Connecting IoT Devices and Mobile Applications to your Enterprise with IBM IoT MessageSight & IBM MQ

Dr. Jonathan Levell
Lead Architect for IBM IoT MessageSight
Hursley Park, IBM
levell@uk.ibm.com
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.
Agenda

- What is MessageSight?
- Brief MQTT Overview
- Connecting MessageSight to MQ
- A Closer Look at MessageSight’s Features
- A couple of MessageSight Use cases
- Q&A
MessageSight is designed to make it **simple** to **reliably** and **securely connect** mobile and Internet of Things devices & data.
MessageSight Eye-View of the World

Clients (Internet)
- Mobile device
- Sensor device
  - MQTT
  - MQTT over websockets

Backend services (Intranet)
- Back-end services
- C Application
- Java Application
- JMS
- Resource Adapter
- JEE Application Server
- WMQ Server

IBM IoT MessageSight (Primary)
- MQTT
- Custom protocol
- Using protocol plug-in

IBM IoT MessageSight (Standby)
- JMS
- MQ Connectivity
- Browser

System Administrator
Installation Options for Linux

- Standard Linux RPM
  - CentOS V7 or Redhat V7
- Cloud configurations
- Allows other software to be installed
  - Monitoring, security, administration
- Cloud, Virtualized environments
  - Public or private
- Support for Docker
- Developer edition now available on IBM Cloud Private
Supported Protocols

- MQTT over TCP/IP
  - Clients
    - C
    - Java
    - Android
    - iOS
- MQTT over WebSockets
- JMS
- Custom Protocols
  - Extensible framework enables customers to write their own protocol plug-ins
So... Devices?

Low power short range

Prototyping boards/kits

Energy harvesting sensors

NodeMcu

Meshing nodes

Texas Instruments

Low power wide area

Raspberry Pi

EnOcean

Advantech

LoRa

MultiTech
But also….

Web Browsers

(WebSockets)

Phone

(Often in combination with push notification)
Reliably and quickly deliver data with MQTT

MQTT

Open

- Open royalty free specification
- Wide variety of clients and servers
  - Hobbyist to enterprise
  - Open source to commercial

Lean

- Minimal pub/sub messaging semantics
  - Asynchronous ("push") delivery
  - Simple set of verbs -- connect, publish, subscribe and disconnect

Simple

- Minimized on-the-wire format
  - Smallest packet size 2 bytes
  - Scalable
  - Low footprint
  - Clients: C=30Kb; Java=100Kb

Reliable

- Three qualities of service
  - 0 – at most once delivery
  - 1 – assured delivery dups ok
  - 2 – once and once only delivery
  - Copes with loss of contact between client and server.
  - “Last will and testament” to publish a message if the client goes offline.

MQTT 3.1.1

- current ISO Standard
MQTT v5

The OASIS MQTT Technical Committee has developed a new version of the MQTT standard, to be called v5:
– The number jumps from v3.1.1 straight to v5, without having a v4. This is to align the spec version number with the version in the Connect packet

The committee has completed its technical work – the final Committee Specification is now available at

http://docs.oasis-open.org/mqtt/mqtt/v5.0/cs02/mqtt-v5.0-cs02.html

The new specification addresses a number of points that users have raised with MQTT 3.1.1, as well as adding new features
Connecting MessageSight to MQ

MQ Connectivity
Configure connections to one or more WebSphere MQ queue managers.

Queue Manager Connection Properties
Define, edit, or delete information about how the server connects to the queue managers.

Add Destination Mapping Rule
Destination mapping rules define the direction in which messages are moved, and the nature of the source and target objects.

- Name: MESSAGESIGHT_TO_MQ
- Rule Type: MessageSight topic subtree to MQ topic subtree
- Source: MessageSightTreeSrcTopic
- Destination: MQTreeDestTopic
- Max Messages: 5000
- Retained Messages: None
- Enabled: Yes

Associated queue manager connections:
- MQ1Connect 10.67.45.4(1414)
- MQ2Connect 10.67.45.5(1414)
- MQ3Connect 10.67.45.6(1414)

Save | Cancel
More on MQ<->MessageSight

• Each connection is a standard MQ Client connection under the covers
• Can have multiple connections to different (or the same) queue manager
• Uses 2-phase transactions for transport of reliable messages
# MessageSight is easy, scalable and secure

