

95-733 Internet of Things Publish Subscribe with MQTT

Where are we?

Internet Protocol Suite

We are here!

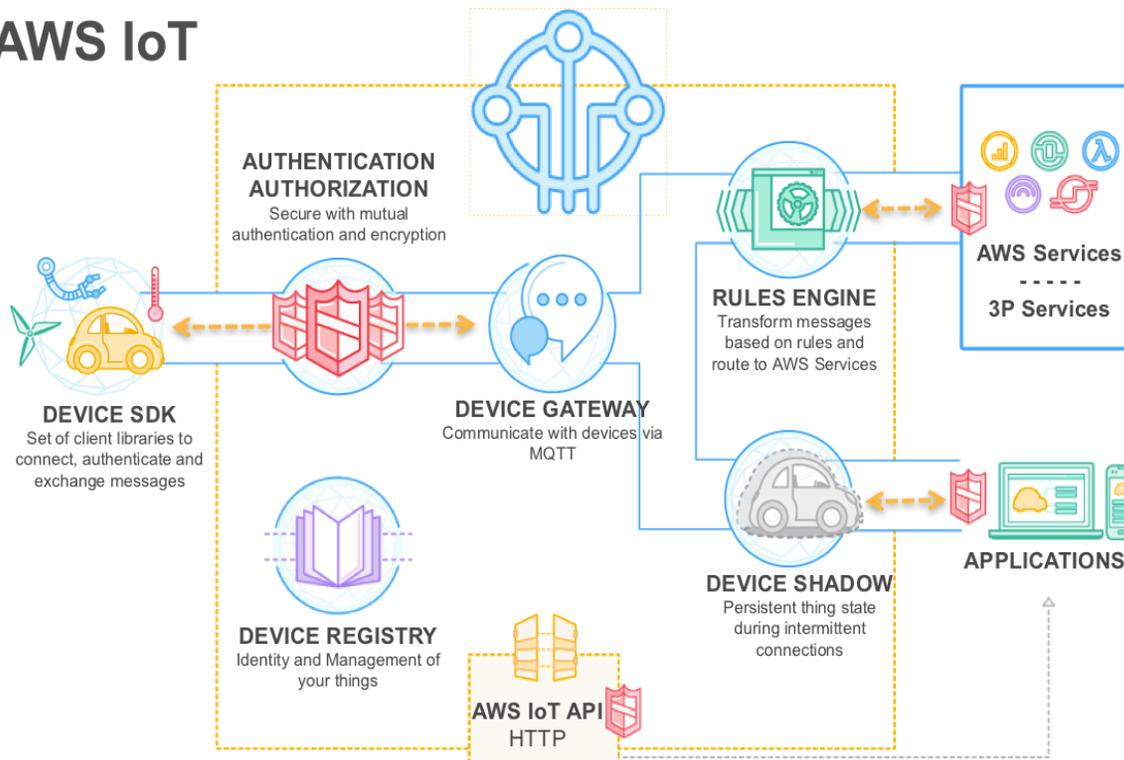
HTTP, Websockets, DNS, XMPP, MQTT, CoAp	Application layer
TLS, SSL	Application Layer (Encryption)
TCP, UDP	Transport
IP(V4, V6), 6LowPAN	Internet Layer
Ethernet, 802.11 WiFi, 802.15.4	Link Layer

