



MQ Light

Carl Farkas

IBM Europe zWebSphere consultant

farkas@fr.ibm.com

*(based upon charts from Alan Chatt, MQ Light
Product Manager)*



Agenda

- ▶ Application Messaging
- ▶ The journey that got us here...
- ▶ What does MQ Light allow application developers to do?
- ▶ What does MQ Light look like now?



IBM Messaging portfolio



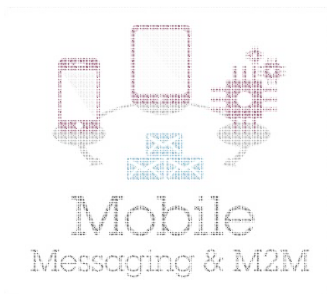
Enable developers to build more scalable, responsive applications

Focus on application developer use cases, breadth of language support, no administration, ease-of-deployment, lightweight & powerful API, as software or a cloud service



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, security and privacy, feed into real-time analytics, location-based notifications



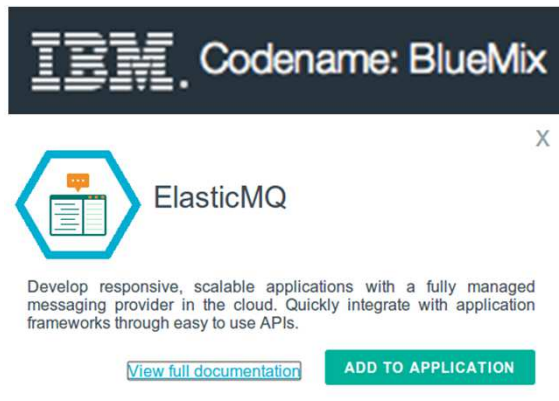
What is MQ Light and Application Messaging?

Software



- ▶ Messaging for application developers to help create responsive applications that scale easily
- ▶ Trivially easy to get started; no setup, no configuration, no administration

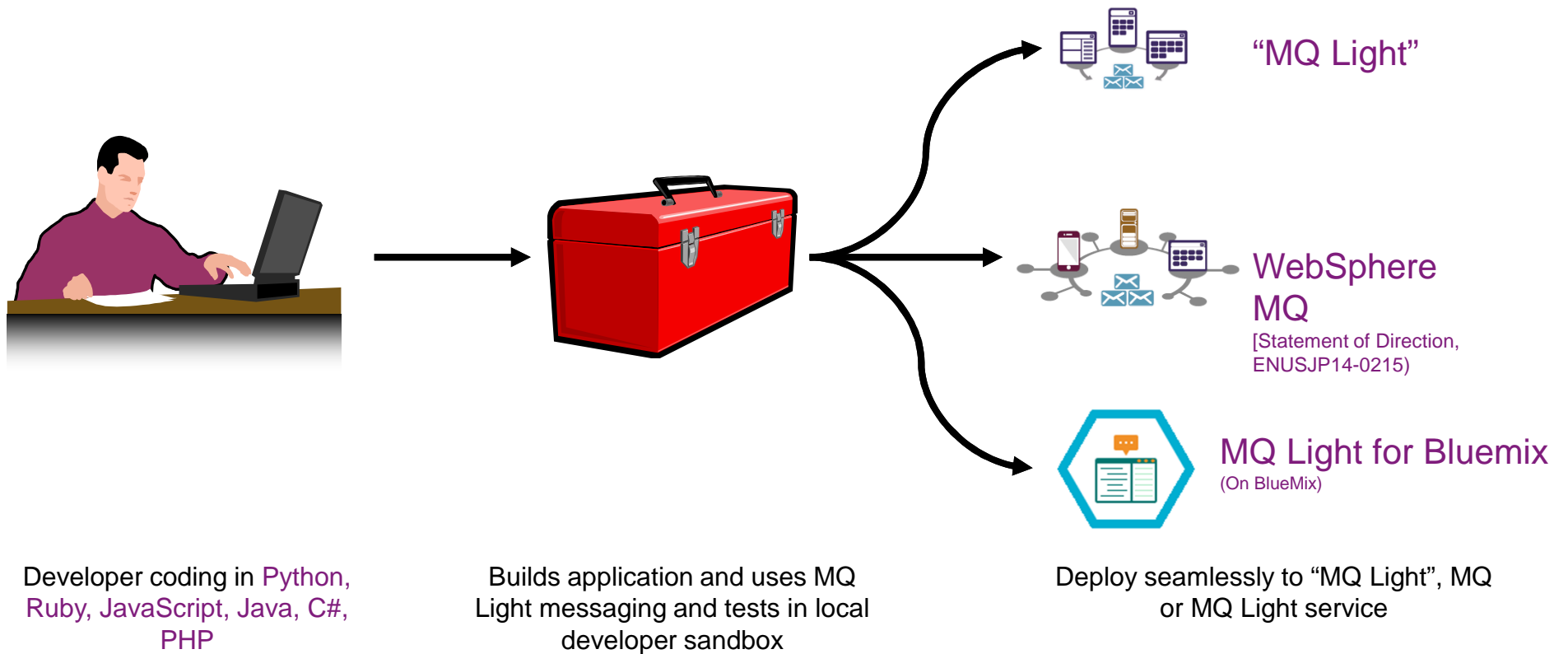
Cloud Service



- ▶ Available as software download or full cloud service in Codename : BlueMix
- ▶ APIs crafted specifically for each language
- ▶ Tooling that supports app development



