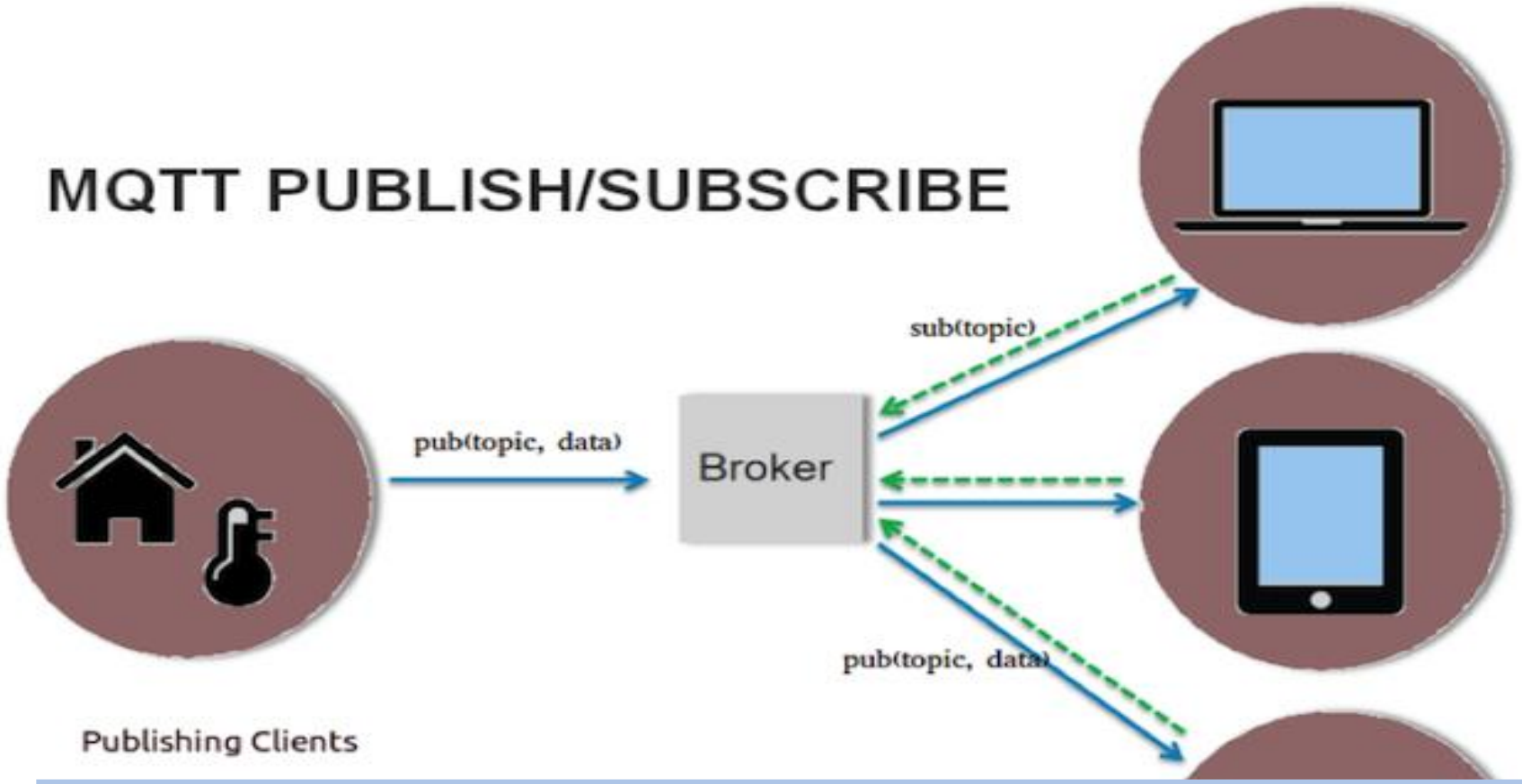


IoT Networking

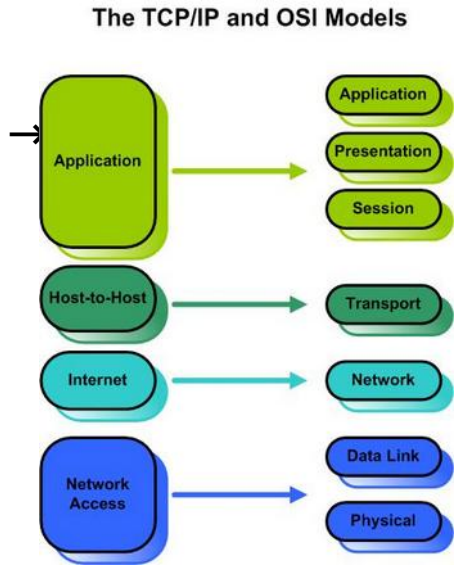
MQTT

Sebastian Büttrich 202402

Networking RECAP & MQTT dependencies

- Layer models
- **MQTT is an application**

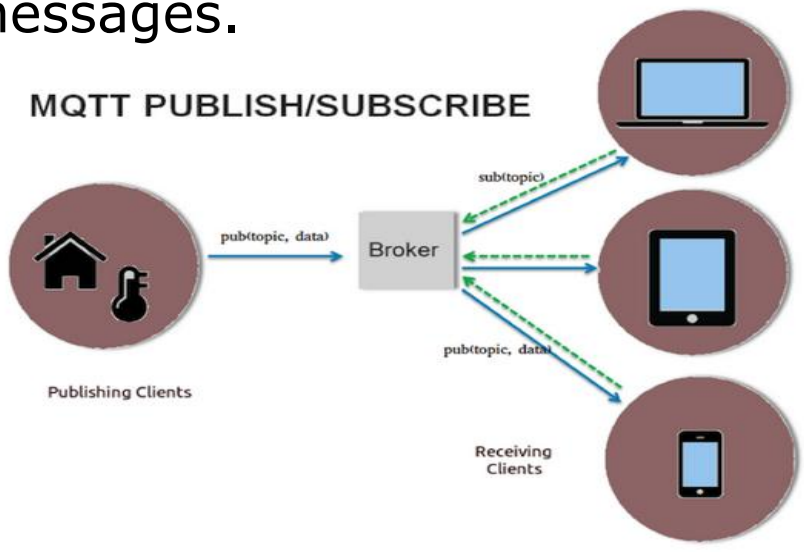
  and depends on having
- **TCP/IP connectivity** →
  which in turn depends on having
- **physical connectivity.** →

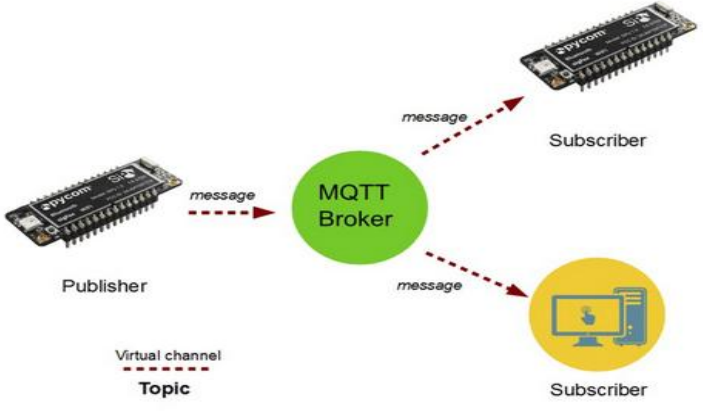


There is a variation called MQTT for Sensor Networks, which does not need TCP/IP, but that s later ...