2

Example: AWS IoT

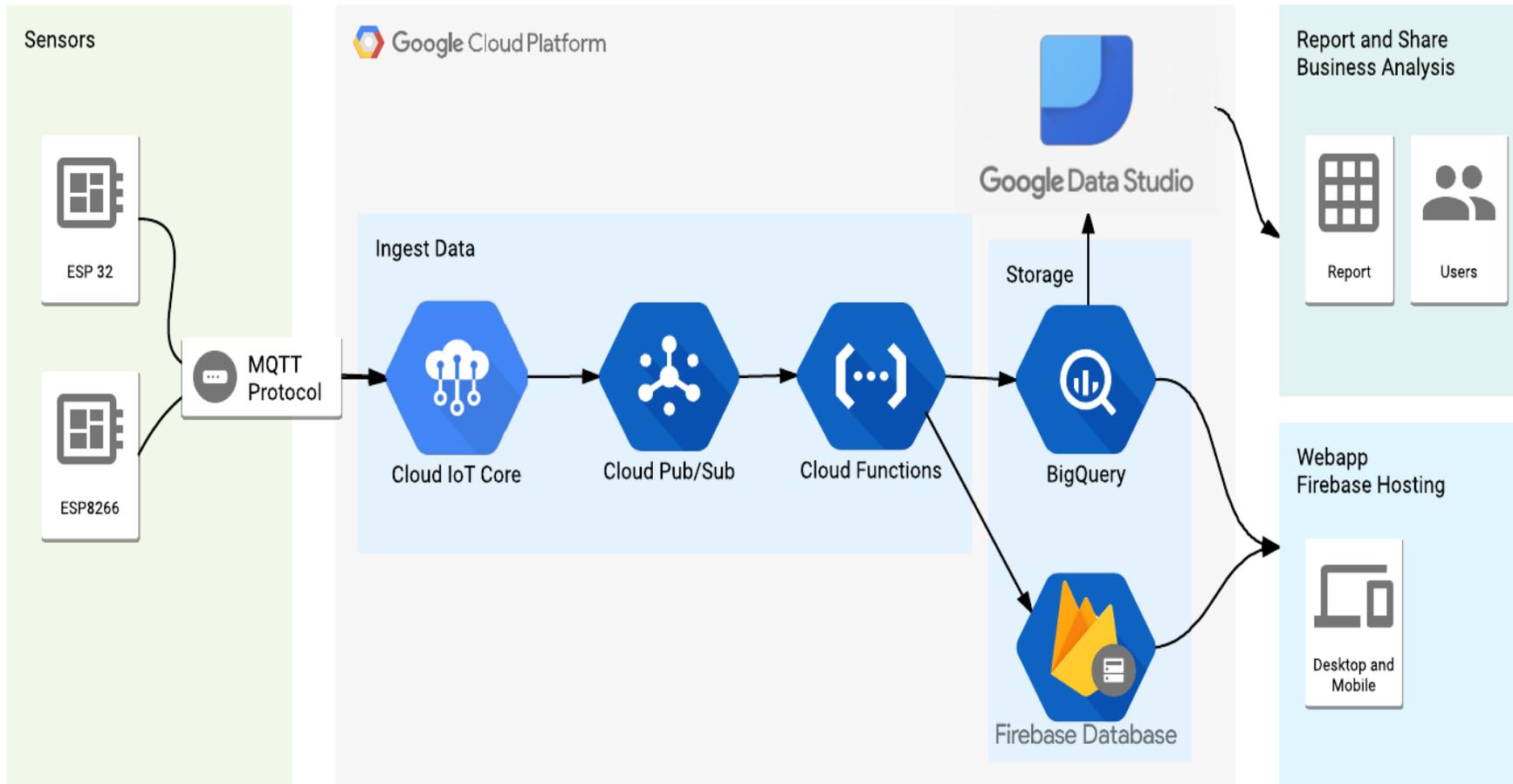
Ask Amazon Alexa to turn on the car and heat until 70 degrees.

