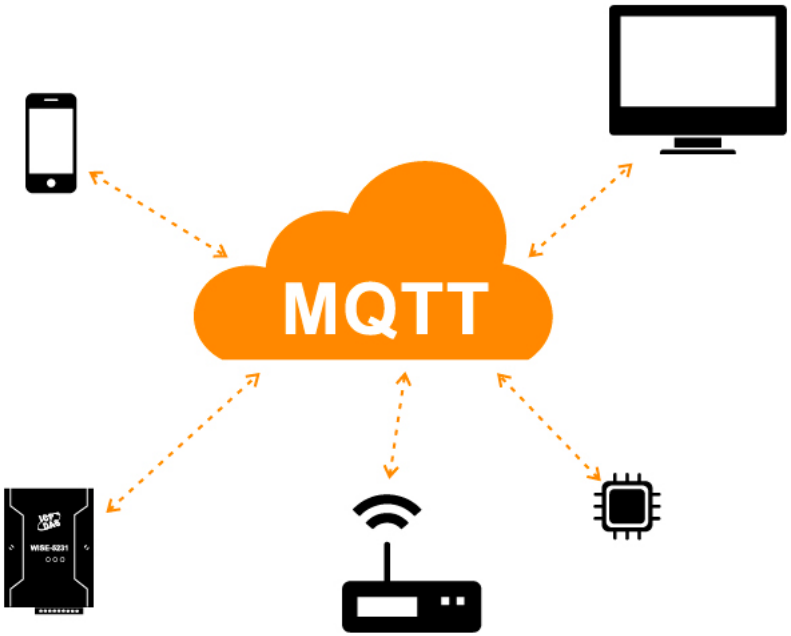
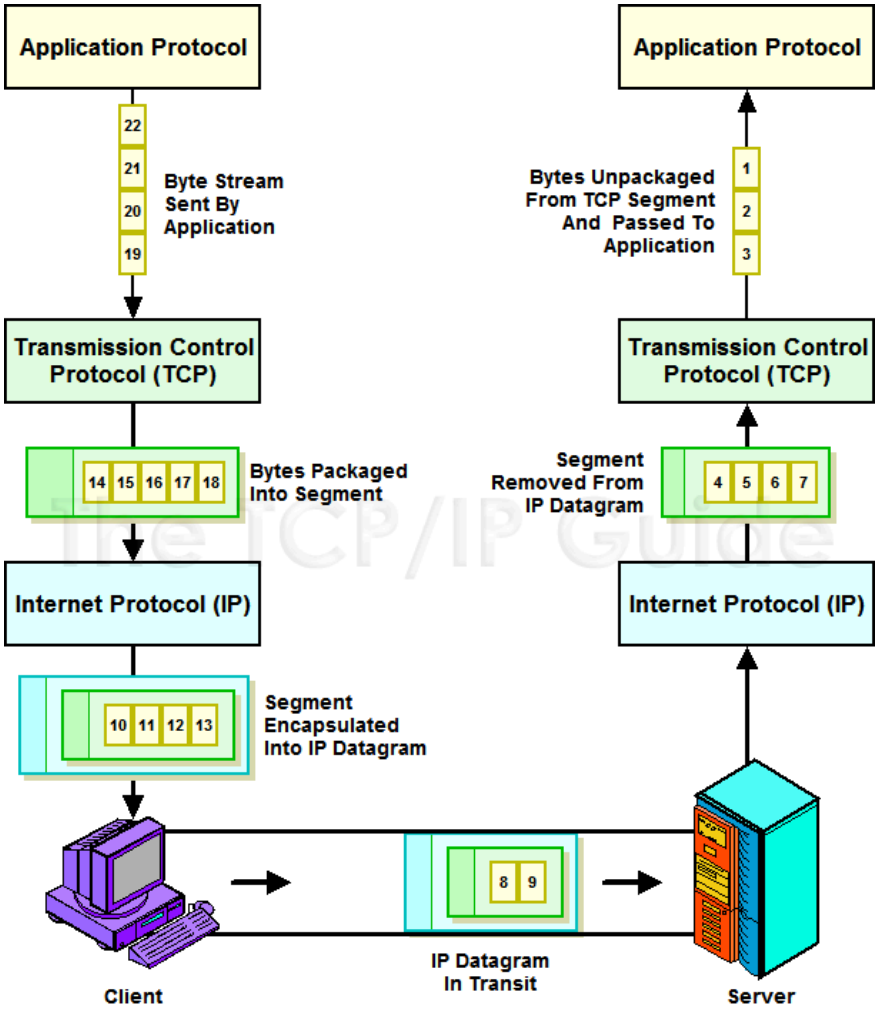
MQTT basics