<table>
<thead>
<tr>
<th>Designed for Things</th>
<th>Developer Friendly</th>
<th>Easy to Deploy</th>
<th>Easy to Integrate</th>
<th>Internet Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Optimized gateway for Things and Mobile devices</td>
<td>• Active dev community</td>
<td>• Up and running &lt; 30 minutes</td>
<td>• JMS</td>
<td>• Large numbers of concurrent connections</td>
</tr>
<tr>
<td>• Efficient open protocol</td>
<td>• Free developer edition</td>
<td>• Task oriented UI guides administrator through first steps</td>
<td>• WebSockets</td>
<td>• High throughput</td>
</tr>
<tr>
<td>• Event-driven awareness</td>
<td>• Simple yet powerful APIs</td>
<td>• Simple and scalable management through policies</td>
<td>• MQ</td>
<td>• Highly available</td>
</tr>
<tr>
<td>• Open and industry agnostic</td>
<td>• Simple messaging paradigm</td>
<td></td>
<td>• Integration Bus</td>
<td>• Horizontally scalable</td>
</tr>
<tr>
<td>• Fine-grained security policies</td>
<td>• 40+ MQTT client libraries</td>
<td></td>
<td>• Worklight</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• InfoSphere Streams...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Supports Oasis standard MQTT 3.11</td>
<td></td>
</tr>
</tbody>
</table>

MessageSight is scalable, secure, and easy to deploy. It offers a simple yet powerful API, a task-oriented UI for administrators, and supports multiple integration methods like JMS, WebSockets, MQ, Integration Bus, Worklight, InfoSphere Streams, and supports Oasis standard MQTT 3.11. It is scalable and supports large numbers of concurrent connections, high throughput, and high availability.
MessageSight Performance

128 byte messages on server with:
Dual Xeons 10-core, 256GB RAM, SSDs, multiple 10GbE NICs

<table>
<thead>
<tr>
<th>QoS</th>
<th>#Pubs</th>
<th>#Subs</th>
<th>Max Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Many</td>
<td>~100</td>
<td>~500K/sec</td>
</tr>
<tr>
<td>1</td>
<td>Many</td>
<td>~100</td>
<td>~200K/sec</td>
</tr>
<tr>
<td>2</td>
<td>Many</td>
<td>~100</td>
<td>~200K/sec</td>
</tr>
</tbody>
</table>

Connection Rate: 10K conn/sec non-TLS, 1-4K conn/sec TLS
4K concurrent/conn per GB up to 1M concurrent connections
Single WebUI for multiple MessageSights

- WebUI administers multiple MessageSight Instances

- Separate Install (RPM/Docker)

- Same or different server

- WebUI uses port 9087
  - Runtime instances use port 9089 for REST Commands
Status Overview

Web UI and REST Interfaces
Endpoints and Policies

Demo Hub

Demo Message Hub.

## Endpoints

An endpoint is a port that client applications can connect to. An endpoint must have at least one connection policy and one messaging policy.

<table>
<thead>
<tr>
<th>Endpoint</th>
<th>Port</th>
<th>Enabled</th>
<th>Status</th>
<th>Connection Policies</th>
<th>Messaging Policies</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DemoEndpoint</td>
<td>16102</td>
<td>✗</td>
<td>✗</td>
<td>✗ 1</td>
<td>✗ 2</td>
<td>Unsecured endpoint for demonstration use only. By default, both JMS and MQTT protocols are accepted.</td>
</tr>
<tr>
<td>DemoMqttEndpoint</td>
<td>1883</td>
<td>✓</td>
<td>✓</td>
<td>✗ 1</td>
<td>✗ 2</td>
<td>Unsecured endpoint for demonstration use with MQTT protocol only. By default, it uses port 1883.</td>
</tr>
</tbody>
</table>

*Total: 2 Selected: 0*
### Who can connect?

A connection policy authorizes clients to connect to IBM IoT MessageSight endpoints. Each endpoint must have at least one connection policy.

#### Edit Connection Policy

A connection policy authorizes clients to connect to IBM IoT MessageSight endpoints. Each endpoint must have at least one connection policy.

<table>
<thead>
<tr>
<th>Connection Policy</th>
<th>Endpoints</th>
<th>Allow Clients with Durable Subscriptions</th>
<th>Allow Persistent Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>DemoConnectionPolicy</td>
<td>2</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>

Total: 1 Selected: 1

To restrict connections using this policy to specific clients, specify one or more of the following filters. For example, select Group ID to restrict this policy to members of a particular group. The policy allows access only when all of the specified filters are true:

- **Client IP Address:**
- **User ID:**
- **Certificate Common Name:**

Specify the resources that an MQTT client is permitted to consume:

- Allow Clients with Durable Subscriptions: **On**
- Allow Persistent Messages: **On**

---

**Save** | **Cancel**
# Configuring Topic Policies

A topic policy authorizes connected clients to perform specific messaging actions, such as which topics the client can access on IBM MQ MessageSift. Each endpoint must have at least one topic policy, subscription policy, or queue policy.