AWS IoT

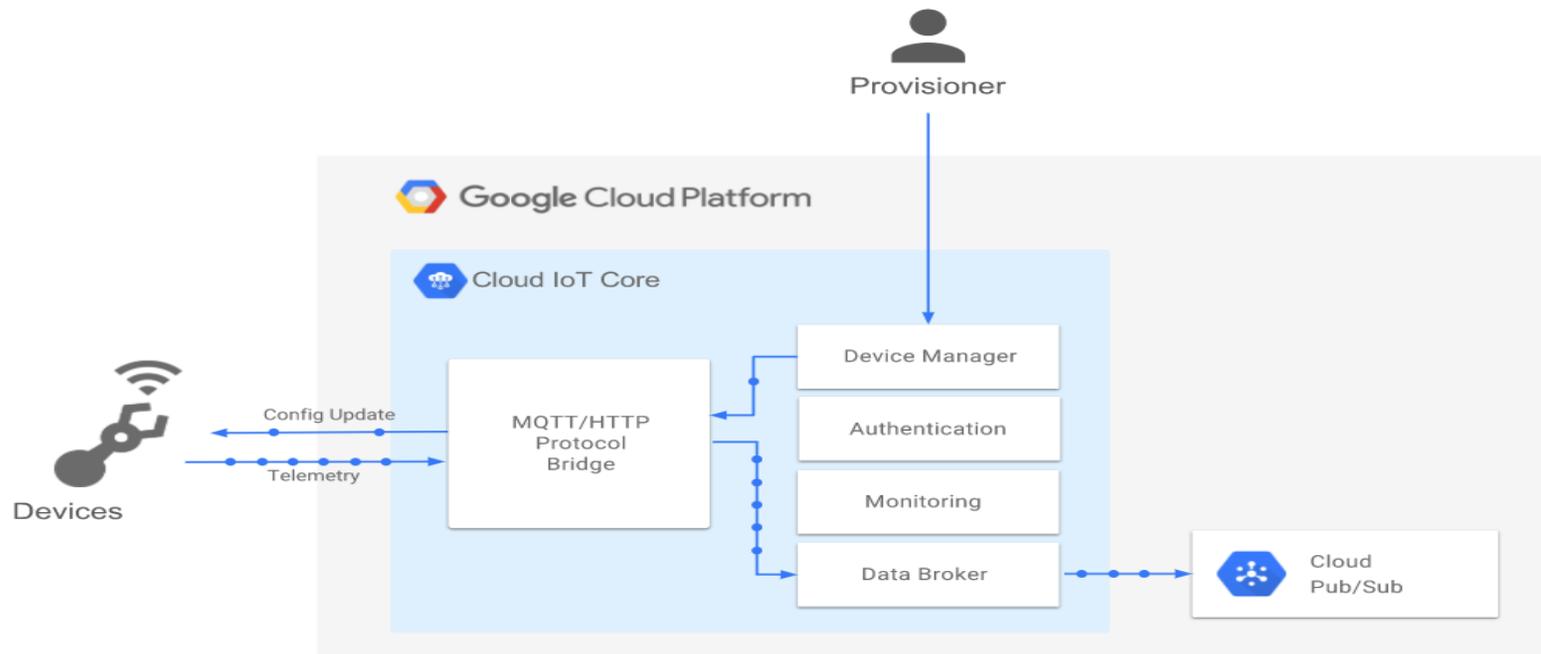


95-733 Internet of Things

Example: Google and MQTT

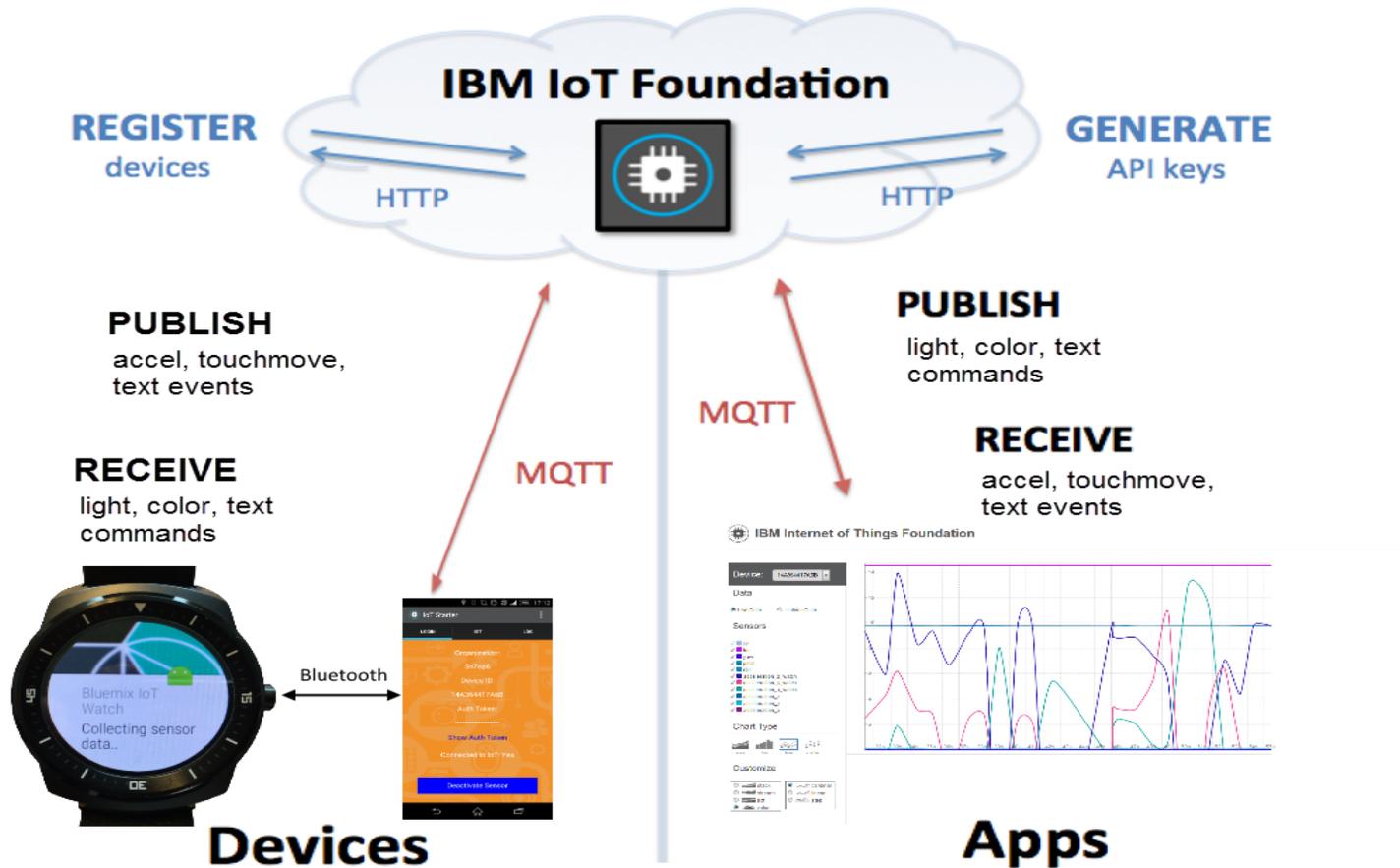


Google IoT-Core