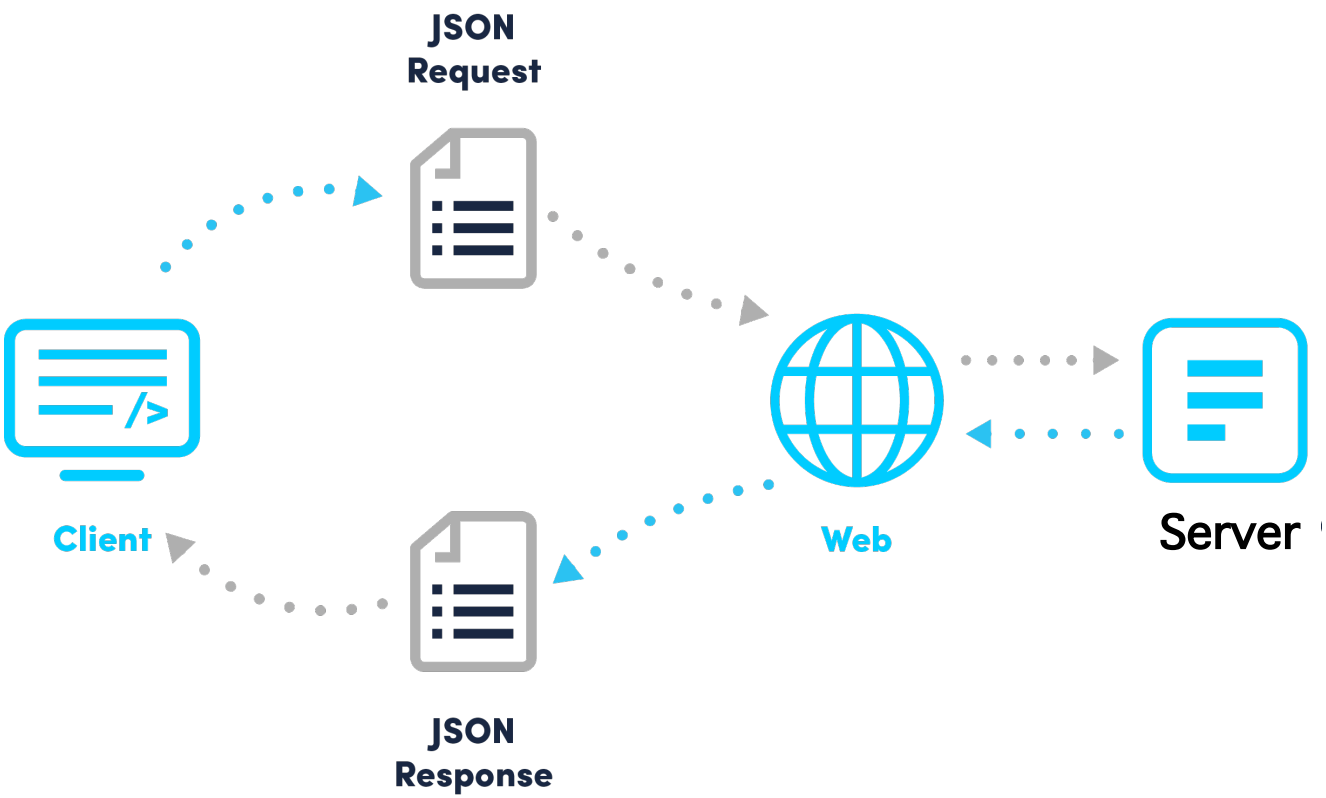
From "byte streams" to "messages"

- The "old" vision of data communication was based on reliable byte streams, i.e., TCP
- Nowadays messages interchange is becoming more common
  - E.g., Twitter, Whatsapp, Instagram, Snapchat, Facebook,…
- Actually is not that new…
  - emails: SMTP+MIME,
  - FTP,



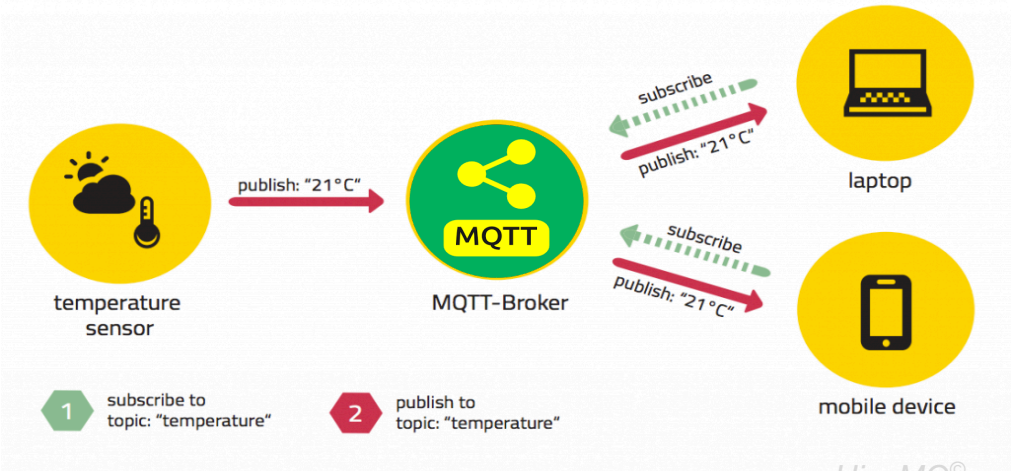
Request/response approach

- **REST**: Representational State Transfer
- Widely used; based on HTTP
- *Lighter version: **CoAP** (Constrained Application Protocol)*