## Topic Policies

<table>
<thead>
<tr>
<th>Messaging Policy</th>
<th>Endpoints</th>
<th>Max Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepQ</td>
<td>1</td>
<td>20,000,000</td>
</tr>
<tr>
<td>DemoTopicPolicy</td>
<td>2</td>
<td>5,000</td>
</tr>
</tbody>
</table>

Total: 2 Selected: 1

## Subscription Policies

<table>
<thead>
<tr>
<th>Messaging Policy</th>
<th>Endpoints</th>
<th>Max Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>DemoSubscriptionPolicy</td>
<td>1</td>
<td>5,000</td>
</tr>
</tbody>
</table>

Total: 1 Selected: 0

## Edit Topic Policy

To restrict the messaging actions that are defined in this policy to specific clients, specify one or more of the following filters. For example, select Group ID to restrict this policy to members of a particular group. The policy allows access only when all of the specified filters are true:

- **Client IP Address:**
- **User ID:**
- **Certificate Common Name:**
- **Client ID:**
- **Group ID:**
- **Protocol:**

### Specify the resources and messaging actions that the client is permitted to access:

- **Topic:**
- **Authority:** Publish, Subscribe

### Subscriber Settings

- **Max Messages:** 20,000,000
- **Max Messages Behavior:** Reject new messages
- **Disconnected Client Notification:**

### Publisher Settings

- **Max Message Time To Live:** unlimited seconds

[Save] [Cancel]
Flexible Policies

Policies can have variables substitution, a small number of policies can apply to many users and topics.
Expiry and Discard

Administrative control of maximum message time-to-live
- Automatic expiration of messages beyond configured time, even with MQTT v3
- Choose the maximum messages behaviour
  - When a subscriber cannot keep up
    - Reject new messages
    - Discard old messages
Monitoring Subscriptions

Subscription Monitor
Monitor subscriptions using various subscription statistics. Delete durable subscriptions. Up to 100 subscriptions can be viewed.

<table>
<thead>
<tr>
<th>Topic String</th>
<th>Subscription Name:</th>
<th>Query:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Subscriptions with Most Messages Published</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Client ID:</th>
<th>Subscription Type:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All</td>
</tr>
</tbody>
</table>

Last Updated: 8/11/16 1:54:08 PM

### Subscription Monitor

<table>
<thead>
<tr>
<th>Topic String</th>
<th>Subscription Properties</th>
<th>Total Messages</th>
<th>Buffered Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Name</td>
<td>Published</td>
<td>Maximum</td>
</tr>
<tr>
<td></td>
<td>Client ID</td>
<td>Rejected</td>
<td>Expired / Discarded</td>
</tr>
<tr>
<td></td>
<td>Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Messaging Policy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Consumers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>#</td>
<td>772,846</td>
<td>0</td>
</tr>
<tr>
<td>topic1</td>
<td>topic1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>topic1</td>
<td>dursubber2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>topic1</td>
<td>dursubber1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Total: 4 Selected: 0

Page 23
Throughput charts with incoming/outgoing rates

Connection charts

Detailed resource usage charts
Interactive MQTT Client

MQTT Helper

Connect

Server
messagesight.demos.ibm.com

Port
1883

Client ID
MQTTHelper-26380

Username
(optional)

Password
(optional)

Clean Session
OFF ON

SSL
OFF ON

Subscribe

Log
Clear
Follow

Publish

http://mqtt-helper.mybluemix.net/
Shared Subscriptions

- Sharing messages on a subscription amongst multiple clients

- From MQTT, subscribe from **standard client** by modifying topic filter:
  
  $\text{SharedSubscription/\langle subname\rangle/\langle topicfilter\rangle}$

![Diagram of shared subscription with three clients: Client 1, Client 2, and Client 3 connected to a shared inbox/todo topic.]
Highly Available Pairs

• An HA Pair is consists of a actively running system (“Primary”) and another MessageSight ready to take over (“secondary”).

• Communication between Primary & Secondary takes place over two separate Network Interfaces (discovery & replication) to minimise chances of “Split Brain”

• Messages (QoS > 0) are recorded at both systems before ack is sent to client.

• No Shared Storage
Clusters add a link that messages can flow over between MessageSights.

“Device-like” subscriptions made on one MessageSight are advertised around the cluster and other nodes in the cluster forward publications to a MessageSight if it has matching subscriptions....
A Few Clustering Key Points

Durable clients still need a home - and to be able to find it
Buffered messages and which messages are inflight are not broadcast around the cluster

Back-end apps consuming large message volumes need to consume from all nodes where messages are incoming.