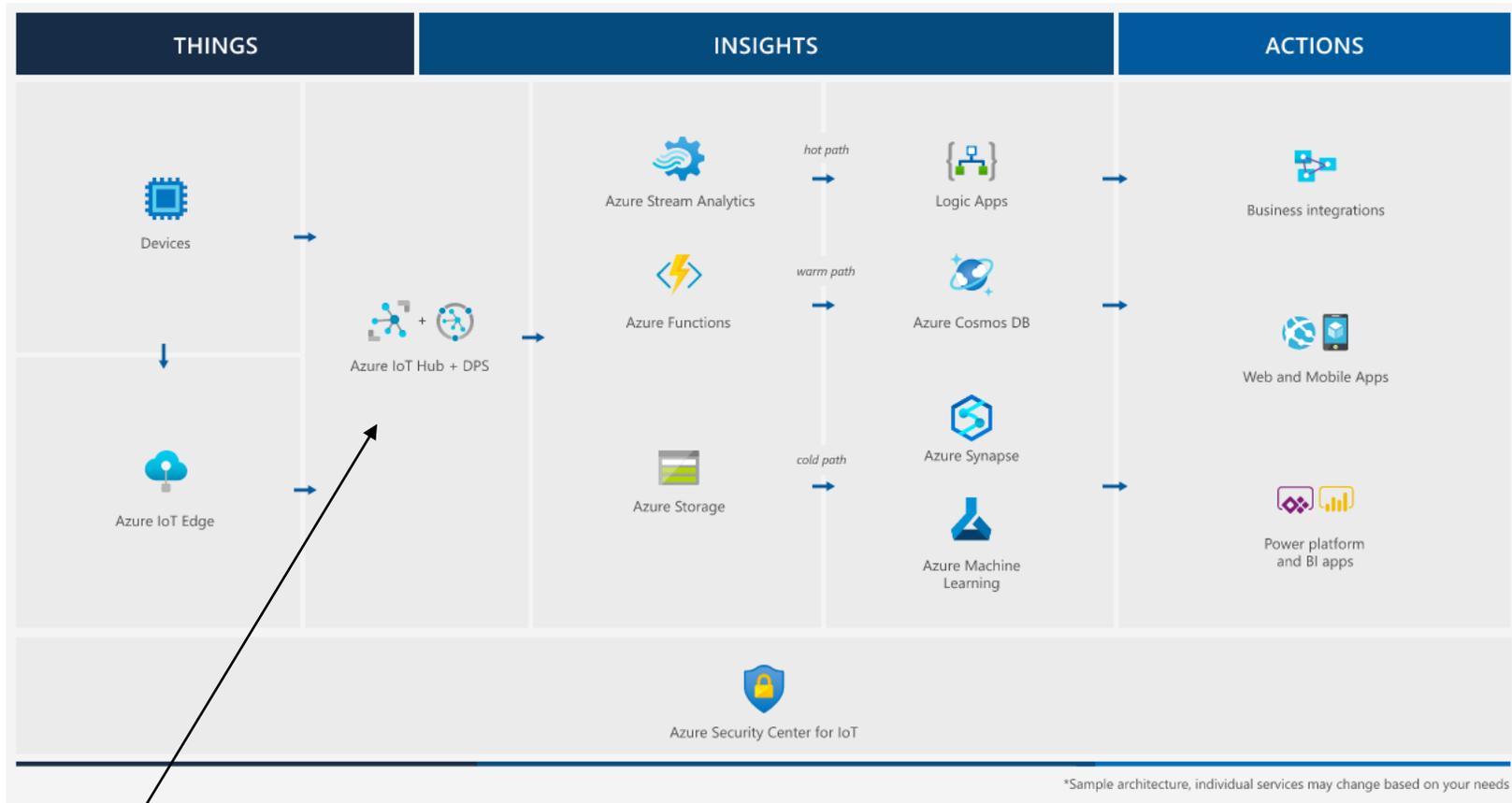


95-733 Internet of Things

Example: IBM and MQTT



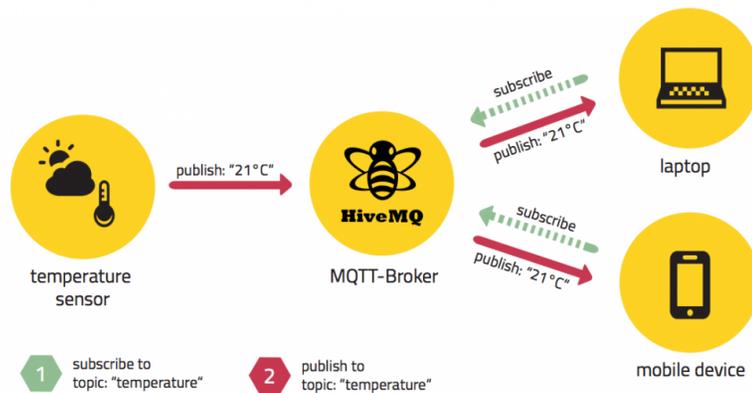
Example: Azure IoT



Uses MQTT or OASIS Advanced Message Queue Protocol (AMQP)
DPS – Device Provisioning Service

95-733 Internet of Things

MQTT: Sensor readings with pub/sub



From MQTT Essentials at HiveMQ

Decoupled in **space** and **time**.