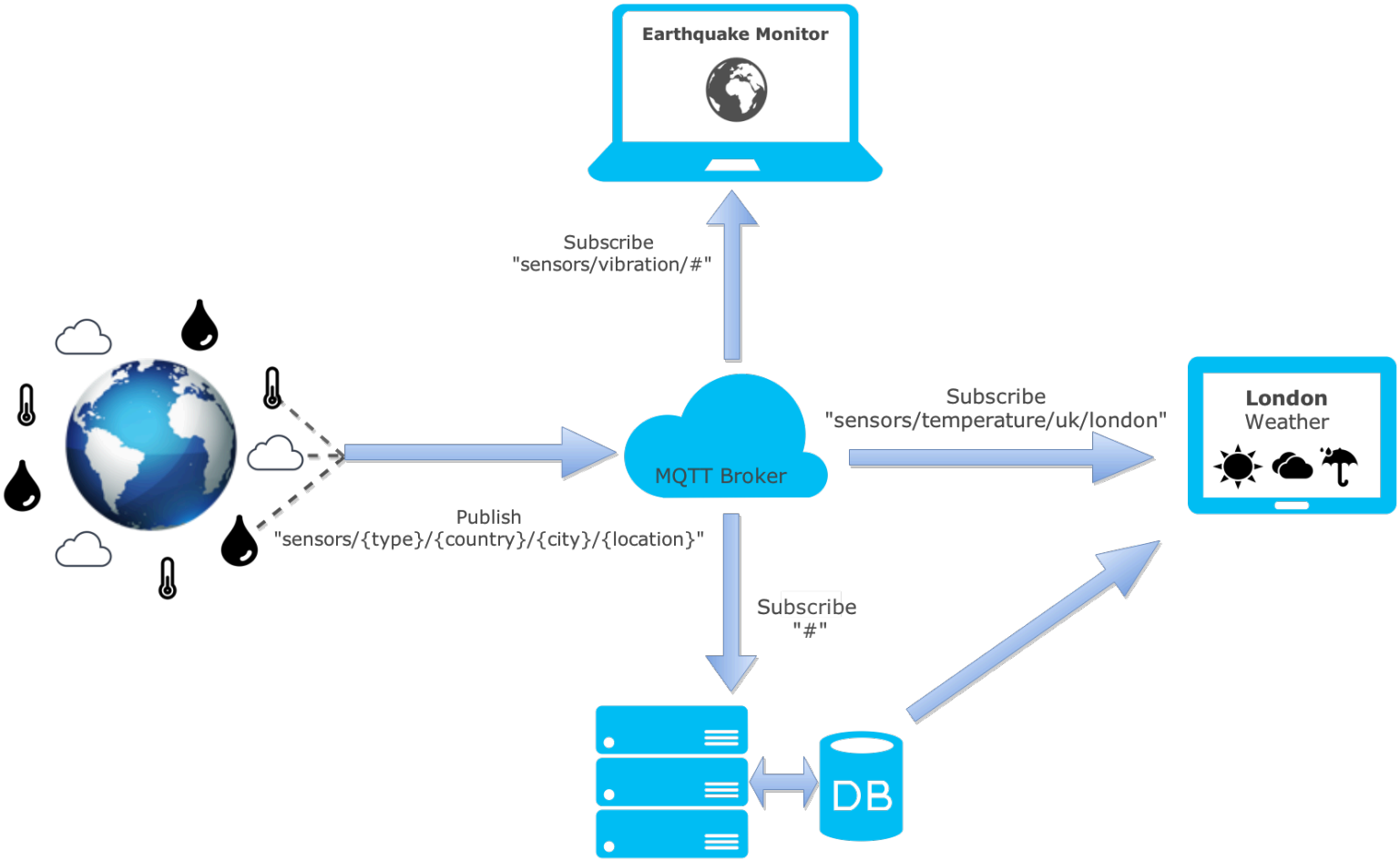
Pub/sub approach

- Pub/Sub separate a client, who is sending a message about a specific topic, called publisher, from another client (or more clients), who is receiving the message, called subscriber.
- There is a third component, called broker, which is known by both the publisher and subscriber, which filters all incoming messages and distributes them accordingly.