MQTT / 1

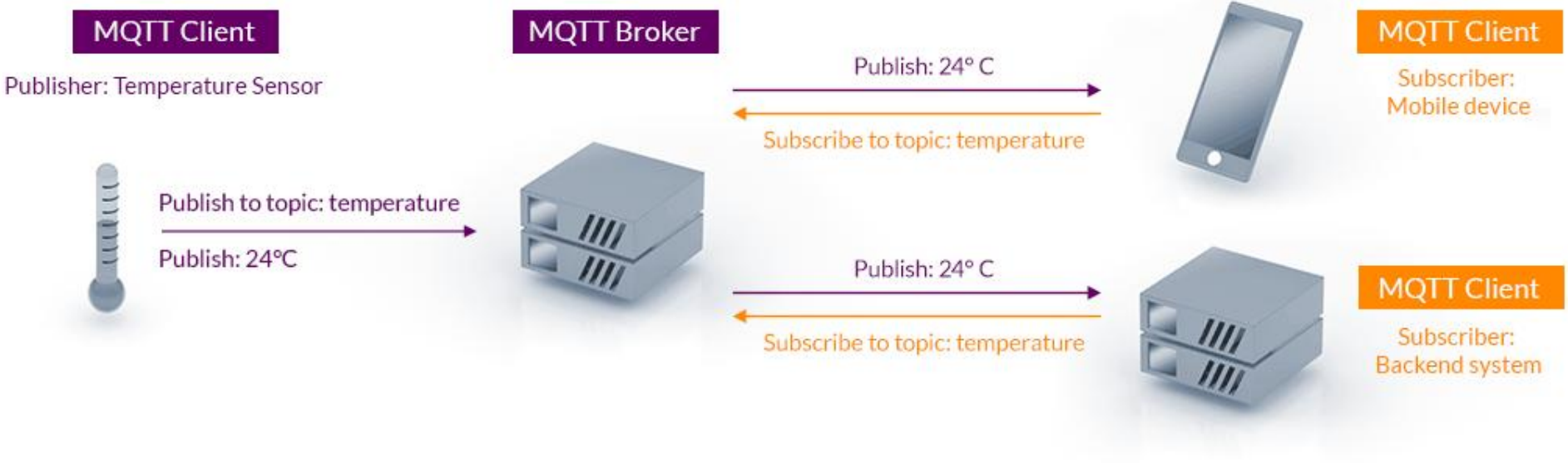
MQTT (Message Queuing Telemetry Transport) is a **publish-subscribe**-based messaging protocol.
Works on top of **TCP/IP**.
A Message broker or server receives and redistributes messages.





MQTT / also 1

MQTT (Message Queuing Telemetry Transport) is a **publish-subscribe**-based messaging protocol.
Works on top of **TCP/IP**.
A Message broker or server receives and redistributes messages.



MQTT / 2 / topics

MQTT publishing and subscription is organized by topics:

e.g.
/house/light
Or
/greenhouse/temperature

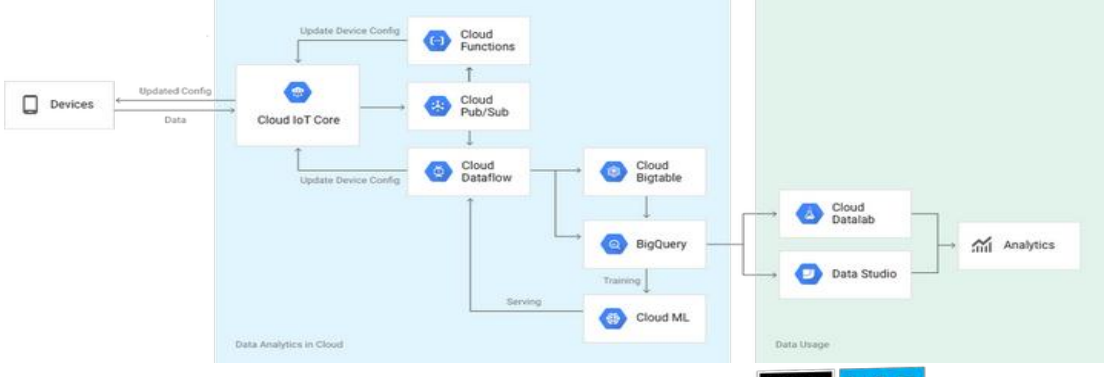
(case sensitive!)

