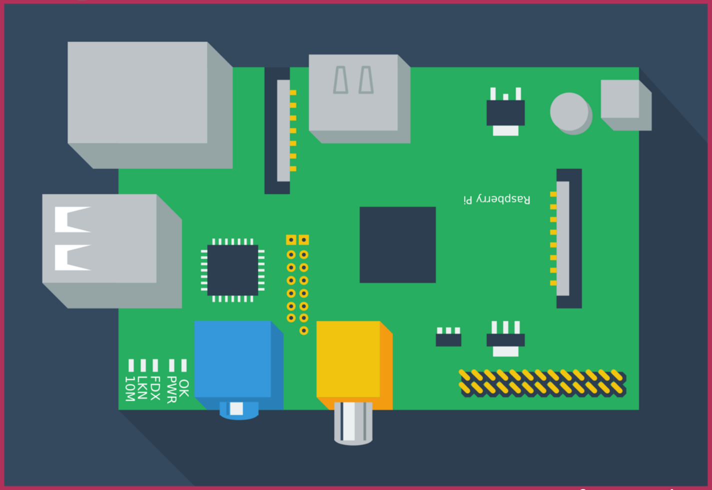
MQTT on Pi

OCT 2021



Safyzan Salim
scriptworkz ent

Why MQTT?

Lightweight and Efficient

MQTT clients are very small, require minimal resources so can be used on small microcontrollers. MQTT message headers are small to optimize network bandwidth.

Bi-directional Communications

MQTT allows for messaging between device to cloud and cloud to device. This makes for easy broadcasting messages to groups of things.

Scale to Millions of Things

MQTT can scale to connect with millions of IoT devices.

Reliable Message Delivery

Reliability of message delivery is important for many IoT use cases. This is why MQTT has 3 defined quality of service levels: 0 - at most once, 1- at least once, 2 - exactly once

Support for Unreliable Networks

Many IoT devices connect over unreliable cellular networks. MQTT's support for persistent sessions reduces the time to reconnect the client with the broker.

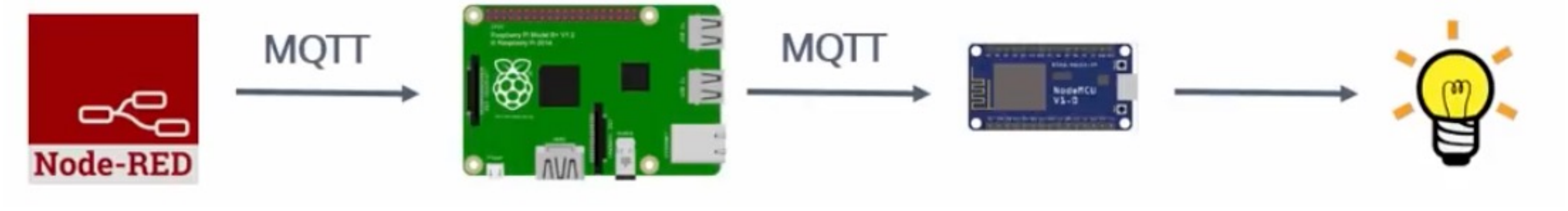
Security Enabled

MQTT makes it easy to encrypt messages using TLS and authenticate clients using modern authentication protocols, such as OAuth.