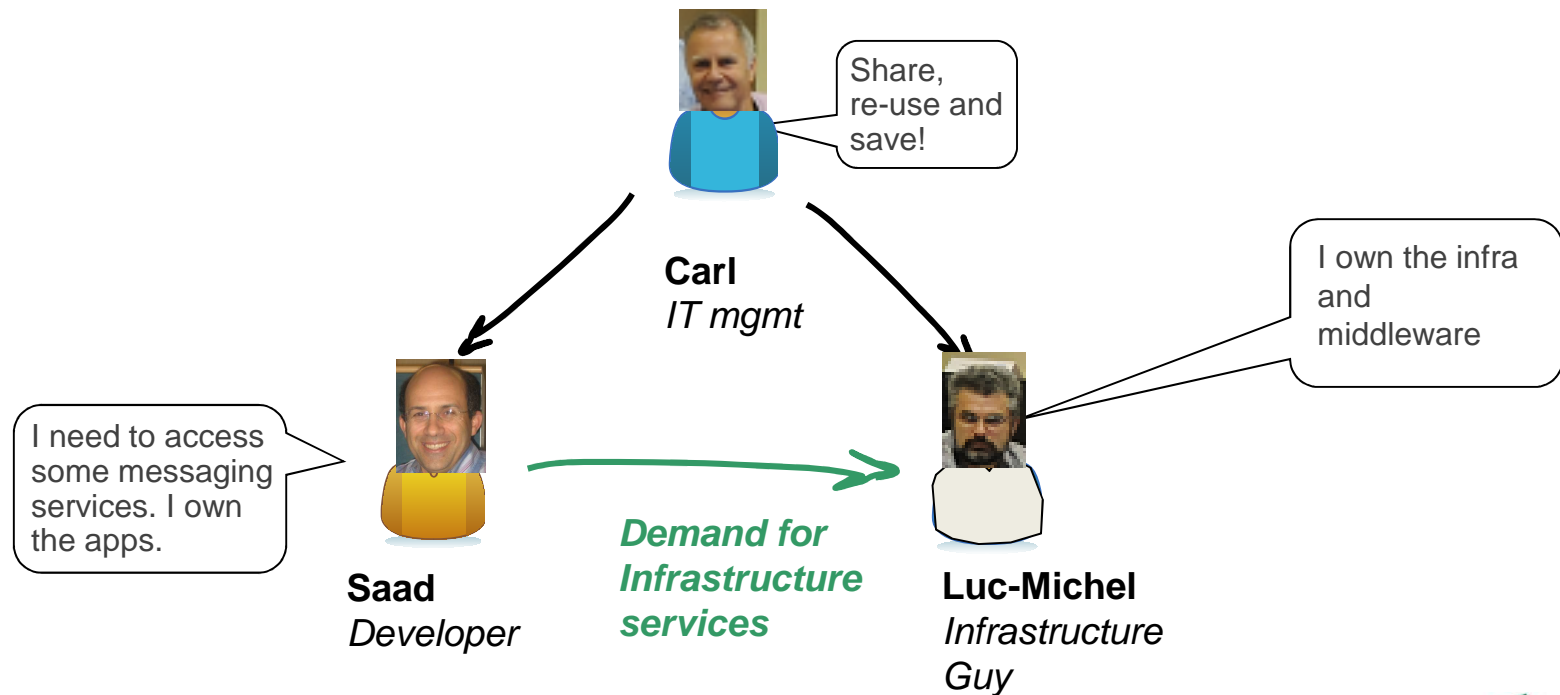
Application messaging deployment options



The journey that got us here...

► Previous developer / IT relationship

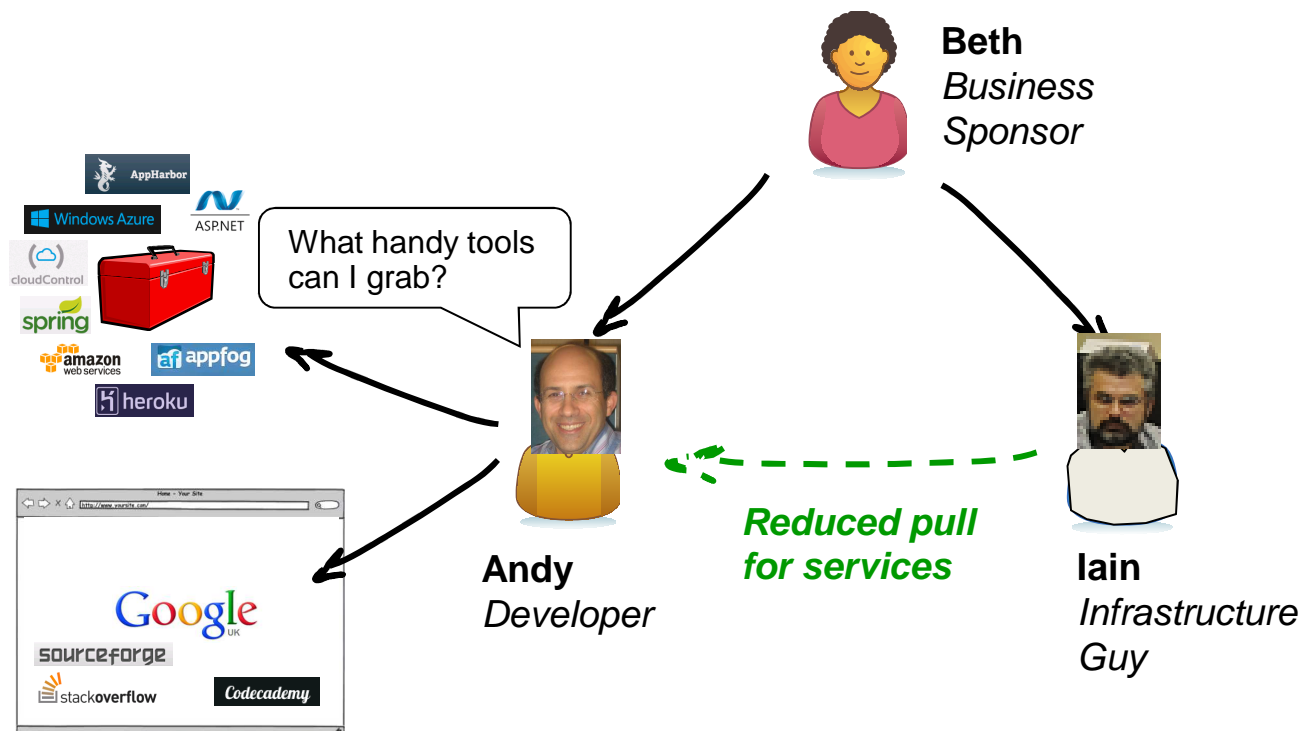
- Driven by centrally controlled common standards
- Carefully planned projects delivering core business systems
- Focused on re-using existing skills and investments



The journey that got us here...