Usually
Do Not Want:

To help prevent this MQ Connectivity, JMS and shared subs (usually associated with high message volumes) are not advertised around the cluster; a separate sub needs to be created on each node.
Application can publish without knowing where the client is connected in the cluster
Clusters: Device to BackEnd

- Need to prevent all messages going through a single point – becoming a bottleneck
- Back-end Subs will not pull data through the cluster connector from other members
- Picture would be same if QMqr replaced with JMS App or MQTT shared sub
Balancing your workload

- Route connection requests
  - Recognize previous state
- Hide MessageSight topology
  - Multiple servers and/or HA instances
  - F5 has native MQTT support
“Easy” for cleanSession=true clients and QoS < 2 publishers

For CleanSession=false, need client affinity to a particular “home” server (or server pair)

Client Affinity can be done e.g. based on ClientID by F5

Can instead have clients know about individual servers (or server-pairs) e.g. by group devices into groups of say 5000 and having a DNS entry per group

(So a device might connect to: group53.iot.mycompany.com

(Can then redirect DNS entry to move devices)
Protocol plugin point

- A protocol plug-in point in MessageSight
  - Gives the ability for IBM and trusted third-parties to extend the protocols supported by the appliance
- There is a plug-in protocol SDK to develop plug-ins
  - Protocols must be written in Java
    - Plug-ins run in an isolated environment to ensure robustness
    - The plug-in sends and receives data through MessageSight transport component
    - The plug-in uses a messaging interface into the MessageSight engine
    - The common engine ensures any-to-any communication between protocols
    - Integrated into MessageSight configuration and security models
    - Plugins do not get direct network access
  - Sample HTTP style plugin
- Intended for:
  - Legacy sensors which cannot be changed to MQTT
  - Industry-specific protocols
  - Co-existence with initial versions of IoT deployments
The Chamberlain Group Inc. developed a mobile IoT app that provides competitively differentiated capabilities and user interface.

One million users can now access platform without drop off in service – 10X performance improvement.

Speeds deployment and development of new features with total platform control.

Opens new business opportunities for additional connected products.
European Bank uses IBM MessageSight and push technology to drive personalized loan and savings offers directly to customers’ mobile devices.

- Innovation driven from banking transaction data
- Ensured security, confidentiality and delivery of transactions
- Lightweight responsiveness for users
GreatCall’s innovative suite of easy-to-use mobile products and approach to customer care helps aging consumers live more independent lives.

An IBM IoT MessageSight – driven IoT solution connects mobile and wearable devices that scale to meet GreatCall’s rapid growth, with:

• Secure support for projected 40,000 messages per second
• Single publish with multiple subs
• Low latency connections that provide real time results
Grand Slam tennis tournaments use MessageSight for their scoring system to provide live scores to millions of fans using different devices.

- Tens to hundreds of thousands of concurrently clients connected to MessageSight, and experiencing tournament play via iPad, Android, mobile and web sites.
- Scoreboard loads are 60% faster than original flash-based solution, with subsecond response times compared with 3-5 seconds previously.
Wimbledon 2015

- Publish Live Scores to Desktop and Mobile Users
- Sending over 10 million messages a minute at peak

“I was sat at the side of Centre Court one day with my smartphone app, and I heard the umpire call the point in the match after I saw my mobile phone update.”

Sam Seddon,
IBM Wimbledon Client and Programme Executive
MessageSight is easy, scalable and secure

<table>
<thead>
<tr>
<th>Designed for Things</th>
<th>Developer Friendly</th>
<th>Easy to Deploy</th>
<th>Easy to Integrate</th>
<th>Internet Scale</th>
</tr>
</thead>
</table>
| • Optimized gateway for Things and Mobile devices  
  • Efficient open protocol  
  • Event-driven awareness  
  • Open and industry agnostic  
  • Fine-grained security policies | • Active dev community  
  • Free developer edition  
  • Simple yet powerful APIs  
  • Simple messaging paradigm  
  • 40+ MQTT client libraries | • Up and running < 30 minutes  
  • Task oriented UI guides administrator through first steps  
  • Simple and scalable management through policies | • JMS  
  • WebSockets  
  • MQ  
  • Integration Bus  
  • Worklight  
  • InfoSphere Streams...  
  • Supports Oasis standard MQTT 3.11 | • Large numbers of concurrent connections  
  • High throughput  
  • Highly available  
  • Horizontally scalable |

IBM MessageSight
Questions & Answers
Image Credits

“Grass Court Season is Here!” by Kate
https://www.flickr.com/photos/43555660@N00/18161963964/

“Pair of Seagulls” by Raymond McCrae
https://www.flickr.com/photos/raymccrae/115980566

Formula One Car by Gerald G
https://openclipart.org/detail/8554/formula-one-car