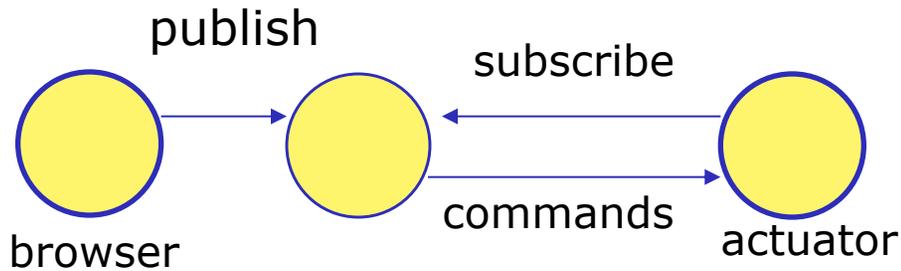
The clients do not need each other's IP address and port (space) and they do not need to be running at the same time (time).

The broker's IP and port must be known by clients.

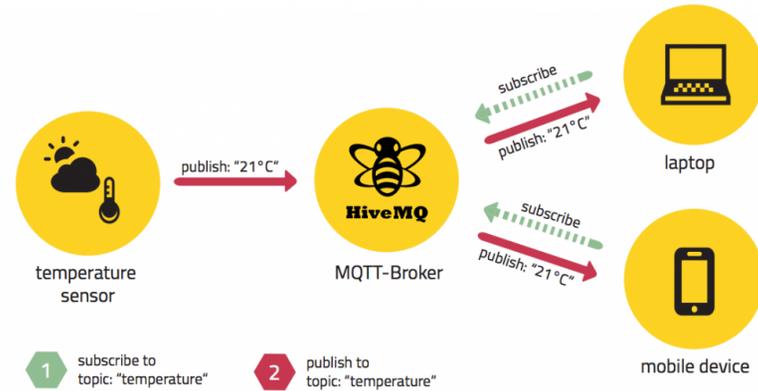
Namespace hierarchy used for topic filtering.

It may be the case that a published message is never consumed by any subscriber.

Actuators too!



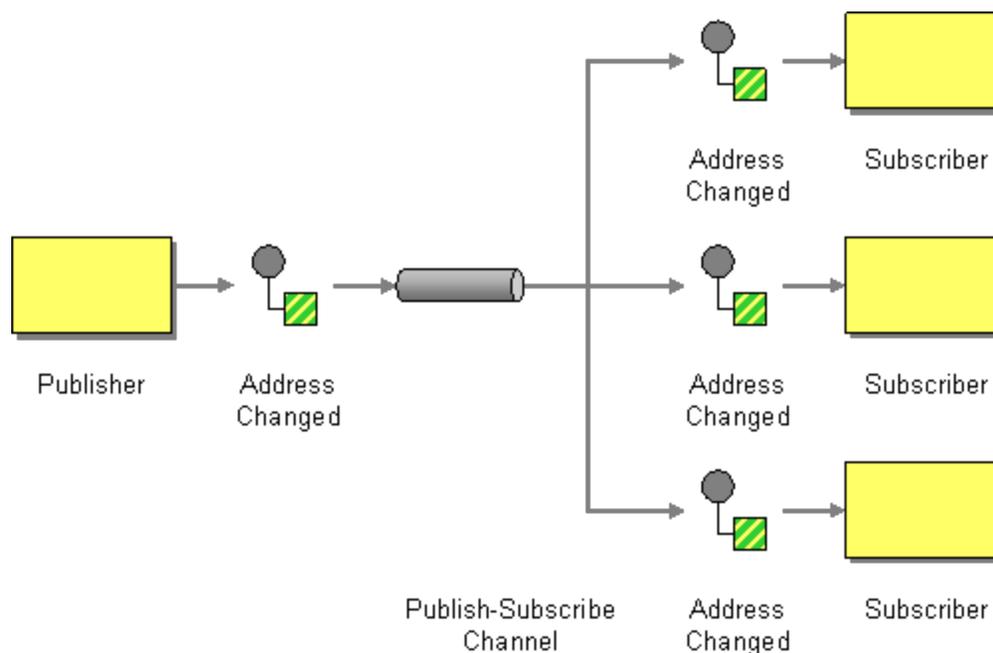
If my toaster is a command subscriber, I can control it over the web!