► Evolving developer / IT relationship

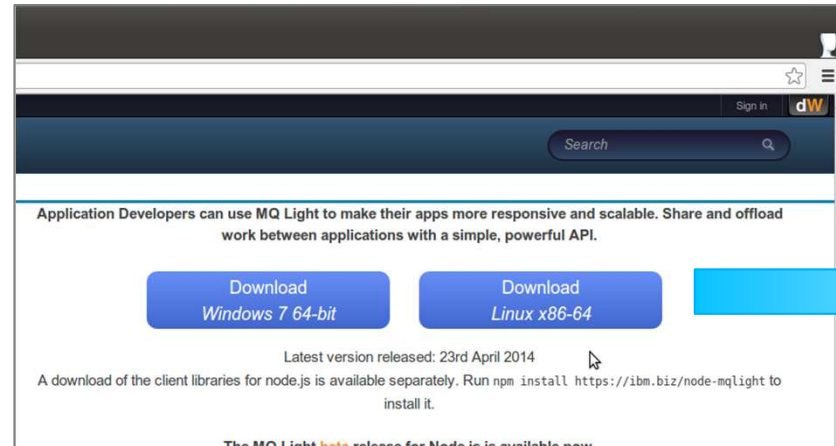
- Driven by business sponsor demands
- Developers download and use the tools to get the job done
 - Easy, cheap access today
 - Pressure from users & management to quickly provide apps
- Focused on trying new apps and concepts in the market



The journey that got us here...



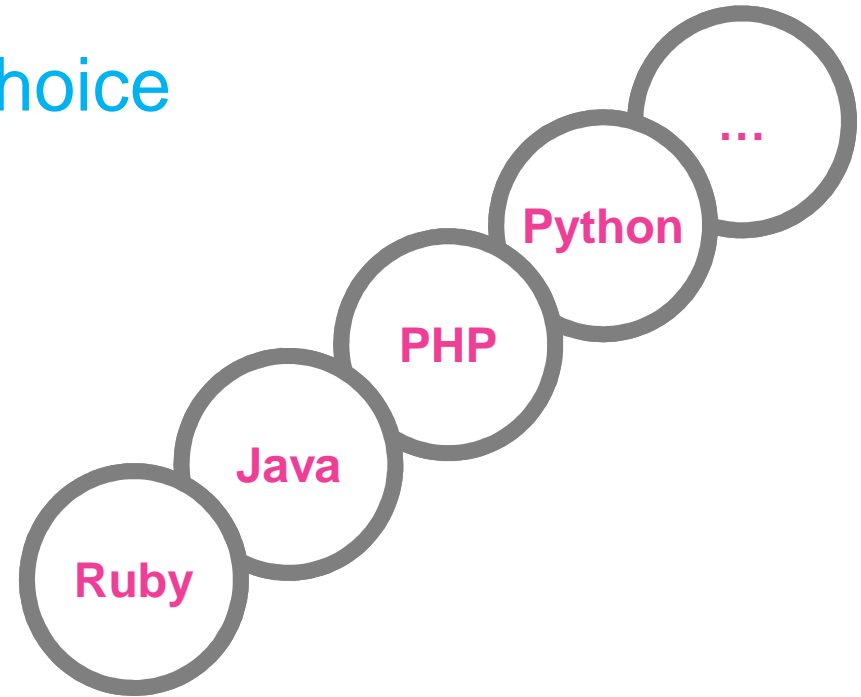
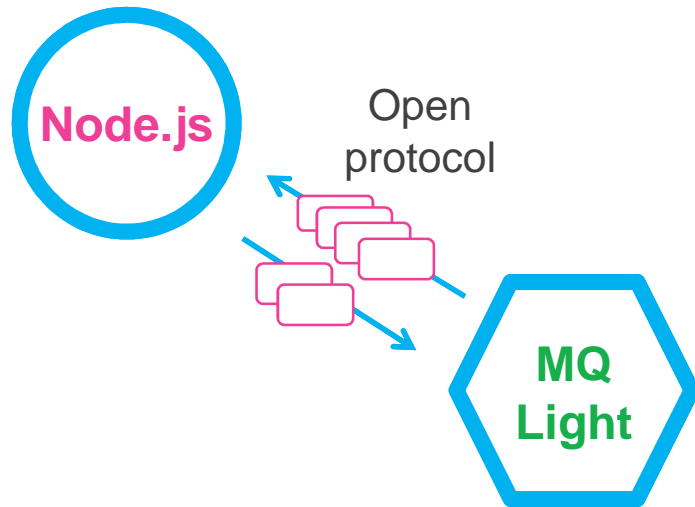
Easy to get going



- ▶ Download and get coding within 5 minutes
- ▶ No administration; just code and go
- ▶ Node API client libraries installed using npm package manager
- ▶ Tutorials and examples in their languages, relevant to actual use



Does not restrict language choice



- ▶ Polyglot language
- ▶ API crafted to feel native within the language
- ▶ New clients can be created by developers



Tooling to assist development

- ▶ Used to answer questions like :
 - Where did this message come from?
 - Where did that message go?
 - What messages has my app consumed?

The screenshot displays the IBM MQ Light web console interface. The top navigation bar includes the IBM logo, a 'View Messages' link, a 'Documentation' link, and a status indicator showing 'Clients: 8 connected' and '0 disconnected'. The main content area is divided into three panels: 'Senders', 'Messages', and 'Receivers'. The 'Senders' panel on the left lists five workers with their respective message counts: worker5 (91), worker4 (96), worker3 (120), worker2 (143), and worker7 (75). The 'Messages' panel in the center shows a list of messages with details such as time, content, and destination. The 'Receivers' panel on the right shows a list of receivers with their respective message counts: tweetsToProcess (787), worker5 (92), worker4 (96), and worker3 (120).