- *Various protocols:*
  MQTT, AMQP, XMPP (was Jabber)
- *Growing technique*
  E.g., https://cloud.google.com/iot/docs/how-tos/mqtt-bridge

An example

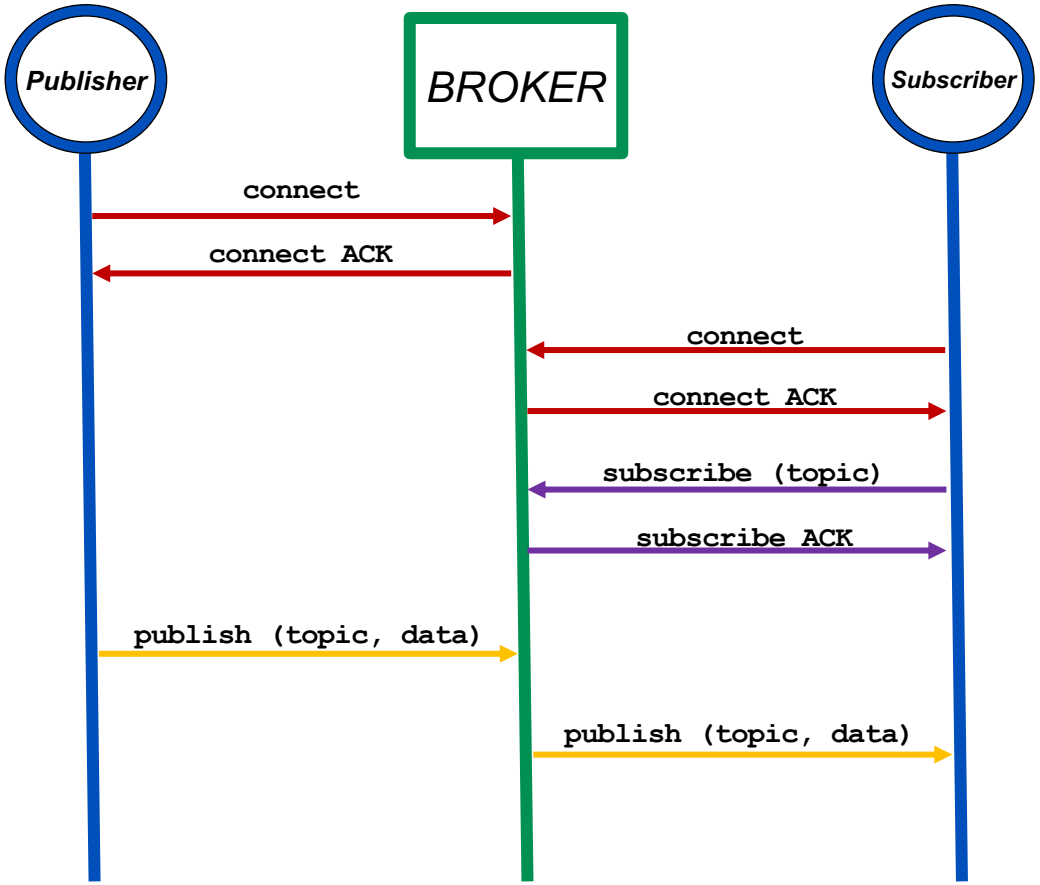


Source: https://zoetrope.io/tech-blog/brief-practical-introduction-mqtt-protocol-and-its-application-iot

Message Queuing Telemetry Transport

- A lightweight publish-subscribe protocol that can run on embedded devices and mobile platforms ➔ http://mqtt.org/
  - Low power usage.
  - Binary compressed headers
  - Maximum message size of 256MB
    - not really designed for sending large amounts of data
    - better at a high volume of low size messages.
- Documentation sources:
  - The MQTT community wiki:
    - https://github.com/mqtt/mqtt.github.io/wiki
  - A very good tutorial:
    - http://www.hivemq.com/mqtt-essentials/

Publish/subscribe interactions sequence



Topics

- MQTT Topics are structured in a hierarchy similar to folders and files in a file system using the forward slash (/) as a delimiter.
- Allow to create a user friendly and self descriptive **naming structures**

- Topic names are:
  - Case sensitive
  - use UTF-8 strings.
  - Must consist of at least one character to be valid.
- Except for the $SYS topic **there is no default or standard topic structure.**

Special $SYS/ topics

- $SYS/broker/clients/connected
- $SYS/broker/clients/disconnected
- $SYS/broker/clients/total
- $SYS/broker/messages/sent
- $SYS/broker/uptime

Topics wildcards

- Topic subscriptions can have wildcards. These enable nodes to subscribe to groups of topics that don't exist yet, allowing greater flexibility in the network's messaging structure.
  - '+' matches anything at a given tree level
  - '#' matches a whole sub-tree
- Examples:
  - Subscribing to topic `house/#` covers:
    - house/room1/main-light
    - house/room1/alarm
    - house/garage/main-light
    - house/main-door
  - Subscribing to topic `house/+/main-light` covers:
    - house/room1/main-light
    - house/room2/main-light
    - house/garage/main-light
  - but doesn't cover
    - house/room1/side-light
    - house/room2/side-light

