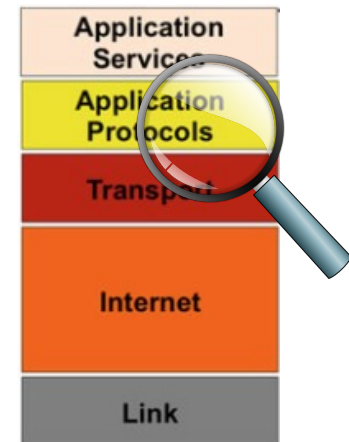


Ch. 13 - IoT Application Protocol Layer

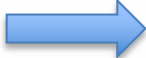
Sec 4 – MQTT Protocol

COMPSCI 147

Internet-of-Things; Software and Systems



APP PROTOCOLS FOR IOT - STANDARDIZATION

- **HTTP**
 - IETF standard (RFC 2616 is HTTP/1.1)
- **CoAP**
 - IETF standard (RFC 7252)
- **XMPP**
 - IETF standard (RFC 6272)
-  • **MQTT**
 - **OASIS standard**
- **AMQP**
 - OASIS and ISO 19464 standard (1.0)
- **SIP**
 - IETF Standard (RFC 3261)
- **IEEE 1888**
 - IEEE Standard
- **DDS (RTPS)**
 - Object Management Group (OMG) Standard

MQTT



- Message Queue Telemetry Transport (MQTT)
- Lightweight messaging protocol for M2M communication
 - Low network bandwidth and small code footprint
- Invented and sponsored by IBM.
 - MQ originated from “message queuing (MQ)” architecture
 - There is no queueing in MQTT.
 - Now Open source. Open-source libraries are available.
- Telemetry = Tele-Metering = Remote measurements
- Telemetry data goes from devices to a broker.
 - Uses a publish/subscribe mechanism.

WHO USES MQTT



- Facebook messenger uses MQTT to minimize battery usage.
FB bought “Beluga group chat” that used MQTT
- Extremely light-weight, perfect for M2M battery powered IoT devices
- Several other applications uses this protocol for medical and environmental applications.

MQTT - BROKER

- Many open-source implementations of clients and brokers are available
 - Really small message broker (RSMB): C
 - Mosquitto
 - Micro broker: Java based for PDAs, notebooks
- Simple to implement
- Provide a Quality-of-Service Data Delivery
- Data Agnostic
- Continuous Session Awareness

MQTT Components

MQTT

Broker

Handles routing of messages

Client

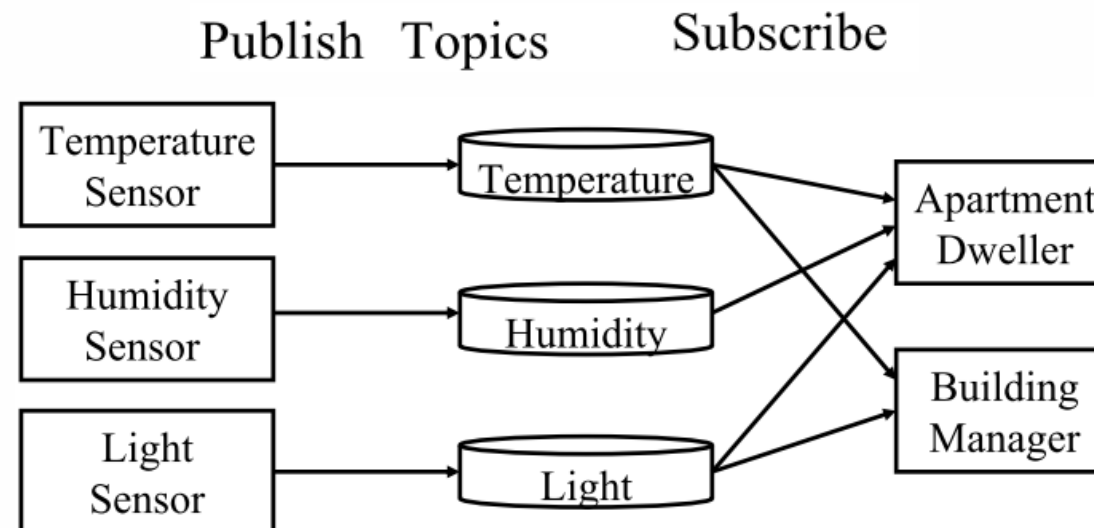
Publish and subscribe to topics they want to send/receive data to/from

