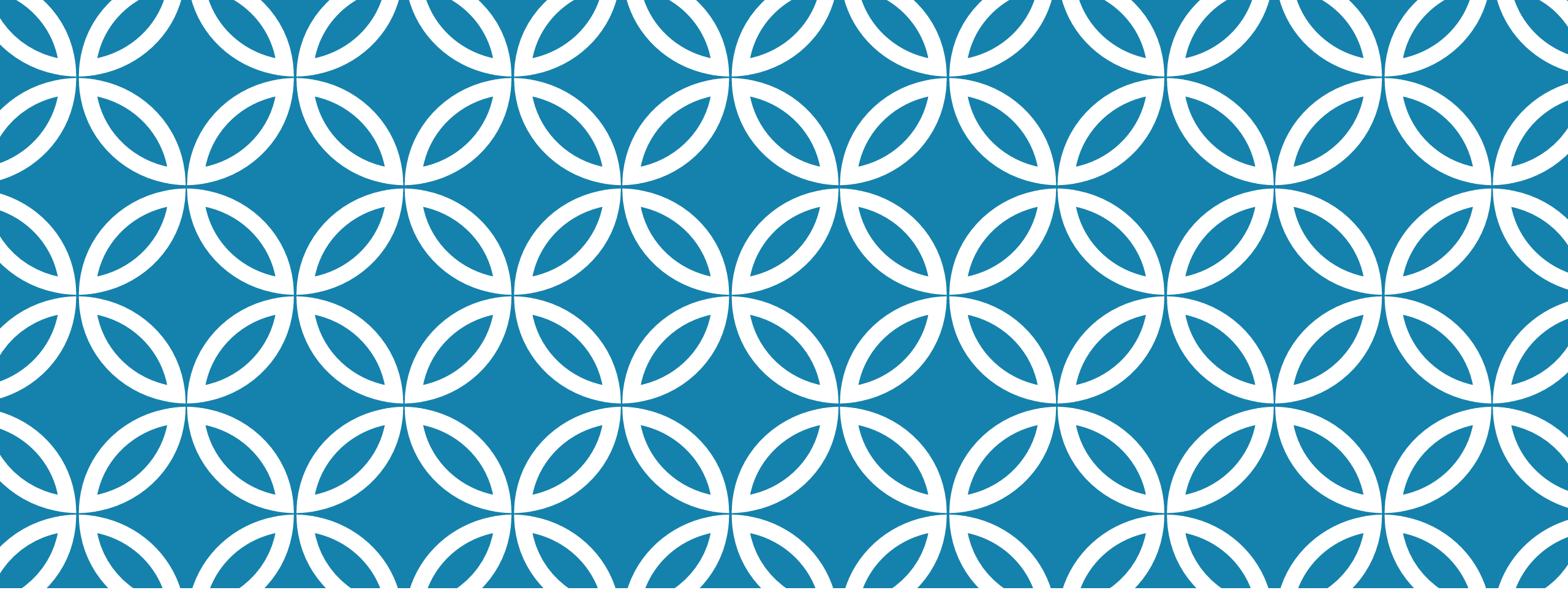


MQTT

MQ Telemetry **T**ransport
(for **I**nternet **o**f **T**hings)

