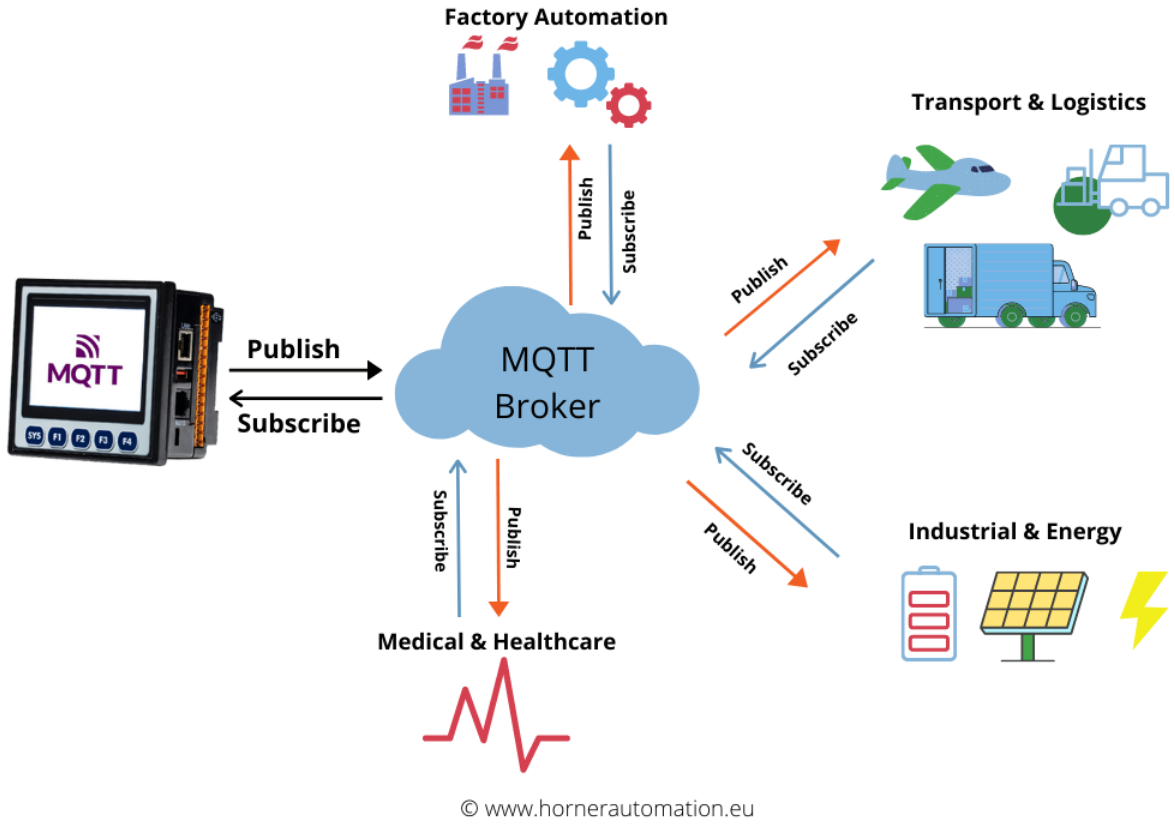
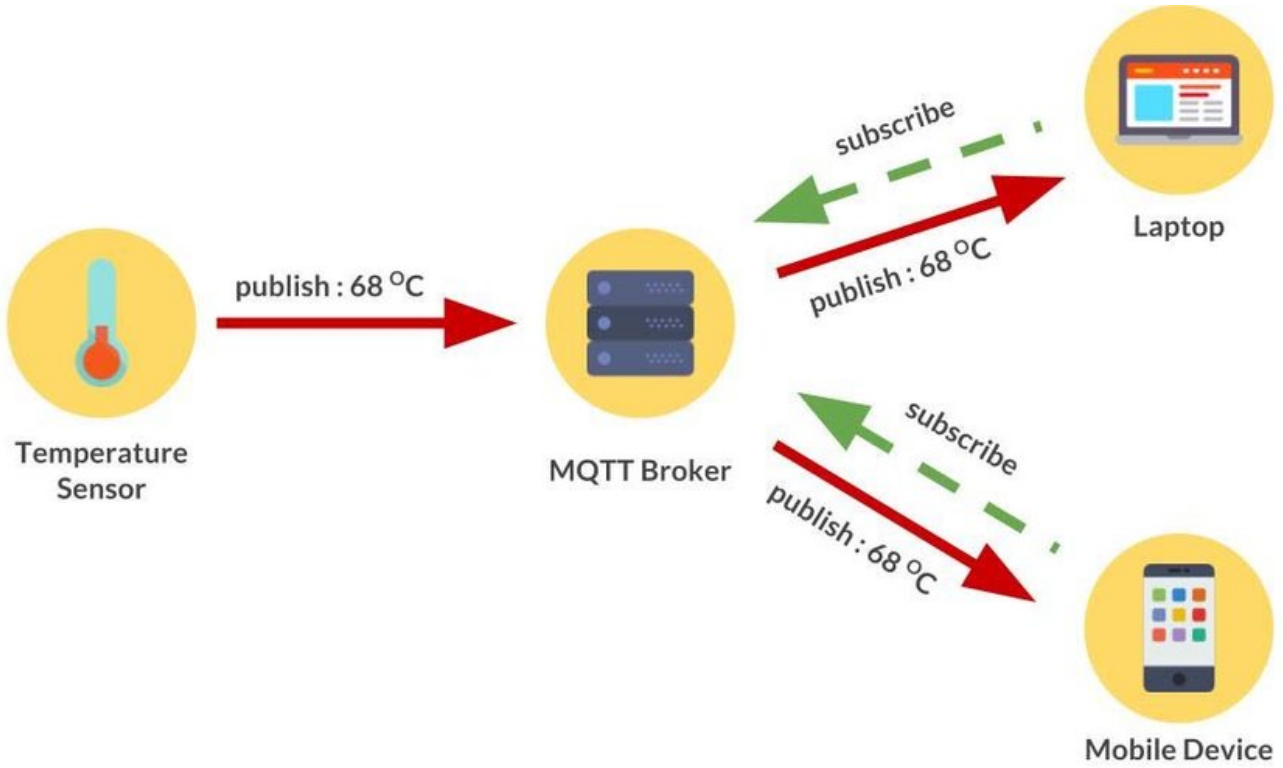
MQTT : Message Queuing Telemetry Transport