Topics

Messages published to a topic are received by all subscribers

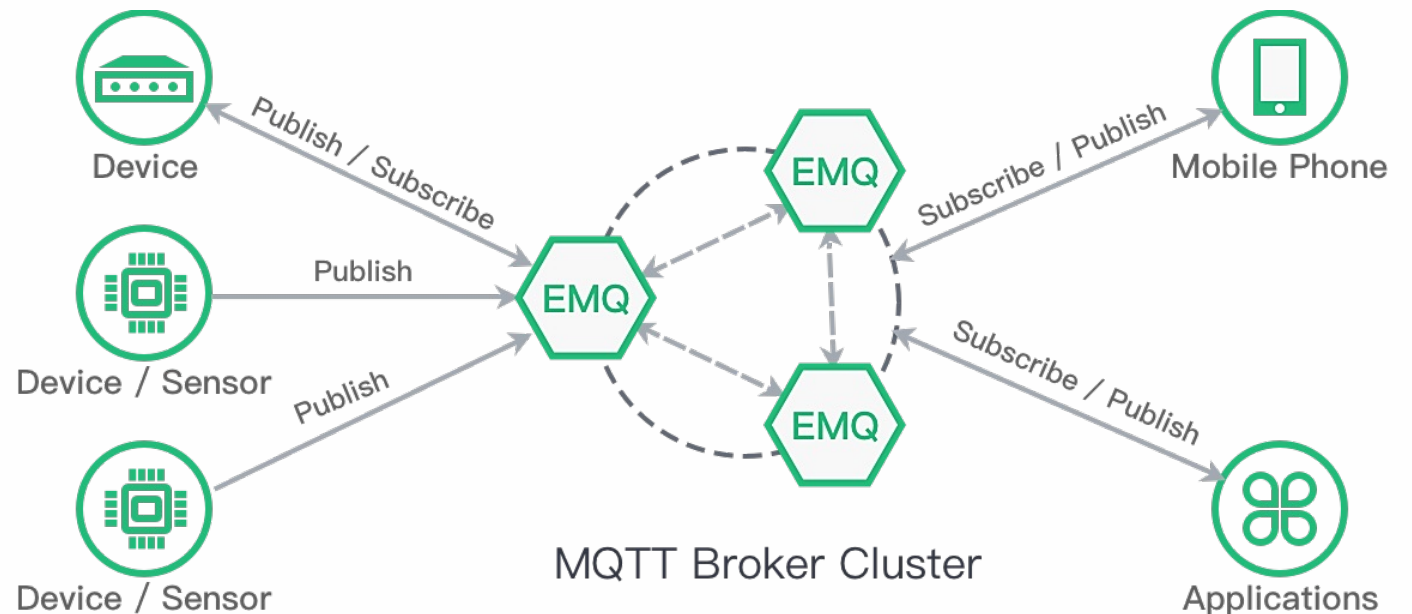
MQTT - CONCEPTS

- **Topics/Subscriptions:** Messages are published to topics. Clients can subscribe to a topic or a set of related topics
 - Topics can be hierarchical (e.g., /house/bedroom/light)
- **Publish/Subscribe:** Clients can subscribe to (multiple) topics or publish to topics.



MQTT - ARCHITECTURE

- Broker and connected Clients
 - broker receives subscriptions from clients on topics
 - broker receives messages and forward them
 - clients subscribe/publish on topics
- Brokers allow bridge configuration