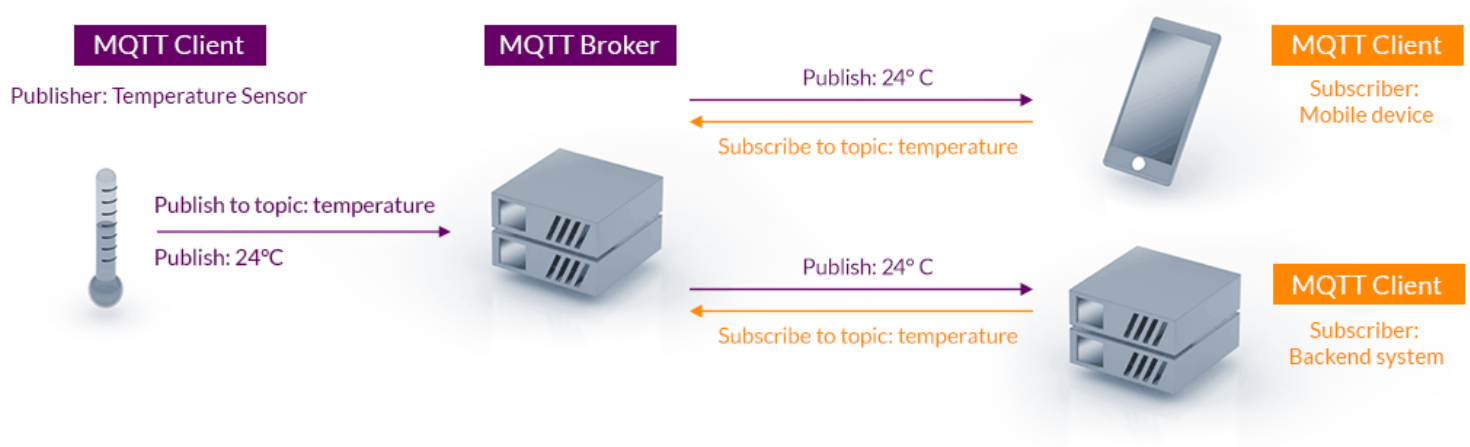
MQTT

https://mqtt.org

Introduced by IBM

Standardized by OASIS (2014)

Uses Publish/Subscribe mechanism controlled by Broker



Broker

- Software component
- Responsible for distributing messages from Publishers to interested Subscribers