Panel	Item	Count
Senders	worker5	91
	worker4	96
	worker3	120
	worker2	143
	worker7	75
Receivers	tweetsToProcess	787
	worker5	92
	worker4	96
	worker3	120

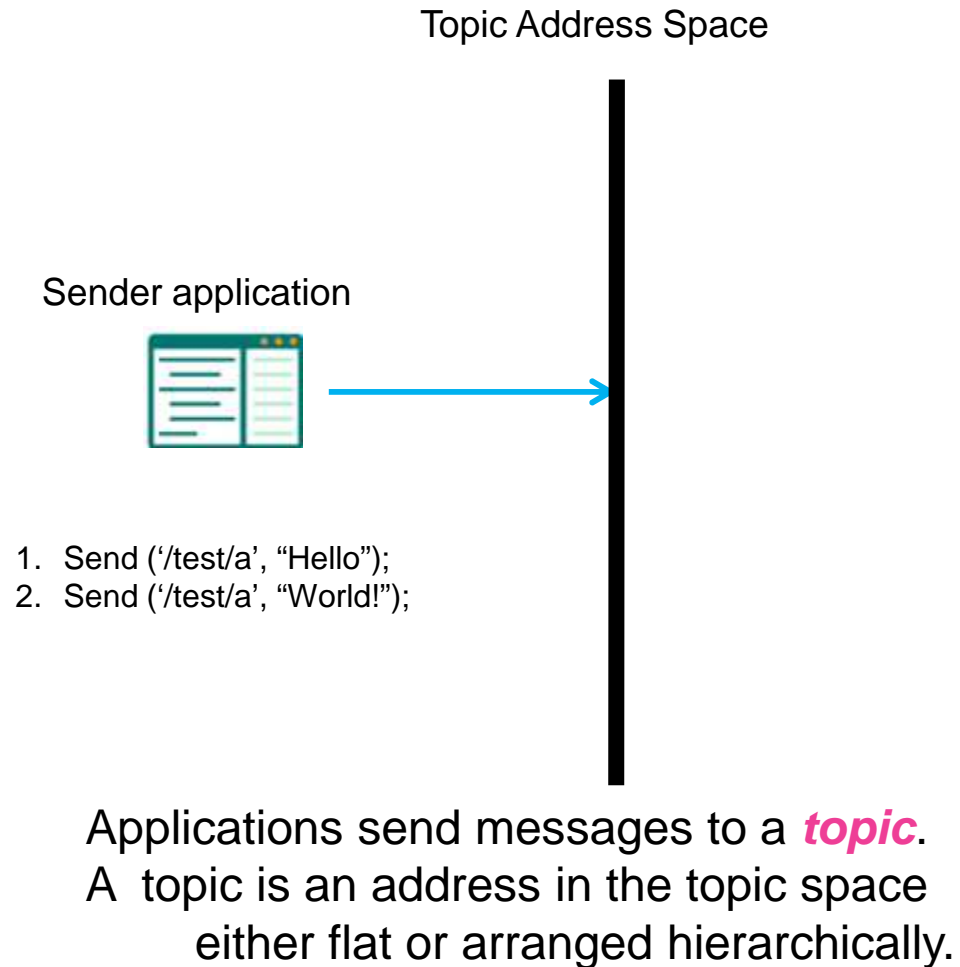


Open standards and community

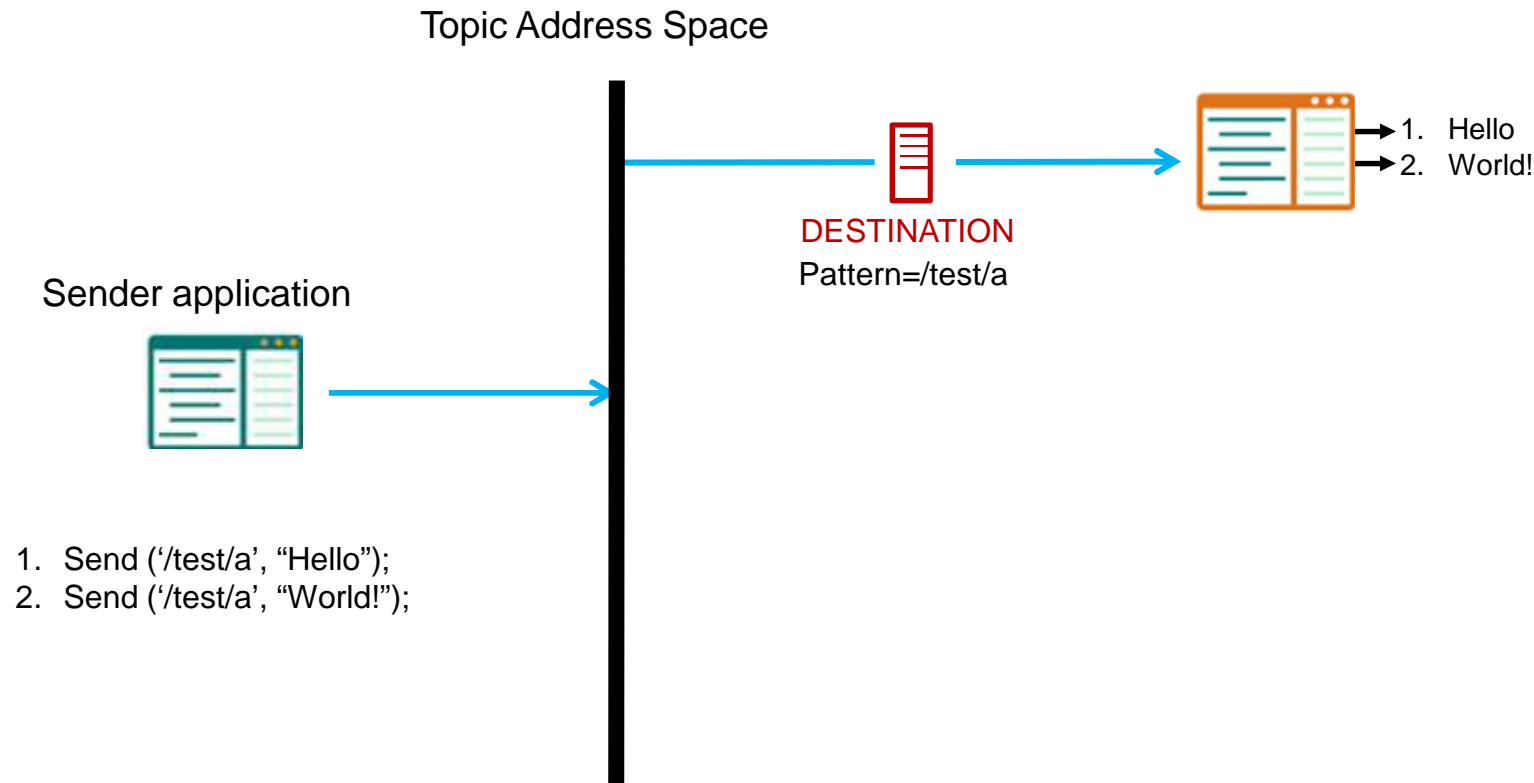
- ▶ Using open protocol to build community around messaging clients
- ▶ Based on Apache Proton OSS project using AMQP1.0
- ▶ Both consuming and contributing code to the proton project