Available MQTT brokers

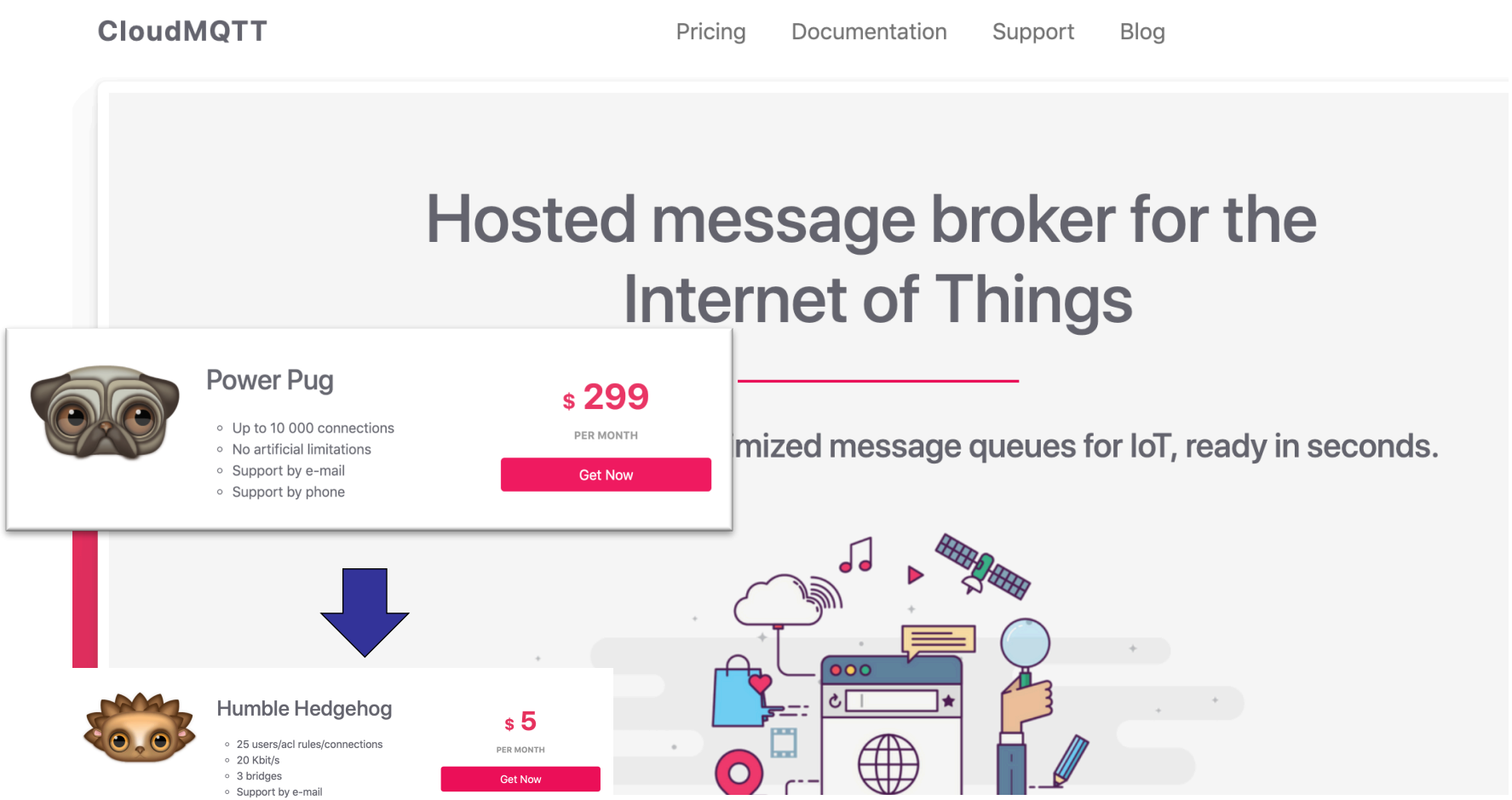
- The most widely used are:
  - http://mosquitto.org/
    - man page: https://mosquitto.org/man/mosquitto-8.html
  - http://www.hivemq.com/
    - The standard trial version only supports 25 connections.

- And also:
  - https://www.rabbitmq.com/mqtt.html
  - http://activemq.apache.org/mqtt.html

- A quite complete list can be found here:
  - https://github.com/mqtt/mqtt.github.io/wiki/servers