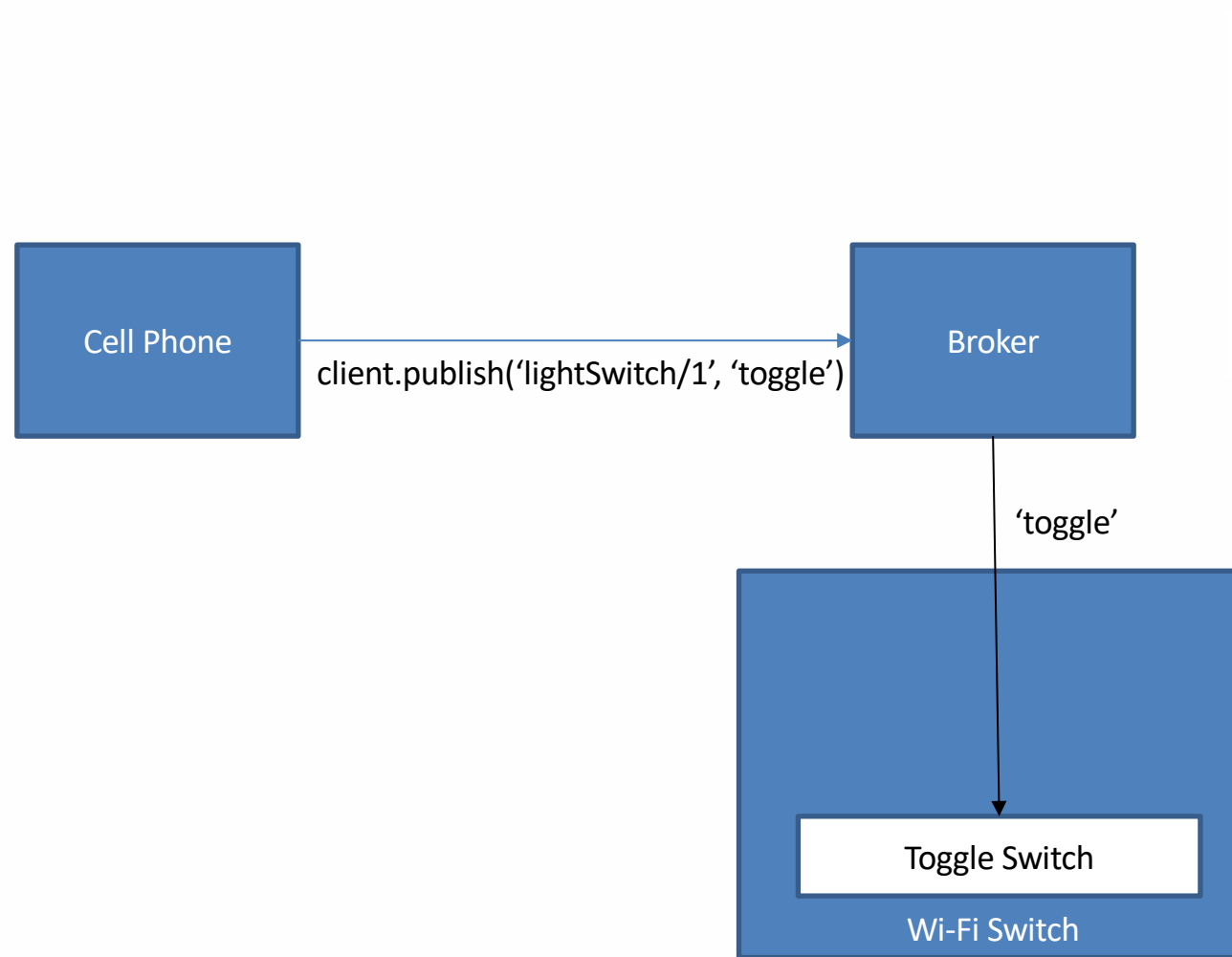
MQTT CONCEPTS II

- **Quality of Service Levels:** Three levels:
 - 0 = At most once (Best effort, No Ack // fire and forget),
 - 1 = At least once (Acked, retransmitted if ack not received),
 - 2 = Exactly once [Request to send (Publish), Clear-to-send (Pubrec), message (Pubrel), ack (Pubcomp)]
- **Retained Messages:** Server **keeps** messages even after sending it to all subscribers. New subscribers get the retained messages