MQ Light Messaging Model – Send Messages



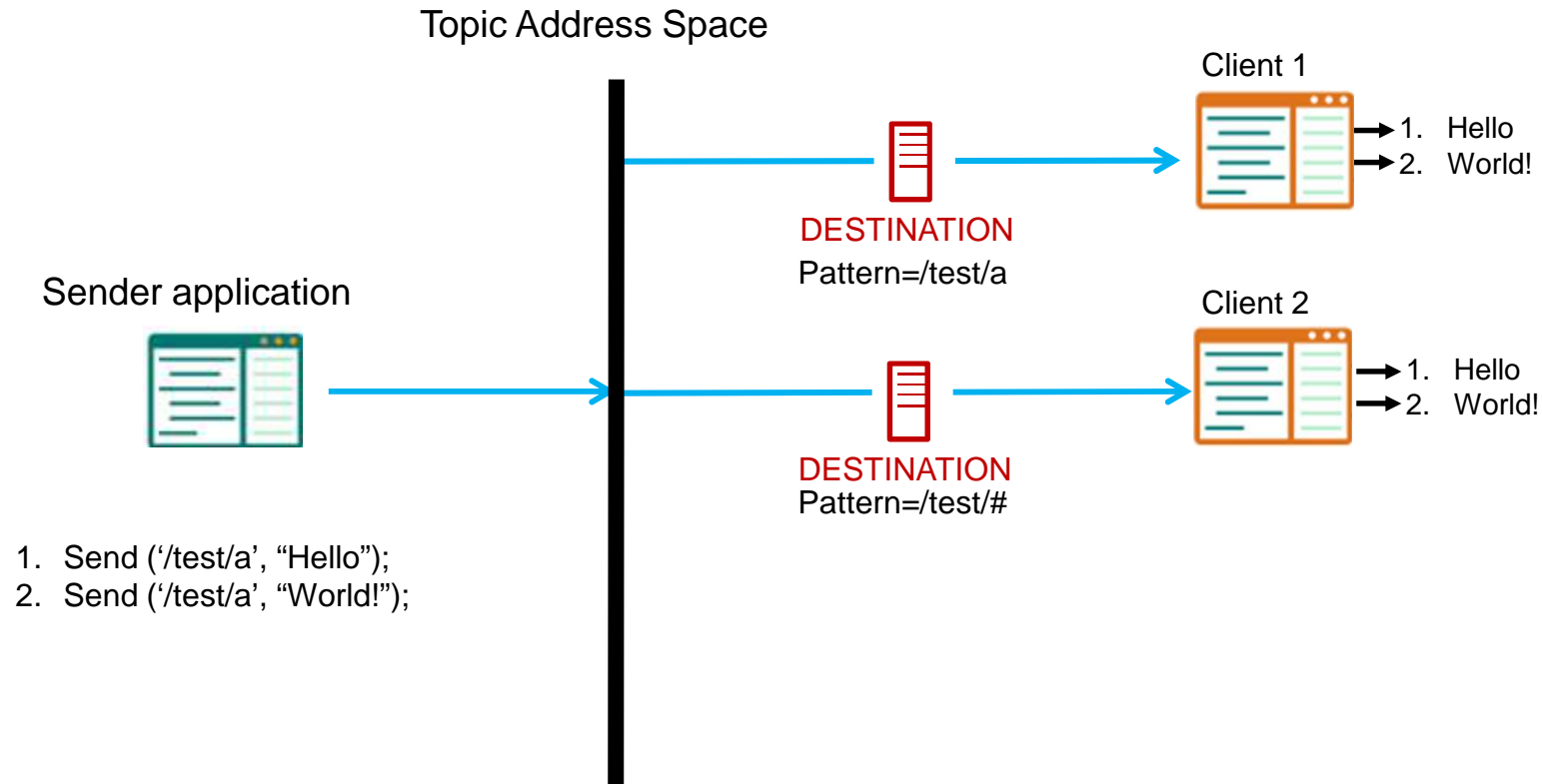
MQ Light Messaging Model – Simple Receive