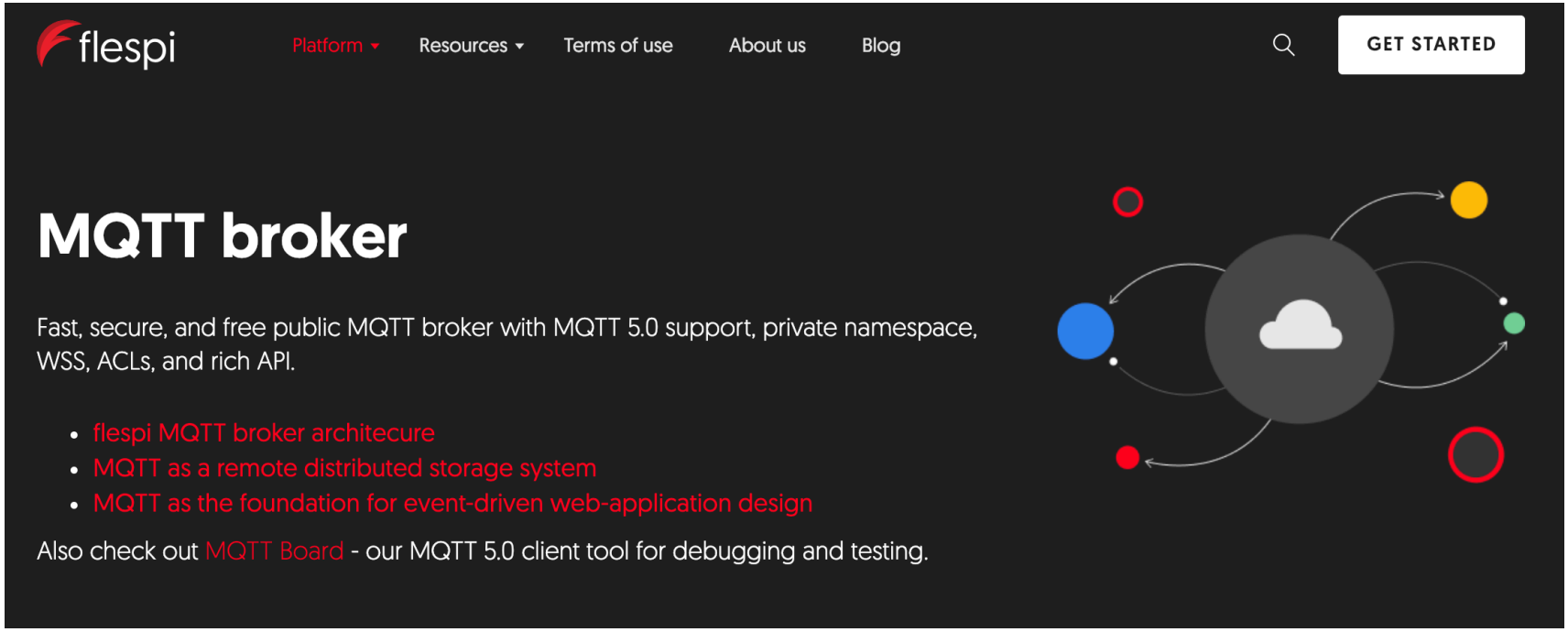
Cloud based MQTT brokers: CloudMQTT

https://www.cloudmqtt.com/ ➔ based on Mosquitto



Cloud based brokers: flespi

https://flespi.com/mqtt-broker



Open brokers ("Sandboxes")

- TCP based:
  - https://iot.eclipse.org/getting-started/#sandboxes
    - Hostname: **iot.eclipse.org**
  - http://test.mosquitto.org/
    - Hostname: **test.mosquitto.org**
  - https://www.hivemq.com/mqtt-demo/
    - Hostname: **broker.hivemq.com**
    - http://www.mqtt-dashboard.com/
  - Ports:
    - standard: 1883
    - encrypted: 8883 *(TLS v1.2, v1.1 or v1.0 with x509 certificates)*

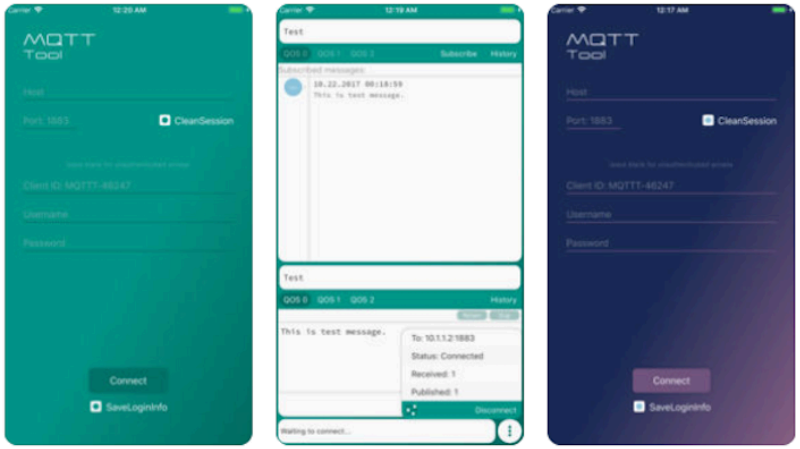
- Websockets based:
  - broker.mqttdashboard.com port: 8000
  - test.mosquitto.org port: 8080
  - broker.hivemq.com port: 8000

