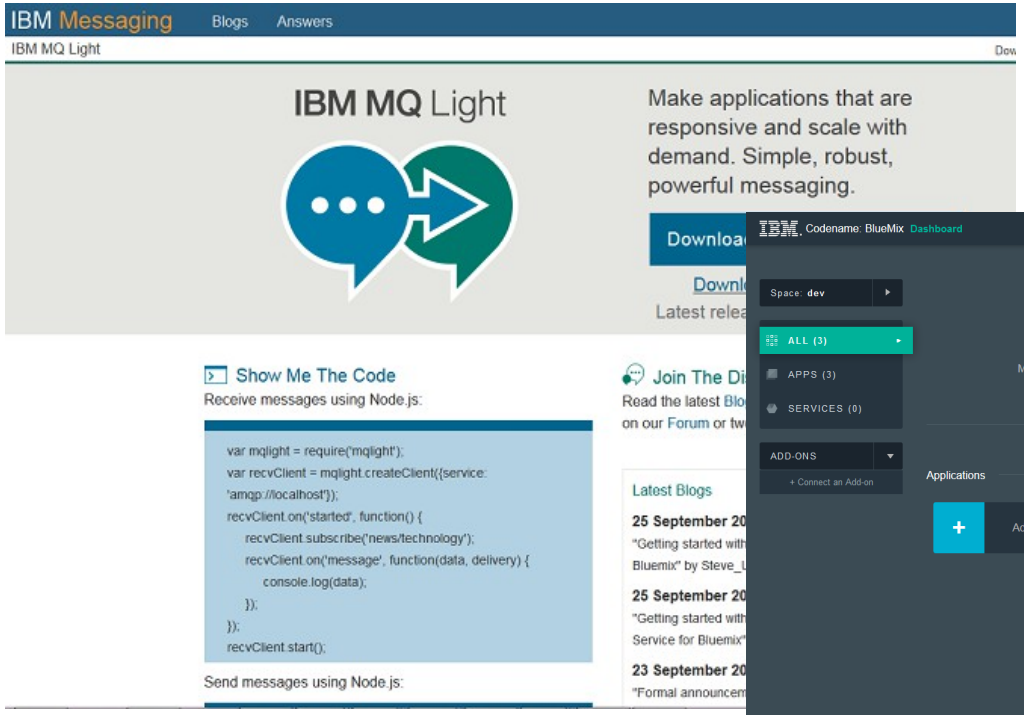


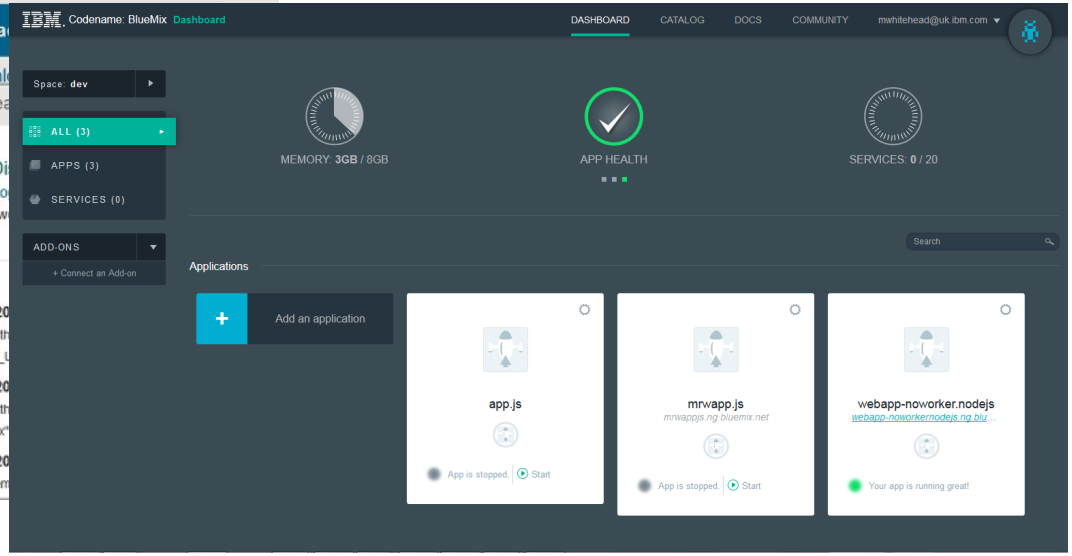
www.ibmdw.net/messaging/mq-light/

Web Based Tooling



Try it all out for yourselves!





MQ Light is available now at https://www.ibmdw.net/messaging/mq-light/ or Google “IBM MQ Light”

MQ Light Service is available in IBM BlueMix today at http://bluemix.net or Google “IBM BlueMix MQ Light”

Questions?

Legal Disclaimer