Messages are in **free format,** however, often you will see e.g. json or xml messages.
Some service might restrict message formats.

MQTT / 3 / adoption

MQTT is a widely accepted and deployed standard in services/platforms such as

- Amazon AWS IoT
- Microsoft Azure
- IBM Watson for IoT
- Facebook Messenger (to an unknown degree)
- Google Cloud IoT Core is able to support data “from millions of globally dispersed devices.” Like similar services, Cloud IoT Core supports the standard MQTT and HTTP protocols for talking to devices.



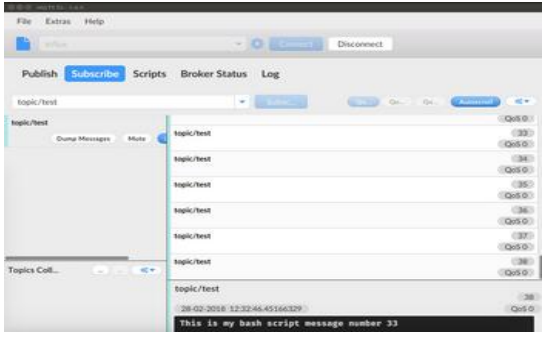
MQTT / 4 / usage

MQTT clients exist for a wide variety of platforms and languages:
https://github.com/mqtt/mqtt.github.io/wiki/software?id=software
https://www.hivemq.com/blog/seven-best-mqtt-client-tools

e.g
MQTT.fx
(available for Win/MacOSX/Linux, http://www.jensd.de/apps/mqttfx/, free)
mqtt-spy
(based on Java 8, http://kamilfb.github.io/mqtt-spy/ , OpenSource)
mosquitto_tools
(commandline, for Win/MacOSX/Linux, https://mosquitto.org/download/ , OpenSource)

And of course for

Arduino (C)
Raspberry (== Debian Linux)
Pycom (micropython)



MQTT / 5 / servers (aka brokers)

A wide choice of servers is available, many of them open source:
https://github.com/mqtt/mqtt.github.io/wiki/servers

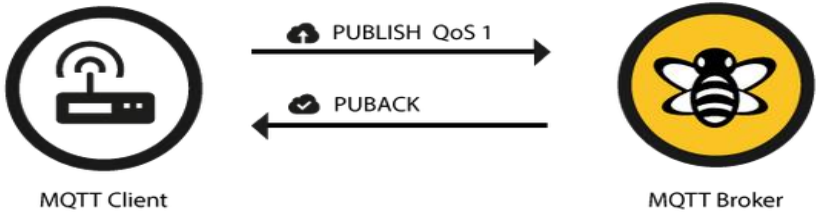
influx.itu.dk runs a public mosquitto server.

MQTT / 6 / QoS

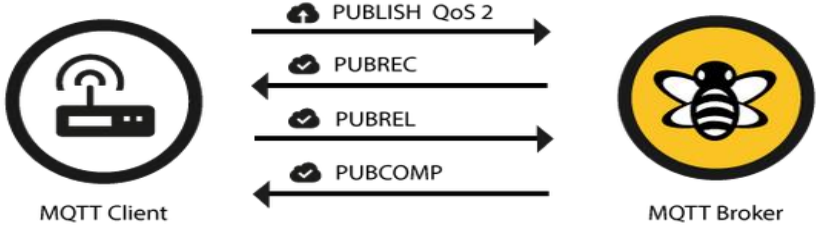
MQTT implements 3 levels of QoS:

At most once (0)
Just send it

At least once (1)
Send and confirm



Exactly once (2)
Send, confirm, reference, stop.



MQTT / 7 / Last Will

MQTT Last Will and Testament

The Last Will and Testament (LWT) feature is used in MQTT to notify other clients about an ungracefully disconnected client. Each client can specify its last will message (a normal MQTT message with topic, retained flag, QoS and payload) when connecting to a broker. The broker will store the message until it detects that the client has disconnected ungracefully, and in tat case, send it out to subscribers.

Why is this important?

In a lean communications protocol, dependent devices and services need to know about and have some chance to react to the disappearance of devices.