- https://github.com/mqtt/mqtt.github.io/wiki/public_brokers

MQTT clients: iOS

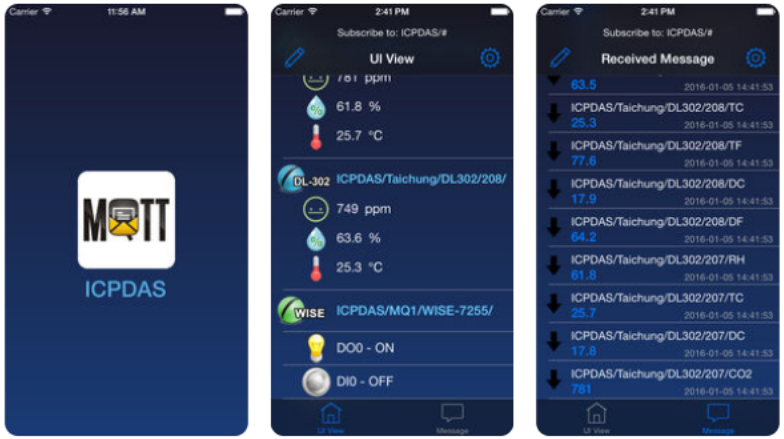
Mqttt

Utilidades

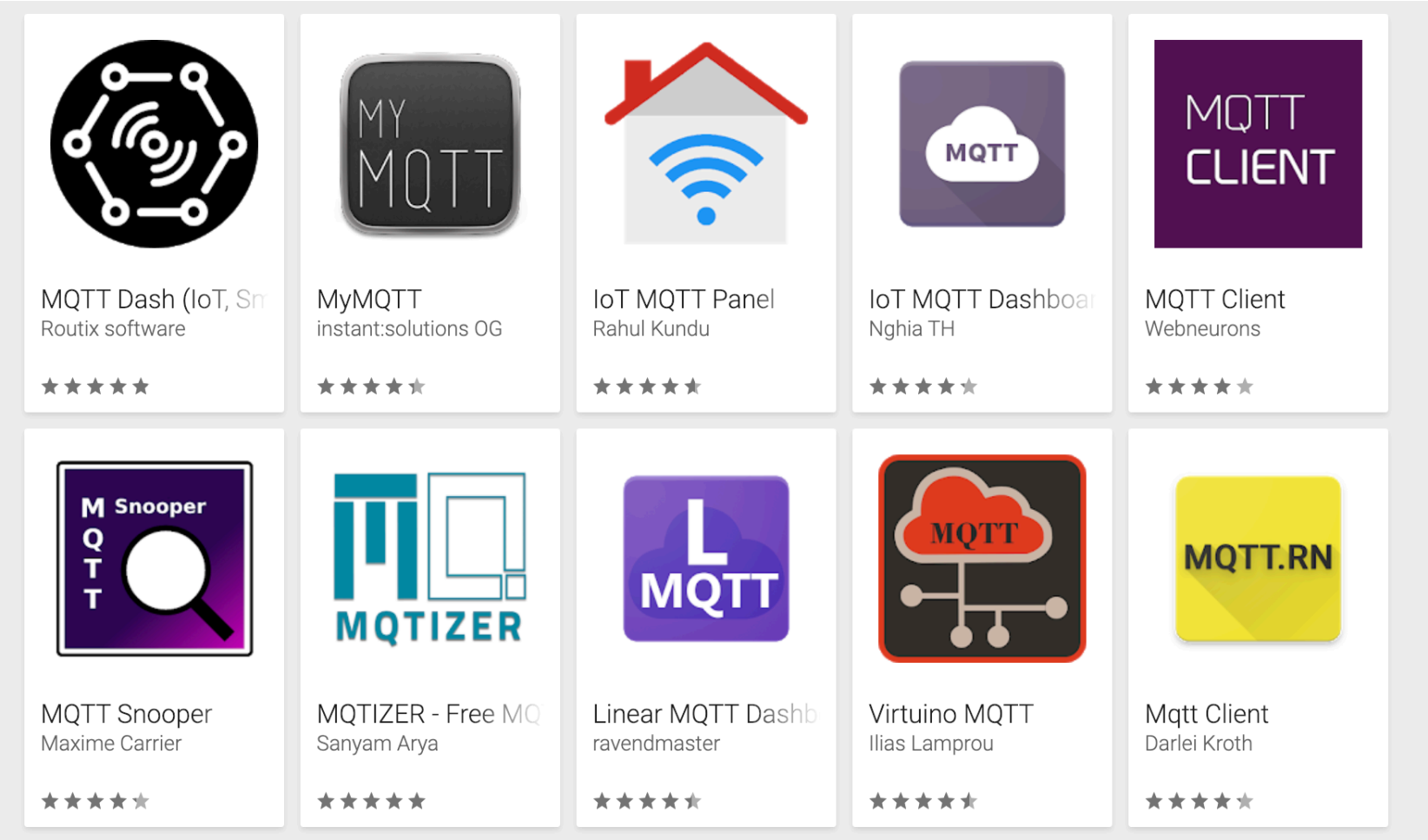


ICPDAS MQ...