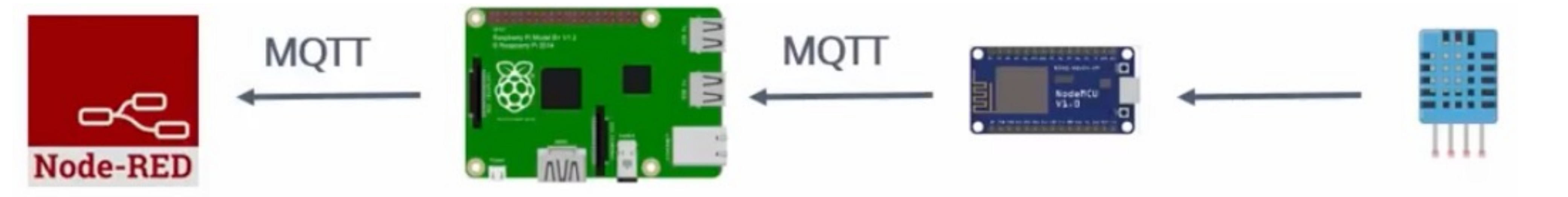
https://mqtt.org

The Operation of MQTT

i. Send a command to control an output



ii. Read and publish data



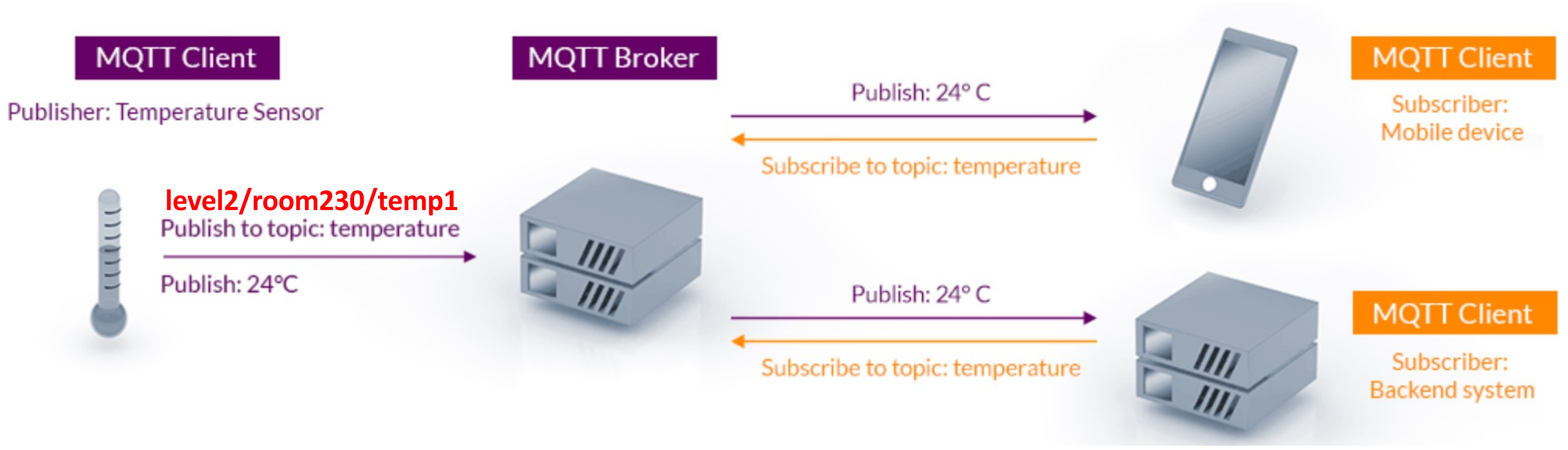
https://randomnerdtutorials.com/what-is-mqtt-and-how-it-works/

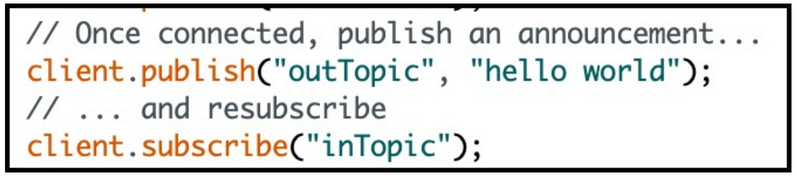
Basic MQTT Components

- Broker
- Publish/Subscribe
- Topics
- √ Quality of Service

https://www.youtube.com/watch?v=QSwR-JMmNOo

MQTT Publish/Subscribe Architechture


MQTT Client
Publisher: Temperature Sensor
level2/room230/temp1
Publish to topic: temperature
Publish: 24°C
MQTT Broker
Publish: 24° C
Subscribe to topic: temperature
MQTT Client
Subscriber: Mobile device
Publish: 24° C
Subscribe to topic: temperature
MQTT Client
Subscriber: Backend system



```
// Once connected, publish an announcement...
client.publish("outTopic", "hello world");
// ... and resubscribe
client.subscribe("inTopic");
```