MQTT CONCEPTS III

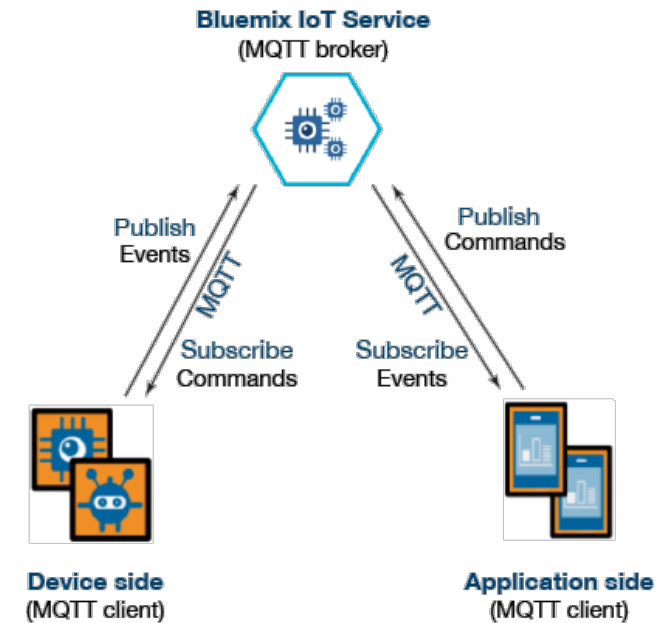
- **Persistent** (clean session) and **non-persistent** connections
 - Clean session flag
 - all subscriptions/messages are removed on disconnect
 - Otherwise, subscriptions/messages remain in effect after disconnection
 - Subsequent messages with high QoS are stored for delivery after reconnection
- **Wills:** At connection, a client can inform that it has a will or a message that should be published if unexpected disconnection
 - Brokers send an alarm if the client loses connection.
- Periodic **keep-alive** messages
 - Checking if a client is still alive
- **Topic Trees:** Topics are organized as trees using / character
 - /# matches all sublevels
 - /+ matches only one sublevel