MQTT FUNDAMENTALS

Many – to – many Sub to Pub relationship

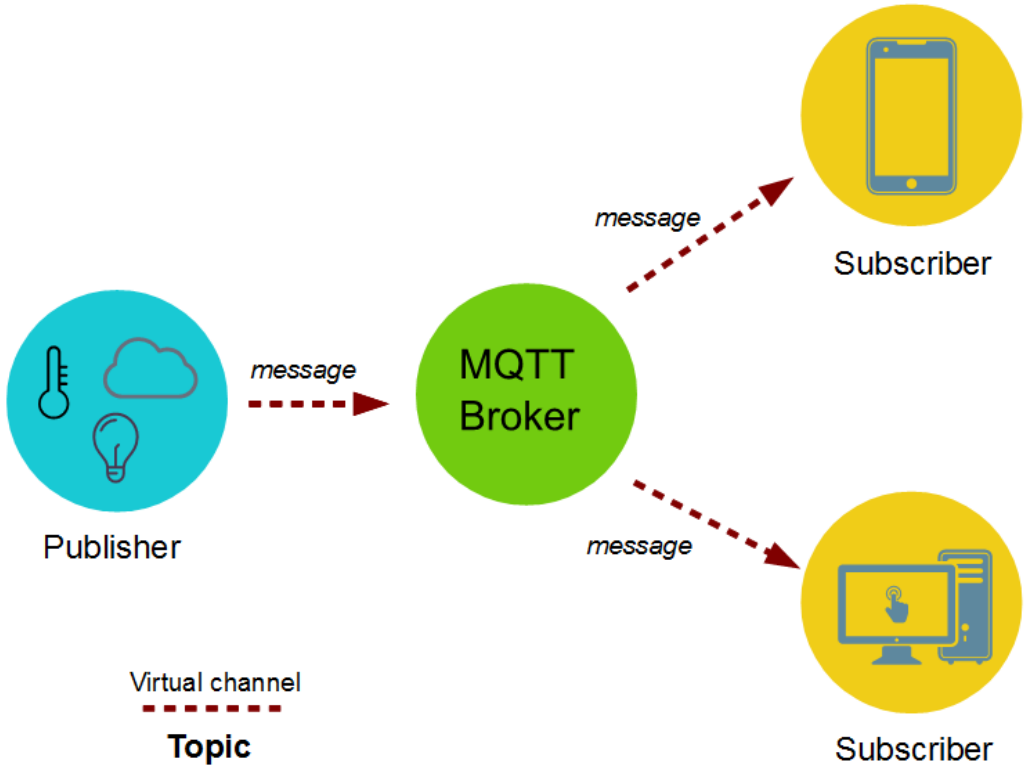
One Broker for every system

Subs authenticated to Broker

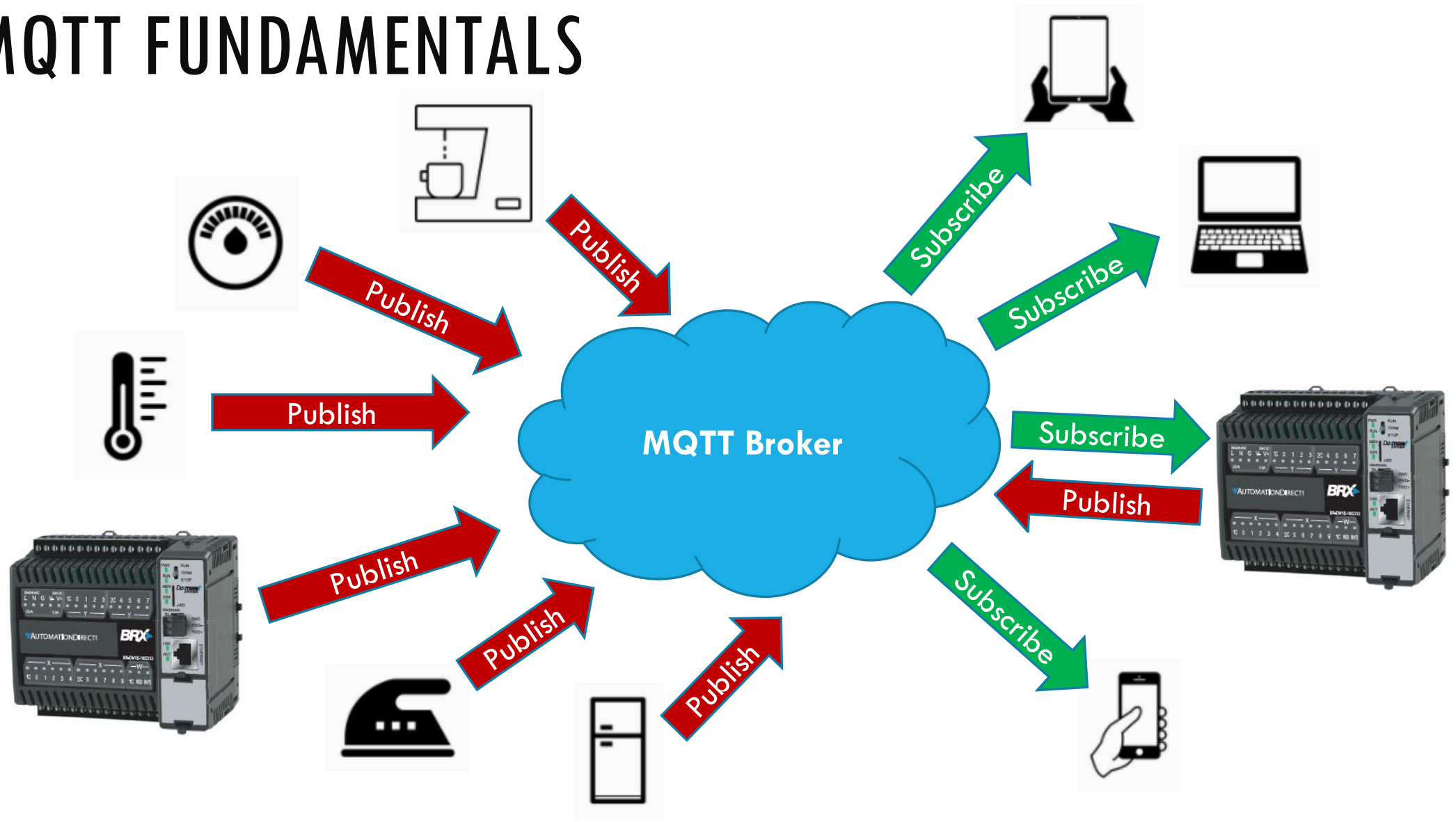
Subs/Pubs can be very constrained

Pub can be even only a sensor

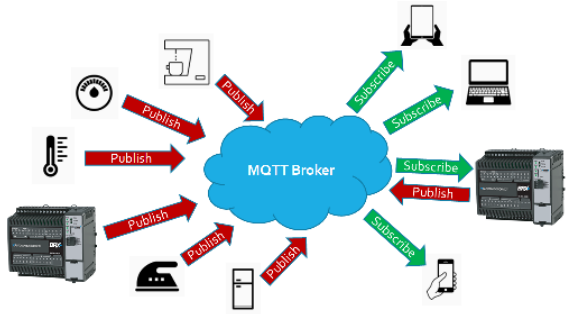
Broker has to provide more computational power



MQTT FUNDAMENTALS



MQTT TERMINOLOGY (1 OF 2)



MQTT Broker

- Receives published topics
- Distributes topics to subscribers
- Keeps Client connections alive
- Sends Last Will & Testament (LWT) to subscribers if a Client "ungracefully disconnects"

MQTT Client

- Can publish topic(s), keep-alive time, Retain bit, QoS, Last Will & Testament
- Can subscribe to topic(s)

Topic

- Name of the data

Payload

- Actual data

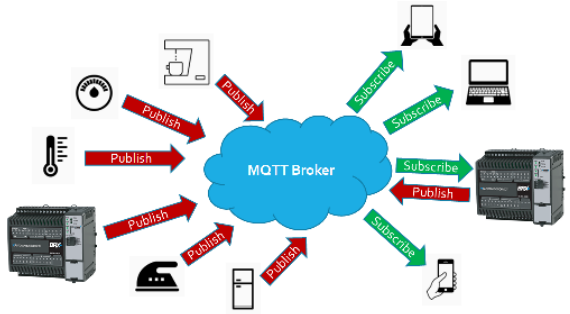
Message

- Topic + Payload

QoS *(Quality of Service)*

- 0 = At most once *(BRX always, publish & subscribe)*: transmits message once (relies on TCP)
- 1 = At least once : transmits message until it is acknowledged by receiver (may receive more than one)
- 2 = Exactly once: transmits message, needs "received" message, asks if it can be "released," needs "complete" message

MQTT TERMINOLOGY (2 OF 2)



Publish

- To send a Topic w/Payload to MQTT Broker

Subscribe

- To request a Topic w/Payload update from MQTT Broker

Retain

- Asks MQTT Broker to save the Topic w/Payload even after sending it to all the subscribing Clients

Keep-alive Time

- How often Broker "pings" client to see if he's there

Last Will & Testament *(LWT)*

- Topic w/Payload initially sent by an MQTT Client to the MQTT Broker for the Broker to send to other Clients if he is "ungracefully disconnected"

MQTT DATA EXCHANGE

- Publishers are fundamentally separate from Subscribers
  - Publishers only care about getting data to Broker
  - Broker is fully responsible for getting data to Subscribers

- Clients connect to an MQTT Broker (TCP/IP, MQTT)
- Clients can publish data to topics, e.g.
  - host/office/upssitech/temperature, 23.3
- Clients subscribe to topics, e.g.
  - host/office/upssitech/temperature

***NOTE**: MQTT supports wildcards for topics*

- Clients receive (from Broker) all data published to topics they subscribe to
- Data can be anything