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



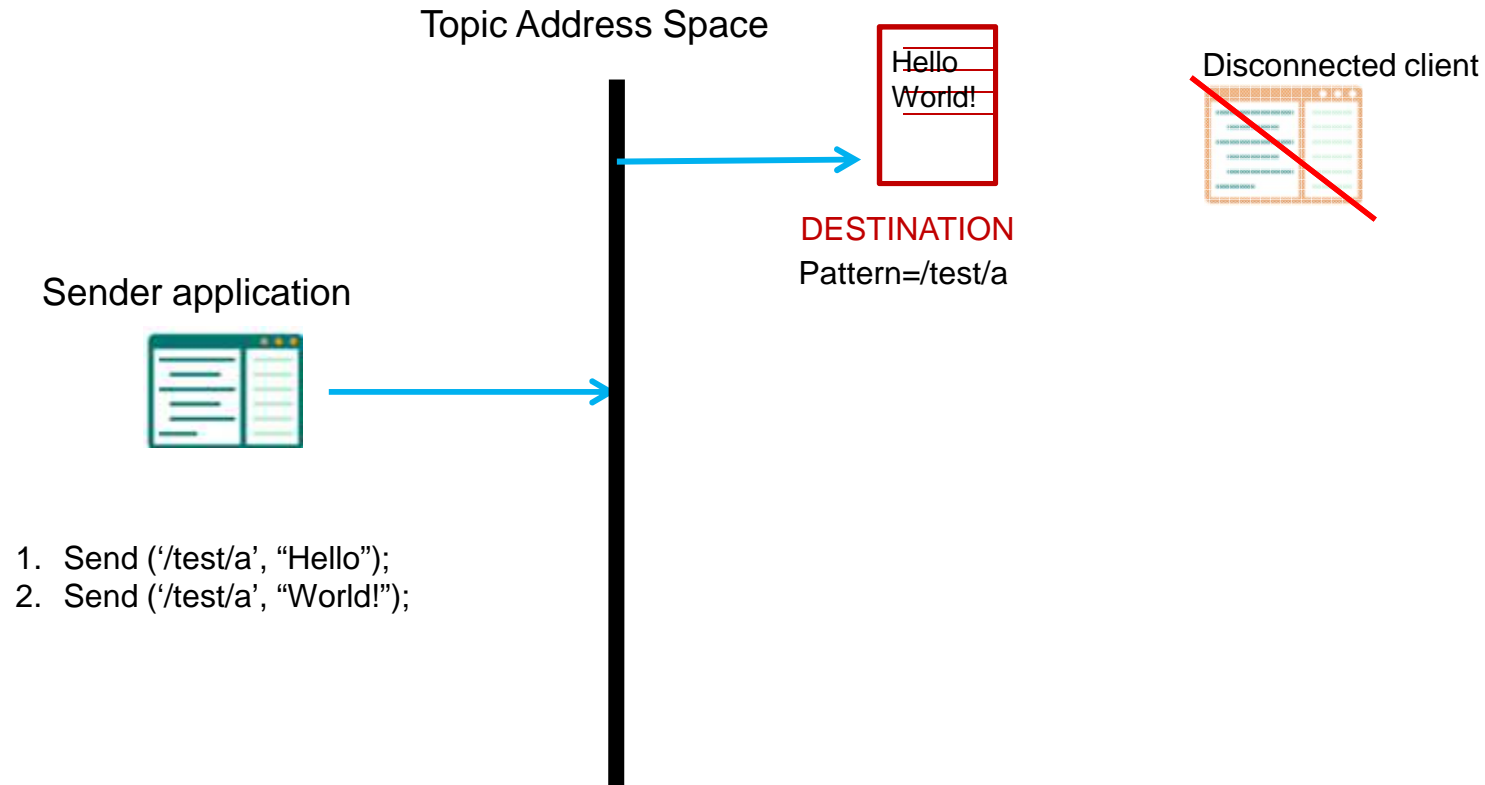
MQ Light Messaging Model – Pub/Sub



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



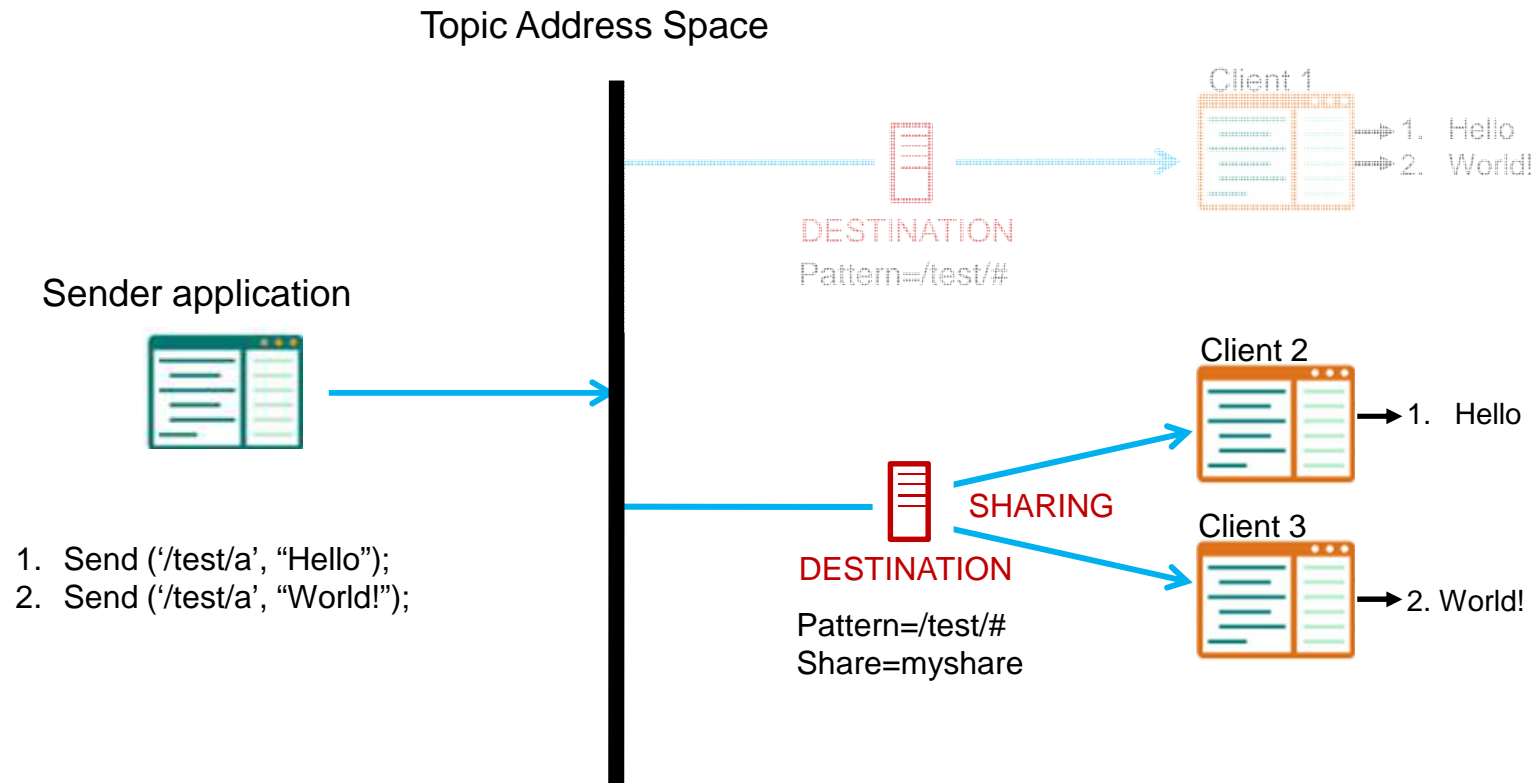
MQ Light Messaging Model – Persistent destinations