Utilidades

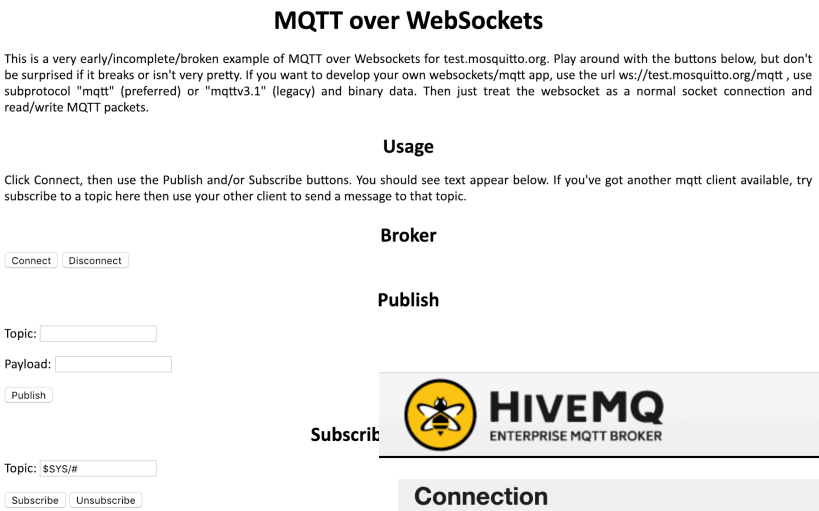


MQTT clients: Android

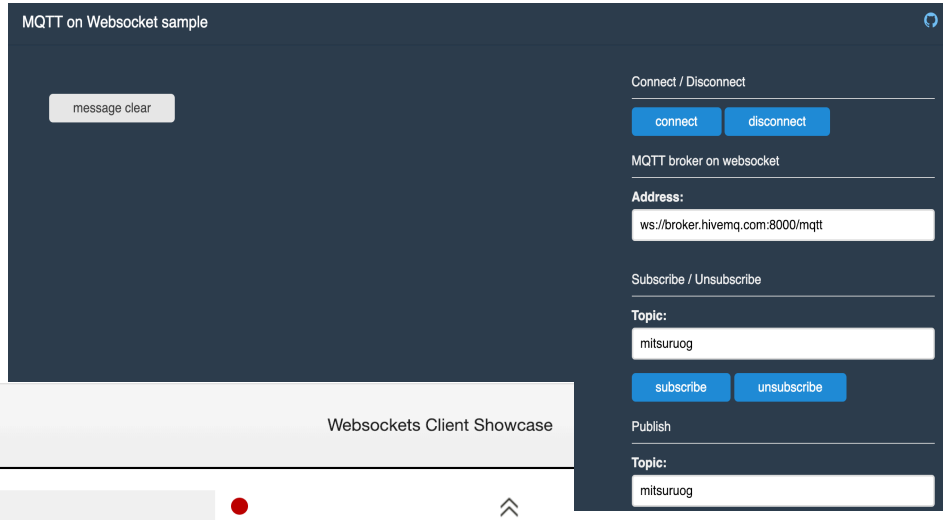


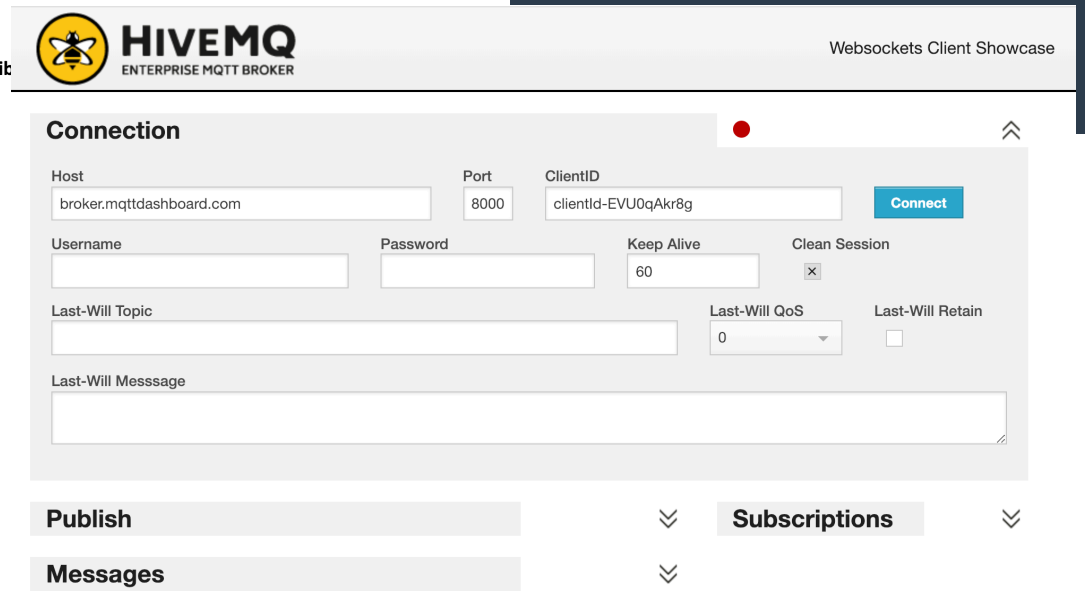
MQTT websocket clients

http://test.mosquitto.org/ws.html



http://mitsuruog.github.io/what-mqtt/





http://www.hivemq.com/demos/websocket-client/

http://mqtt-explorer.com