Publish = Broadcast
Subscribe = Receive
Broker = Server

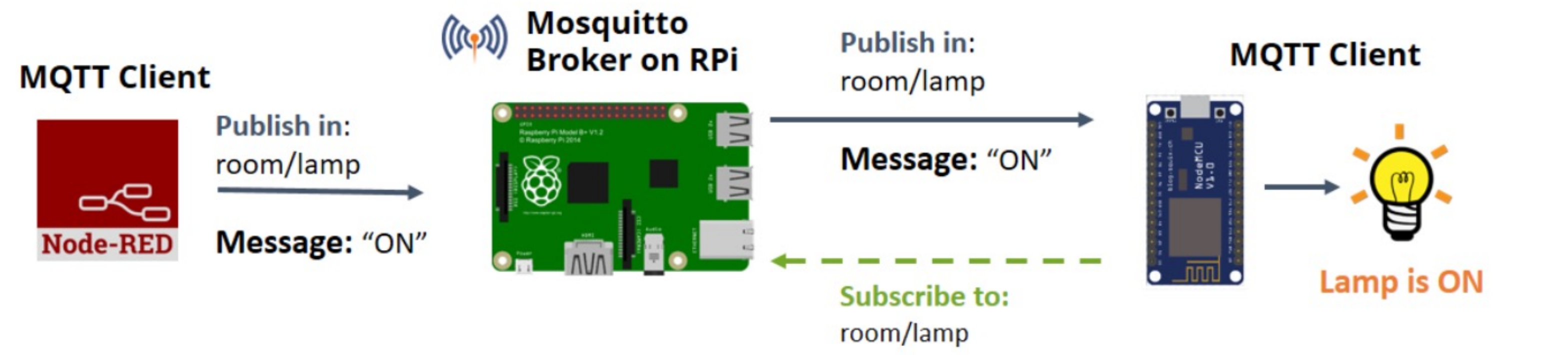
https://mqtt.org

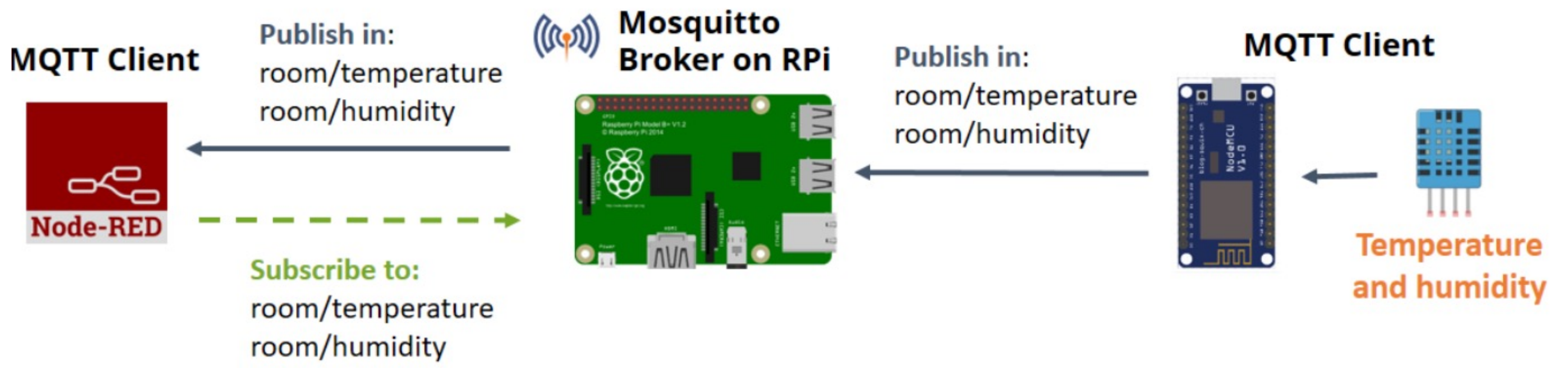
Important Point to Note

- Clients **do not have addresses** like in email systems, and messages are not sent to clients.
- Messages are **published to a broker on a topic**.
- The job of an MQTT broker is to **filter messages** based on topic, and then **distribute them to subscribers**.
- A client can receive these messages by subscribing to that topic on the same broker
- There is **no direct connection** between a publisher and subscriber.
- **All clients** can publish (broadcast) and subscribe (receive).
- MQTT brokers do not normally store messages.

http://www.steves-internet-guide.com/mqtt-works/

Example





https://randomnerdtutorials.com/what-is-mqtt-and-how-it-works/

QUESTIONS

END