- 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.



Where is MQ Light now then (June 2014)?

The screenshot shows the IBM MQ Light website in a Chromium browser. The page features a 'beta' badge and a large image of a box labeled 'work in progress'. The main content area is titled 'Learn about MQ Light' and includes sections for 'How can MQ Light help my applications?' and 'Getting started'. A sidebar on the right contains 'API Samples' for Node.js, Ruby, and Java. The Node.js sample is expanded, showing code for receiving and sending messages.

IBM MQ Light | IBM Messaging - Chromium

https://www.ibmdev.com/websites/messaging/mq-light/

IBM developerWorks

IBM Messaging Blogs Answers

MQ Light **beta**

Application Developers can use MQ Light to work better

Download Windows

A download of the client libraries for no

The MQ Light runtime is now available in the Open Beta program. You can help shape the product.

Learn about MQ Light

How can MQ Light help my applications?

MQ Light provides a configurable messaging runtime, for developers to write apps in just a few minutes.

[Read more...](#)

Getting started

Install the MQ Light beta, and start using the provided samples.

[Read more...](#)

API Samples

Node.js Ruby Java coming soon

Complete sample applications are available in the samples directory of the node module.

Receiving messages – This snippet receives a message asynchronously from a destination called public

```
var mqlight = require('mqlight');
var recvClient = mqlight.createClient({id: 'recv_client_1', service: 'amqp://localhost'});
var address = 'public';
recvClient.on('connected', function() {
  recvClient.subscribe('address');
  recvClient.on('message', function(data, delivery) {
    console.log(data);
  });
});
recvClient.connect();
```

Sending a message – The following snippet sends a message to a destination called public

```
var mqlight = require('mqlight');
var sendClient = mqlight.createClient({id: 'send_client_1', service: 'amqp://localhost'});
var topic = 'public';
sendClient.on('connected', function() {
  sendClient.send(topic, 'Hello World!');
});
sendClient.connect();
```