In the picture, replace the laptop with an actuator, subscribing to a command topic, say, device3/commands. Replace the sensor with a browser sending commands to device3/commands.

Modified from MQTT Essentials at HiveMQ

From Enterprise Integration Patterns Book Hohpe and Woolf



MQTT

- Message Queuing Telemetry Transport (old acronym) since the 1990's
- "Telemetry" is from the Greek remote measure
- Created by Andy Stanford-Clark (IBM) and Alan Nipper - now part of OASIS
- Version 3.1 released royalty free in 2010 (IBM)
- Originally built for oil pipeline monitoring over satellite connections.
- Satellites appropriate because pipelines are remote

MQTT

- Built for a proprietary imbedded system now shifting to IoT
- You can send anything as a message. Up to 256 MB.
- Built for unreliable networks
- Enterprise scale implementations down to hobby projects
- Decouples readers and writers
- Message have a topic, quality of service and retain status associated with them.

MQTT

- MQTT Runs over TCP or TLS. May use Websockets from within a browser.
- MQTT-SN uses UDP packets or serial communication rather than TCP. MQTT-SN may run over Bluetooth Low Energy (BLE).
- Open, industry agnostic, no polling. What does it mean to be open?
- Hierarchical topic namespace and subscriptions with wildcards. MQTT-SN has simpler topics.
- As soon as you subscribe you may receive the most recently published message. One message per topic may be retained by the broker. This feature provides for devices that transmit messages only on occasion. A newly connected subscriber does not need to wait. Instead, it receives the most recent message.

MQTT Last Will and Testament

- Publishing clients may connect with a last will and testament message.
- If publishing client has no data to send, it sends ping requests to the broker to inform the broker that it is still alive.
- If a publisher disconnects in a faulty manner (the broker will miss the ping requests), the broker tells all subscribers the last will and testament. This is for an ungraceful disconnect. How could this happen? Battery failure, network down, out of reach, etc.
- In the case of graceful disconnect, the publisher sends a DISCONNECT message to the broker – the broker will discard the LWT message.

MQTT Clients

- A client may publish or subscribe or do both.
- An MQTT client is any device from a micro controller up to a full fledged server, that has an MQTT library running and is connecting to an MQTT broker over any kind of network. (from HiveMQ MQTT Essentials)
- A client is any device that has a TCP/IP stack and speaks MQTT. MQTT-SN does not require TCP.
- Client libraries widely available (Android, Arduino, iOS, Java, Javascript, etc.)
- No client is connected directly to any other client

MQTT Broker

- The broker is primarily responsible for receiving all messages, filtering them, decide who is interested in it and then sending the message to all subscribed clients. (From HiveMQ MQTT Essentials)
- May authenticate and authorize clients.
- Maintains sessions and missed messages
- Maintains a hierarchical namespace for topics and allows subscribers (but not publishers) to use wildcards (+ and #).

Topics are organized into a hierarchical namespace

- Suppose a client publishes to
mm6House/Kitchen/Sensor/Temperature
- Another client might subscribe to:
mm6House/Kitchen/Sensor/Temperature
- Or, using a single level wildcard (+) we could find all children of Kitchen that have a child called Temperature
mm6House/Kitchen/+/Temperature
- Or, using a multi level wildcard (#) we can go deep
mm6House/Kitchen/#
- The octothorp (#) must be the last character.
- To see every message, subscribe to #.