MQTT / 8 / security

MQTT may be used encrypted or unencrypted - here is mosquittos standard ports:

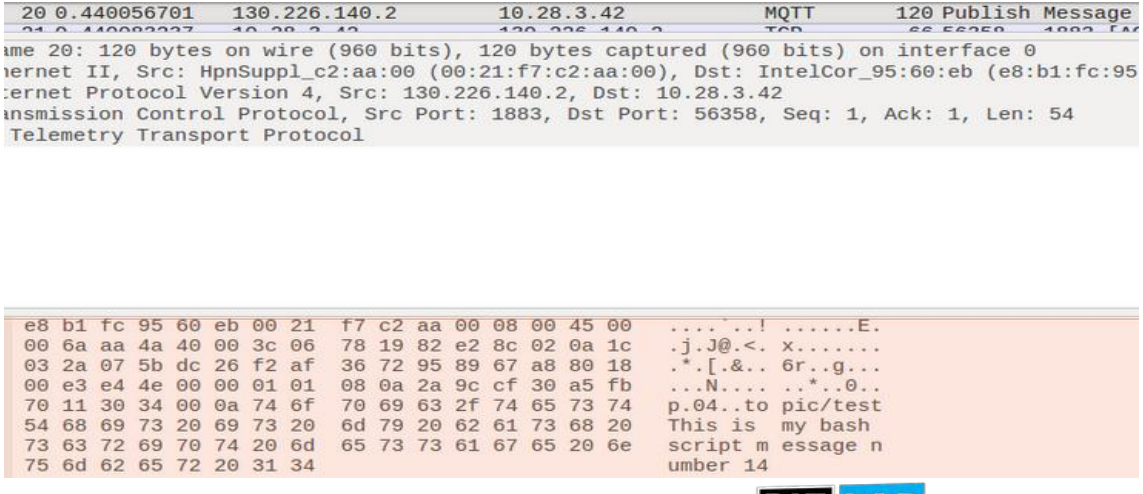
1883 : MQTT, unencrypted

8883 : MQTT, encrypted

8884 : MQTT, encrypted, client certificate required

8080 : MQTT over WebSockets, unencrypted

8081 : MQTT over WebSockets, encrypted



MQTT / 9 / security

MQTT with TLS/SSL works very much the same as https (which we are familiar with from web usage).
Example with letsencrypt certificates:

```
/etc/mosquitto/conf.d/ssl.conf
listener 8883
certfile /etc/letsencrypt/live/influx.itu.dk/cert.pem
cafile /etc/letsencrypt/live/influx.itu.dk/chain.pem
keyfile /etc/letsencrypt/live/influx.itu.dk/privkey.pem
```

A mosquitto_pub client would publish like this:

```
mosquitto_pub -h influx.itu.dk -p 8883 --capath
/etc/ssl/certs/ -t topic/test -m "encrypted msg"
```

MQTT / 10 / security

MQTT with TLS/SSL works very much the same as https.

In addition to “web style” TLS/SSL, specific client certificates can be demanded.

Username / password protection is also available.

MQTT / 11 / security note

Note:

If devices publish encrypted data,
but the broker allows subscribers to listen unencrypted,
data will be readable on the network!

MQTT / 12 / MQTT-SN

MQTT for Sensor Networks is “aimed at embedded devices on non-TCP/IP networks, such as Zigbee. MQTT-SN is a publish/subscribe messaging protocol for wireless sensor networks (WSN), with the aim of extending the MQTT protocol beyond the reach of TCP/IP infrastructure for Sensor and Actuator solutions.”

It makes a lean protocol even leaner.

More via https://mqtt.org/mqtt-specification/
https://www.oasis-open.org/committees/download.php/66091/MQTT-SN_spec_v1.2.pdf

C/C++
https://github.com/eclipse/paho.mqtt-sn.embedded-c
python client
https://pypi.org/project/mqttsn/

Take-Aways, MQTT

Be able to describe

the dependencies & main features of

MQTT

… and most importantly: use it.