

IoT Networking

Functionality based IoT Protocol

Connectivity (6LowPAN, RPL)

Identification (EPC, uCode, IPv6, URIs)

Communication / Transport (WiFi, Bluetooth, LPWAN)

Discovery (Physical Web, mDNS, DNS-SD)

Data Protocols (MQTT, CoAP, AMQP, Websocket, Node)

Device Management (TR-069, OMA-DM)

Semantic (JSON-LD, Web Thing Model)

Multi-layer Frameworks (Alljoyn, IoTivity, Weave, Homekit)

Message Queue Telemetry Transport (MQTT)

Message Queue Telemetry Transport.

ISO standard (ISO/IEC PRF 20922).

It is a publish-subscribe-based lightweight messaging protocol for use in conjunction with the TCP/IP protocol.

MQTT was introduced by IBM in 1999 and standardized by OASIS in 2013.

Designed to provide connectivity (mostly embedded) between applications and middle-wares on one side and networks and communications on the other side.

Message Queue Telemetry Transport (MQTT)

A message broker controls the publish-subscribe messaging pattern.

A topic to which a client is subscribed is updated in the form of messages and distributed by the message broker.

Designed for:

- Remote connections
- Limited bandwidth
- Small-code footprint

Message Queue Telemetry Transport (MQTT)

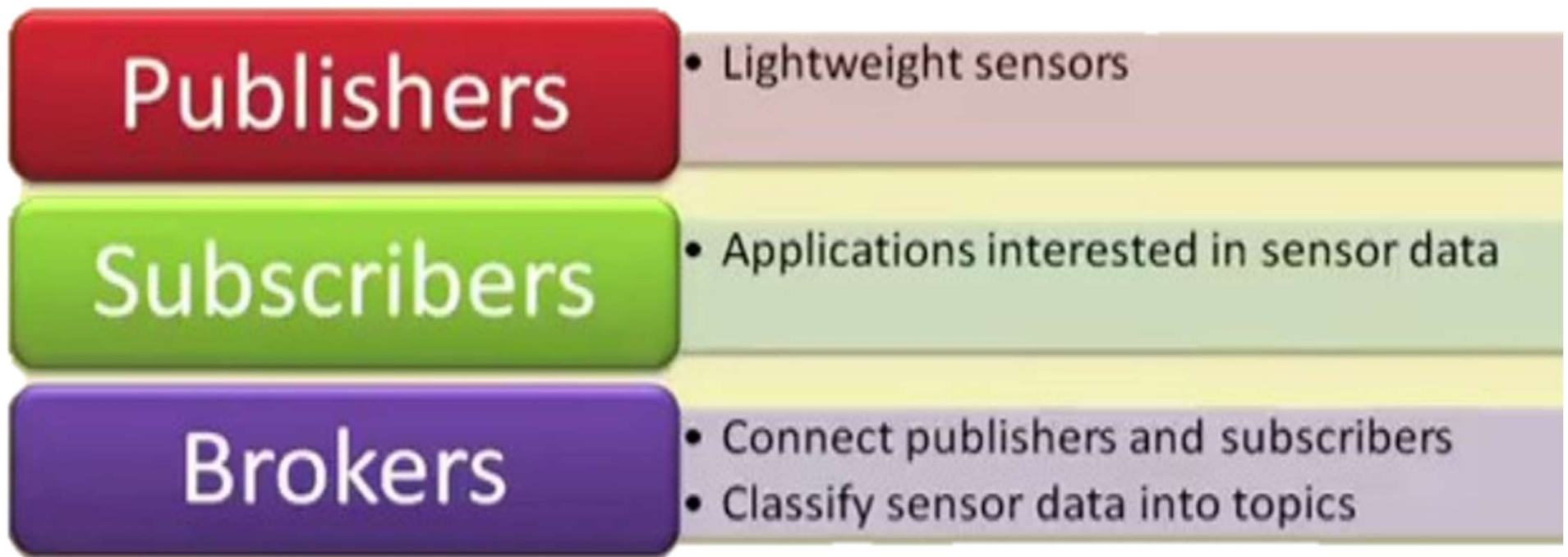
A message broker controls the publish-subscribe messaging pattern.

A topic to which a client is subscribed is updated in the form of messages and distributed by the message broker.

Designed for:

- Remote connections
- Limited bandwidth
- Small-code footprint

MQTT Components

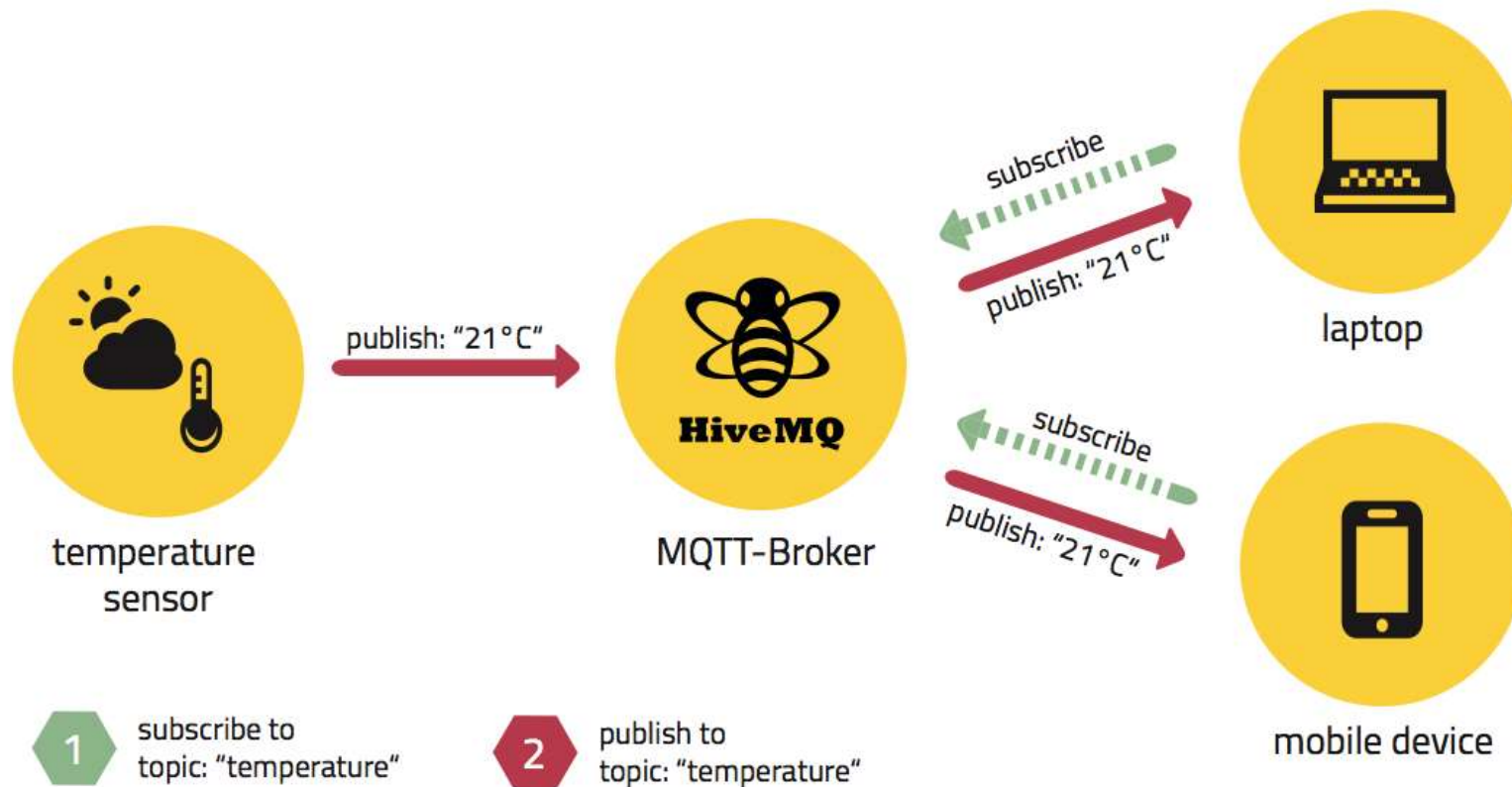


MQTT Methods



Ref: <https://en.wikipedia.org/wiki/MQTT>

MQTT



https://www.eclipse.org/community/eclipse_newsletter/2014/october/article2.php

MQTT Communication

The protocol uses a **publish/subscribe** architecture (HTTP uses a request/response paradigm).

Publish/subscribe is **event-driven** and enables messages to be pushed to clients.

The central **communication point is the MQTT broker**, which is in charge of dispatching all messages between the senders and the rightful receivers.

Each client that publishes a message to the broker, includes a **topic** into the message. The **topic is the routing information for the broker**.

MQTT Communication

Each client that wants to receive messages subscribes to a certain topic and the broker delivers all messages with the matching topic to the client.

Therefore the clients don't have to know each other. They only communicate over the topic.

This architecture enables highly scalable solutions without dependencies between the data producers and the data consumers.

MQTT Topics

A topic is a **simple string** that can have more hierarchy levels, which are separated by a slash.

A sample topic for sending temperature data of the living room could be *house/living-room/temperature*.

On one hand the client (e.g. mobile device) can subscribe to the exact topic or on the other hand, it can use a **wildcard**.

MQTT Topics

The subscription to *house/+/temperature* would result in all messages sent to the previously mentioned topic *house/living-room/temperature*, as well as any topic with an arbitrary value in the place of living room, such as *house/kitchen/temperature*.

The plus sign is a **single level wild card** and only allows arbitrary values for one hierarchy.

If more than one level needs to be subscribed, such as, the entire sub-tree, there is also a **multilevel wildcard** (#).

It allows to subscribe to all underlying hierarchy levels.

For example *house/#* is subscribing to all topics beginning with *house*.

MQTT Applications

Facebook Messenger uses MQTT for online chat.

Amazon Web Services use Amazon IoT with MQTT.

Microsoft Azure IoT Hub uses MQTT as its main protocol for telemetry messages.

The **EVERYTHING IoT platform** uses MQTT as an M2M protocol for millions of connected products.

Adafruit launched a free MQTT cloud service for IoT experimenters called Adafruit IO.

Thank you