- Light weight, Pub-Sub model for machine-to-machine communication
- 1999, Created by Andy-Stanford-Clark from IBM and Arlen Nipper from Eurotech.
- 2011, IBM and Eurotech donated it to Eclipse project called paho.
- 2013, submitted to OASIS for standardization.



Features

- pub/sub architecture:
  - one broker, multiple clients
  - Decoupled publisher and subscribe
  - All clients can publish and subscribe
- Topics-based
- Built over TCP/IP layer
- Quality of Service
  - QoS 0
  - QoS 1
  - QoS 2
- Last will and testament
  - Topic
  - QoS
  - Message
  - Retain



Message Structure and Topic

Topic tree when subscribing

- UM/campus/department/news
- UM/campus/faculty/forum
- UM/campus/research/news
- UM/dearborn/student/registration
- UMD

Single level hierarchy “+”

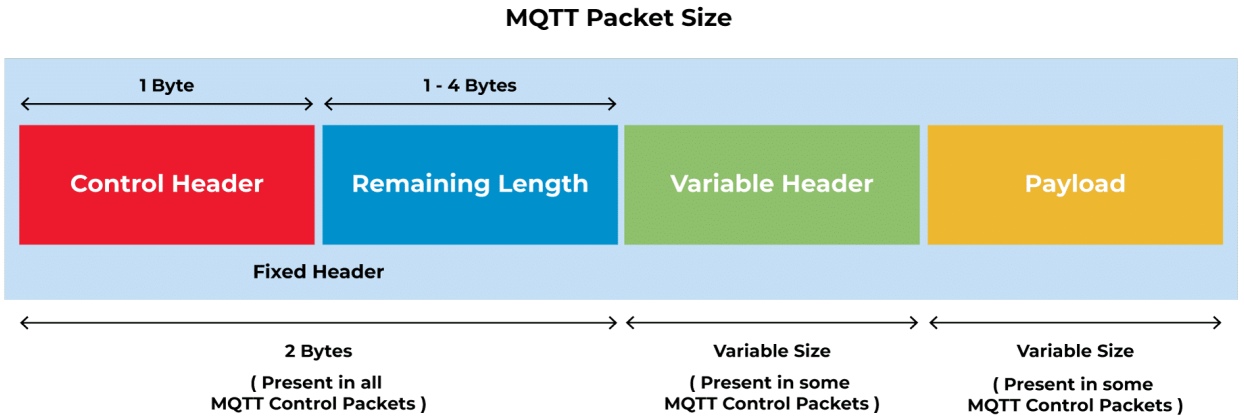
- Eg: UM/campus/+/news

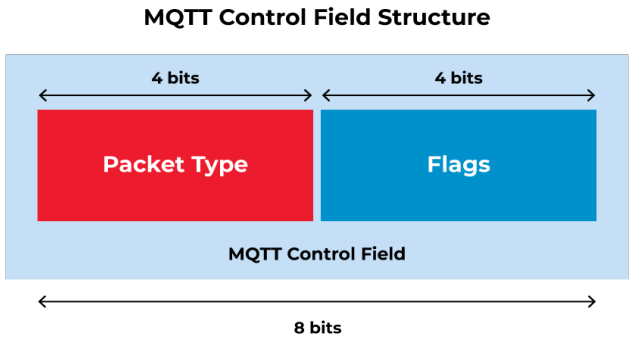
Remaining level hierarchy “#”

- Eg: UM/#

Reserved topics start with “$”

- Eg: $SYS





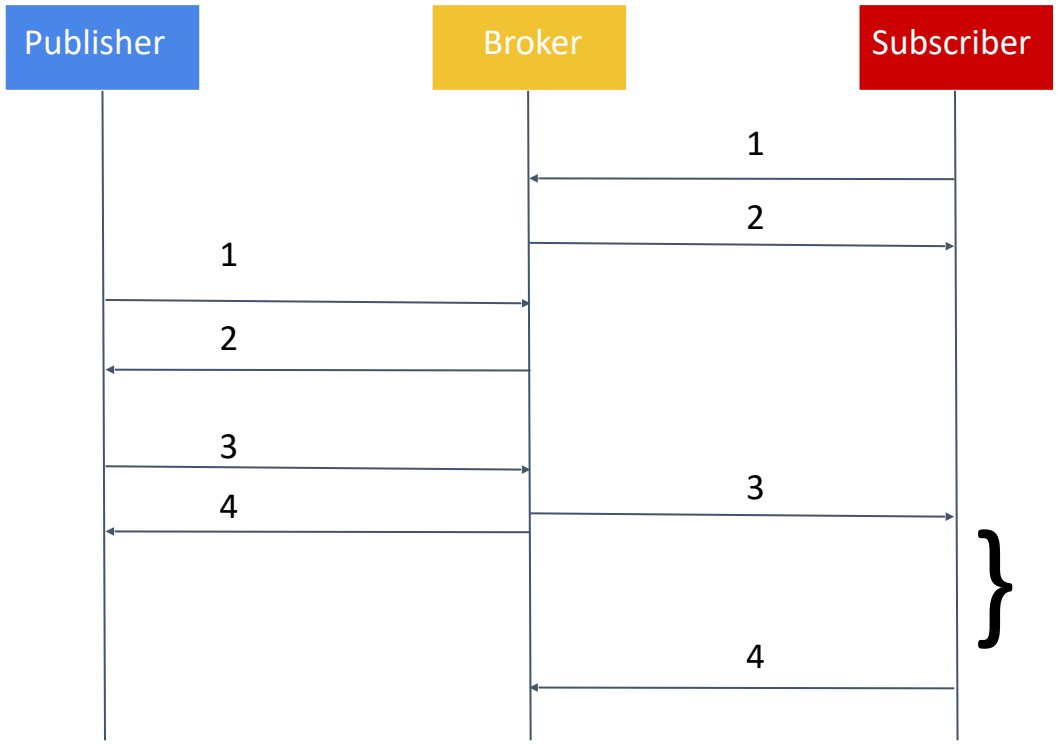
Message exchange

Broker at port 1883, and secure-mqtt 8883

```
mosquitto_pub -h <IP address of the broker> -t 'test' -m 'message'
mosquitto_sub -h <IP address of the broker> -t 'test' -d
```

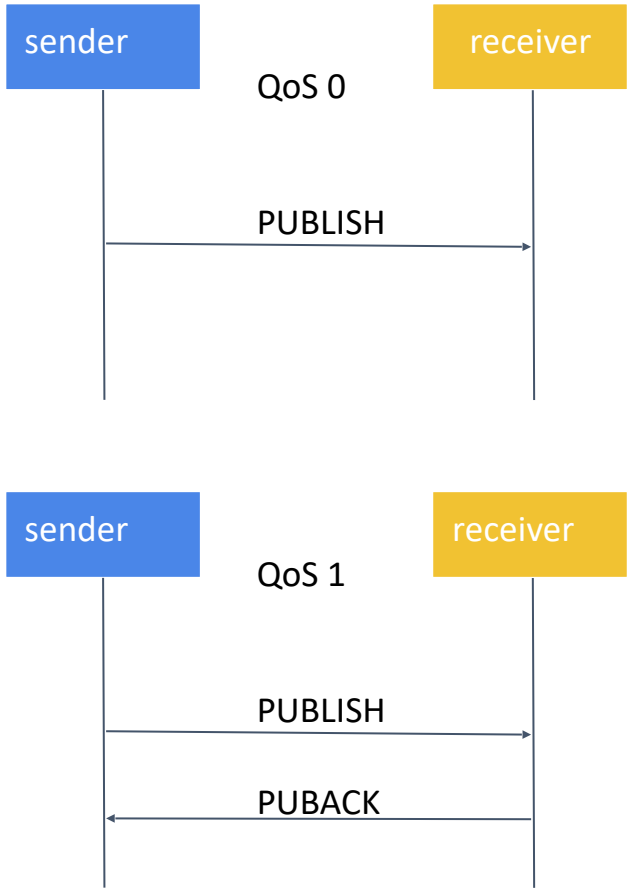
Basic message exchange procedure:

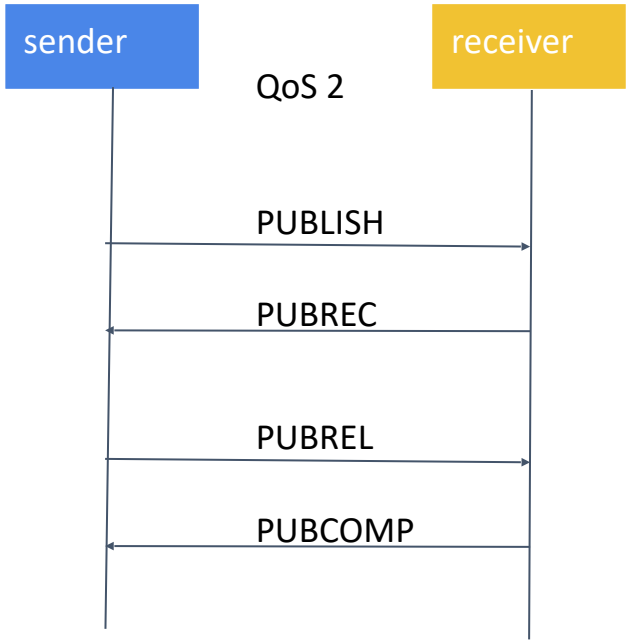
1. Connection
2. Authentication & Acknowledgement
3. Communication
4. Termination



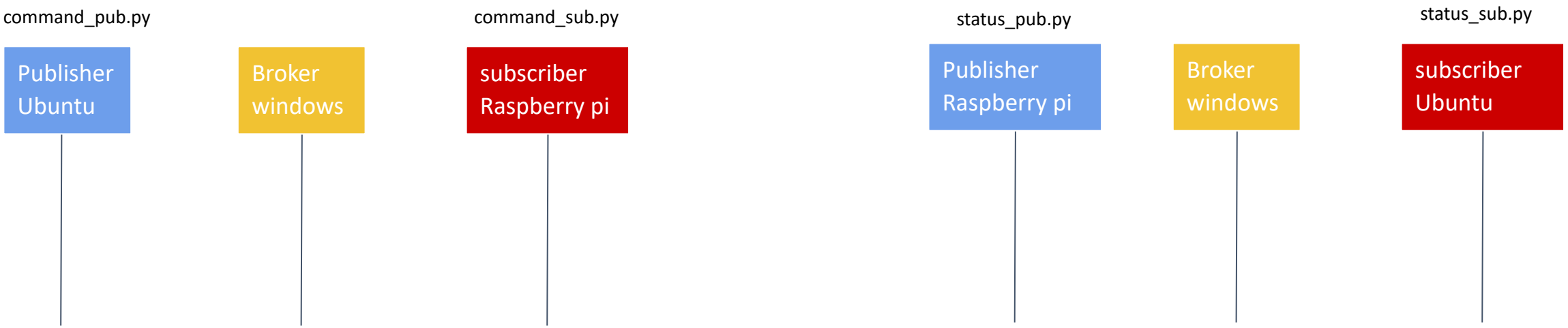
Quality of Service : Message Reliability

- QoS 0: Default
  - Fire and forget
  - Message is deleted.
  - No duplicates
- QoS 1:
  - At most once
  - message is stored.
  - Duplicates possible
- QoS 2:
  - Exactly once
  - message is stored.
  - No duplicates





Demo


command_pub.py
Publisher
Ubuntu
Broker
windows
command_sub.py
subscriber
Raspberry pi
status_pub.py
Publisher
Raspberry pi
Broker
windows
status_sub.py
subscriber
Ubuntu