► Open Beta

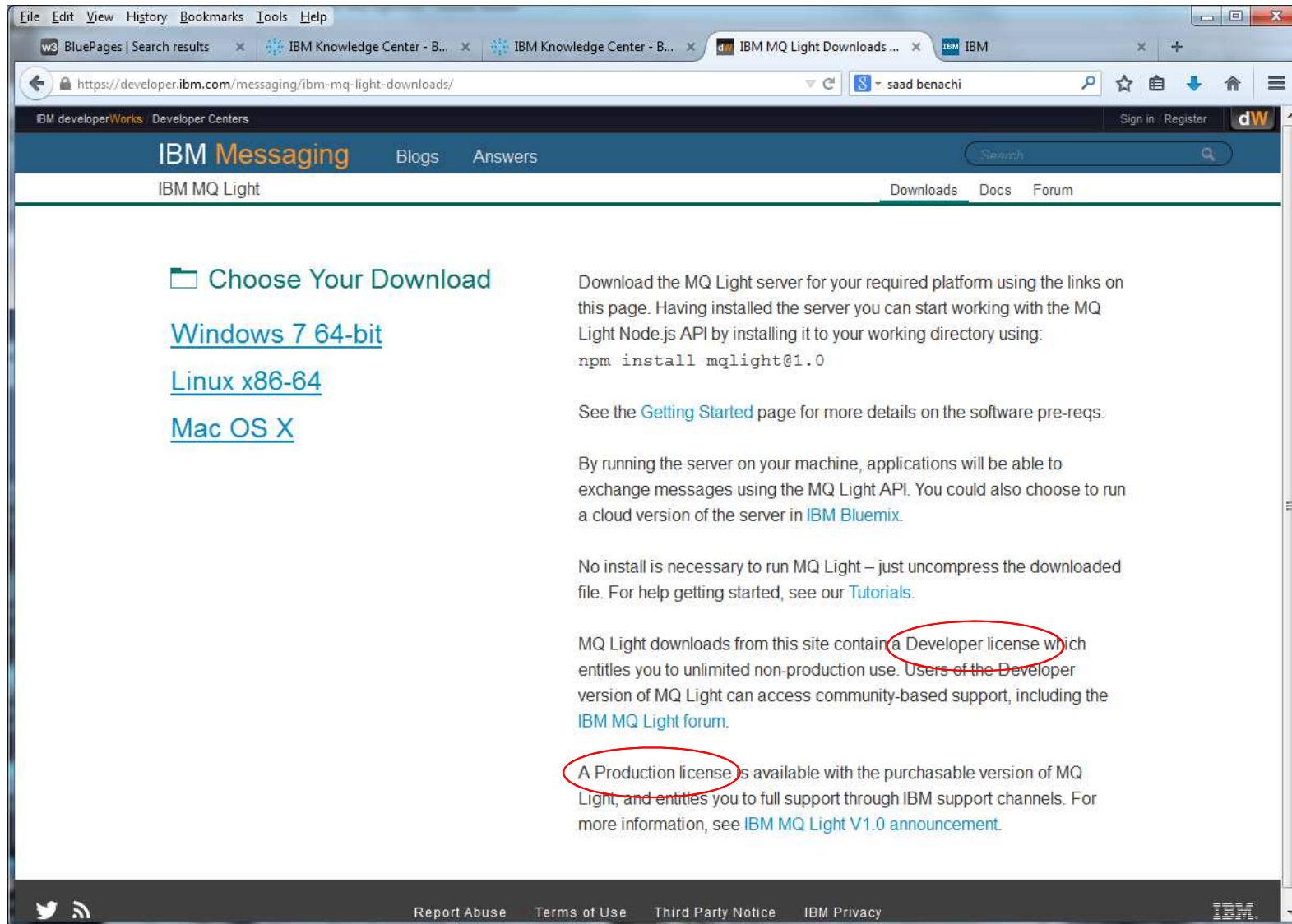


Where is MQ Light now (Sept 2014)?

- ▶ New IBM product, based (of course!) on the MQ Light beta from DeveloperWorks. See ENUSJP14-0492.
- ▶ Available for download for
 - Windows (x86-64, Win 7 & Win Server 2008)
 - Linux x86-64 (Ubuntu, RHEL)
 - Mac OS (development only)
- ▶ Download from standard IBM product delivery channels (electronic delivery only) for the product, or...
<https://developer.ibm.com/messaging/ibm-mq-light-downloads/>
- ▶ 420 MB unzipped (including embarked Java, Liberty and doc!)



MQ Light on DeveloperWorks



File Edit View History Bookmarks Tools Help

w3 BluePages | Search results x IBM Knowledge Center - B... x IBM Knowledge Center - B... x IBM MQ Light Downloads ... x IBM IBM

https://developer.ibm.com/messaging/ibm-mq-light-downloads/ saad benachi

IBM developerWorks Developer Centers Sign in Register dW

IBM Messaging Blogs Answers Search

IBM MQ Light Downloads Docs Forum

Choose Your Download

[Windows 7 64-bit](#)

[Linux x86-64](#)

[Mac OS X](#)

Download the MQ Light server for your required platform using the links on this page. Having installed the server you can start working with the MQ Light Node.js API by installing it to your working directory using:

```
npm install mqlight@1.0
```

See the [Getting Started](#) page for more details on the software pre-reqs.

By running the server on your machine, applications will be able to exchange messages using the MQ Light API. You could also choose to run a cloud version of the server in [IBM Bluemix](#).

No install is necessary to run MQ Light – just unzip the downloaded file. For help getting started, see our [Tutorials](#).

MQ Light downloads from this site contain a [Developer license](#) which entitles you to unlimited non-production use. Users of the Developer version of MQ Light can access community-based support, including the [IBM MQ Light forum](#).

A [Production license](#) is available with the purchasable version of MQ Light, and entitles you to full support through IBM support channels. For more information, see [IBM MQ Light V1.0 announcement](#).

Report Abuse Terms of Use Third Party Notice IBM Privacy

Getting started with MQ Light

1. Make sure that you have Node.js installed
2. Download and deZip MQ Light
3. Run a simple test:

```
C:\CarlTest> mqlight-start  
C:\CarlTest> npm install mqlight  
C:\CarlTest> node C:\CarlTest\node_modules\mqlight\samples\send.js  
Connected to amqp://localhost:5672 using client-id send_1e784d9  
Sending to: public  
Hello World!
```

Start up, once

Needed

Start running the MQ
Light apps



MQ Light Admin

The screenshot displays the IBM MQ Light Admin web interface in a browser window. The browser's address bar shows `localhost:9180/#page=home`. The interface includes a top navigation bar with links for `View Messages` and `Documentation`. A status bar indicates `Clients: 1 connected, 4 disconnected` and `Since last clear history: 1.3 h`. The main content area is divided into three sections: **Sending** (Sent messages: 4), **Messages** (All messages, Topic pattern: Any), and **Receiving** (Received messages: 3). The **Sending** section lists four messages with IDs `send_1e784d9`, `send_c039adb`, `send_9800fd5`, and `send_094b50f`, each with a 'Topic List' link. The **Messages** section shows a list of messages, with the first one being '1 min some data Carl' (1/1) and the second being '6 mins Hello World!' (1/1). The **Receiving** section shows a single message `recv_94c336a` with a 'Pattern: public' and a count of 3. The interface also features a sidebar with links to `IB10NODE`, `TESTNODE`, `MessageSight`, `IB10NODE2`, `JavaScript API`, and `Employee Application`.



IBM Messaging portfolio



Developers : making the technology decisions

Systems of engagement, creating responsive and scalable applications. Sometimes seen as “rogue” developers



IT : Enterprise messaging backbone

Delivering rock solid infrastructure that links together systems of record.



Device manufacturers

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



Backup



Bibliography

- ▶ ENUSJP14-0492 MQ Light Announcement letter
- ▶ MQ Light home page: <https://developer.ibm.com/messaging/mq-light/>
- ▶ MQ Light Node.js API documentation : <https://www.npmjs.org/package/mqlight>



What is MQ Light and Why is it important?

MQ Light : www.ibmmdw.net/messaging/mq-light/

MQ Light service : www.bluemix.net

- ▶ Messaging **designed for developers** to create responsive and scalable applications
 - Simplified messaging API
 - Easy to acquire and incorporate into applications
 - Development focused tooling
- ▶ Range of application **deployment options**
 - MQ Light
 - IBM MQ [Statement of Direction]
 - Elastic MQ in BlueMix