MQTT HYPOTHETICAL LIGHT SWITCH

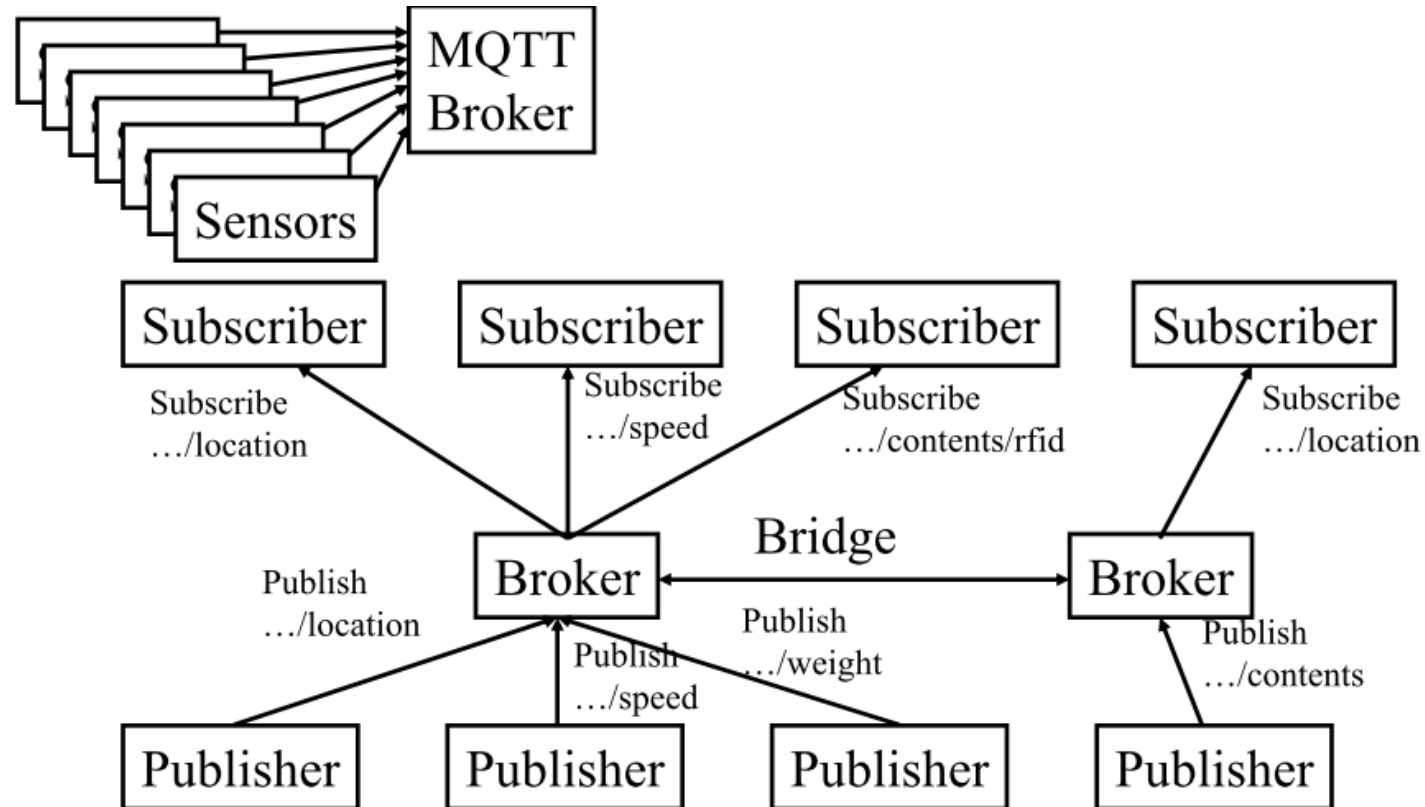


MQTT: PUB/SUB

- Clients connect to a “Broker”
- Clients subscribe to topics eg,
 - `client.subscribe('toggleLight/1')`
 - `client.subscribe('toggleLight/2')`
 - `client.subscribe('toggleLight/3')`
- Clients can publish messages to topics:
 - `client.publish('toggleLight/1', 'toggle');`
 - `client.publish('toggleLight/2', 'toggle');`
- All clients receive all messages published to topics they subscribe to
- **Messages can be anything**
 - Text
 - Images
 - etc

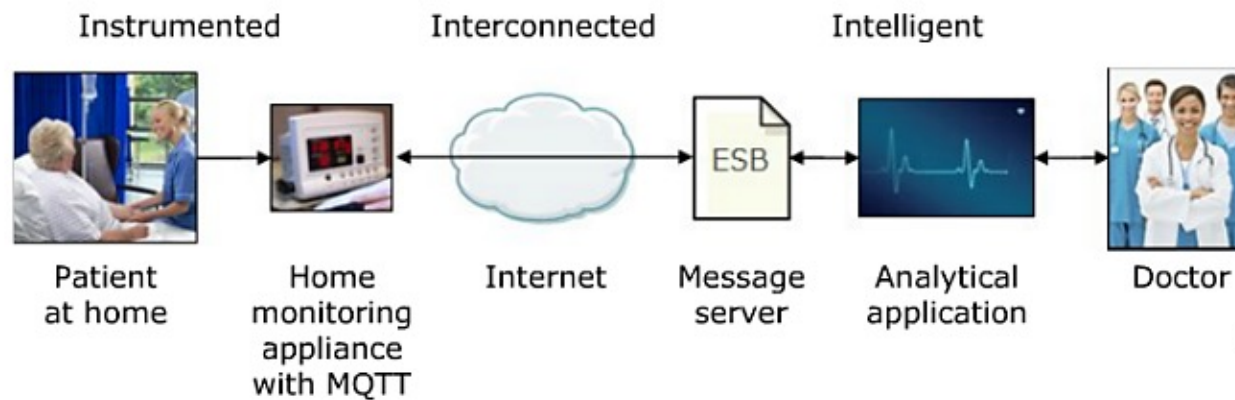


MQTT EXAMPLE



MQTT APPLICATION EXAMPLES

- Home pacemaker monitoring solution
 - Sensors on patient
 - Collected by a monitoring equipment in home (broker) using MQTT
 - Subscribed by a computer in the hospital
 - Alerts the doctor if anything is out-of-order



Source: Lampkin 2012

Demo MQTT with ESP32

- Server: <https://github.com/eclipse/mosquitto>

The screenshot displays the Mosquitto MQTT web interface. The top section, titled "Publish a packet", contains a "topic" field with the value "/home/bedroom/temperature" and a "Payload (template allowed)" field with the value "85". A blue "PUBLISH" button is located below these fields. The bottom section, titled "Listen to a topic", shows the interface after a message has been received. It displays "Listening to /home/bedroom/temperature" with a "STOP LISTENING" link. Below this, it states "Message 0 received on /home/bedroom/temperature at 1:37 PM:" followed by the message content "85". At the bottom, it shows "QoS: 0 - Retain: false".

- Client (ESP32) library:
<https://registry.platformio.org/libraries/knolleary/PubSubClient>