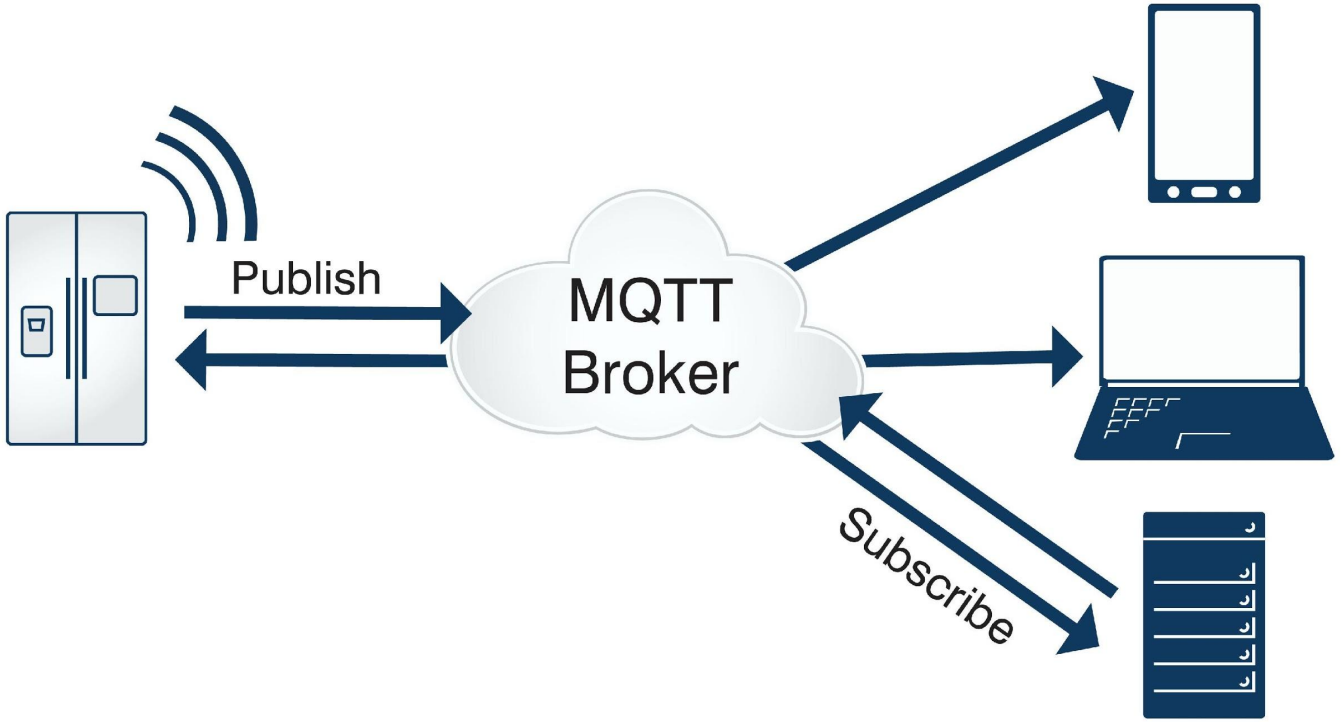


TALLER DE IOT – MQTT

ÍNDICE DE TEMAS

1. Definición
2. Esquema general
3. Broker y Client
4. Publish y Subscribe
5. Topics
6. Quality of Service (QoS)
7. Ejemplo integrador
8. Hive MQTT

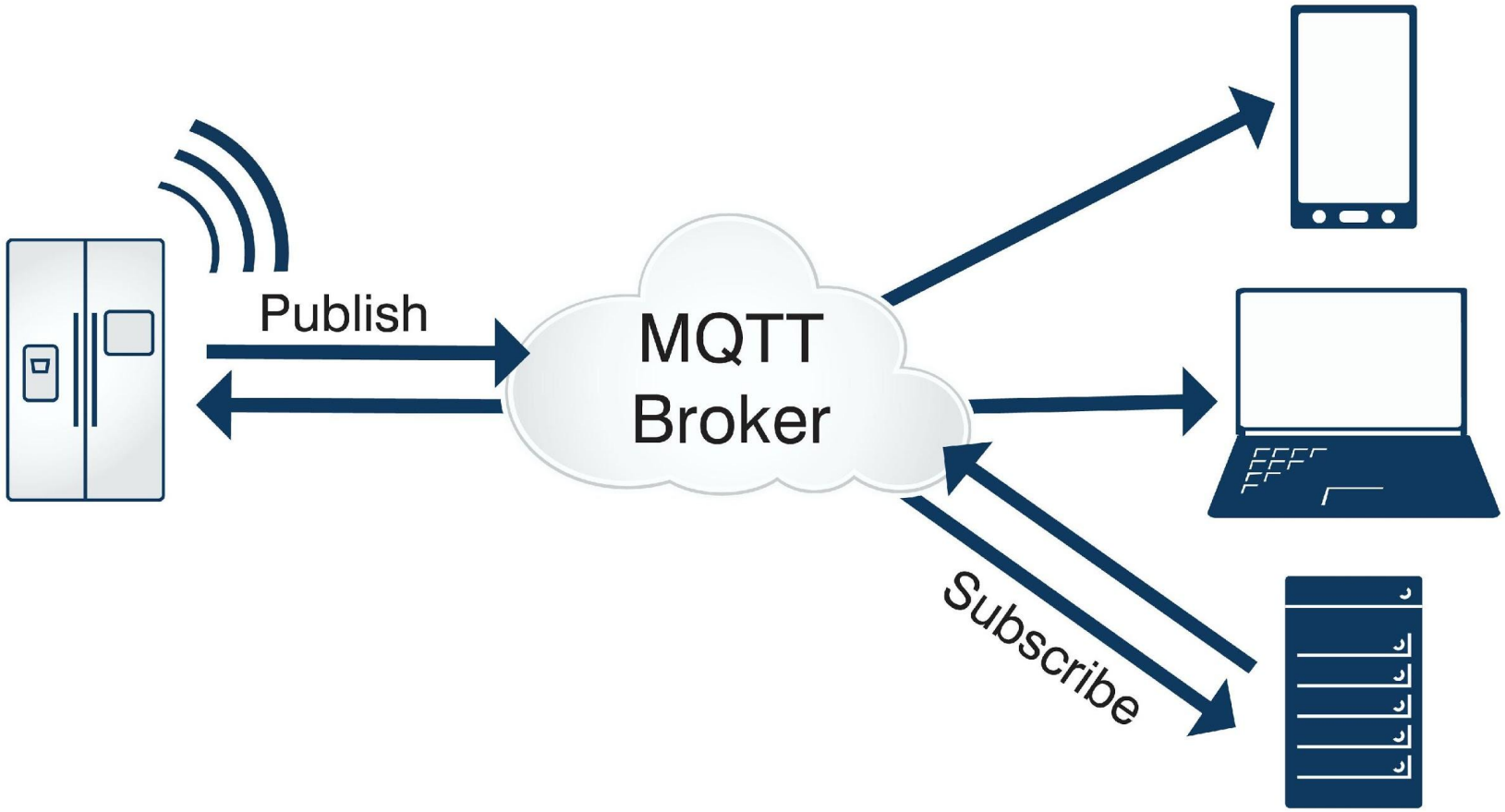
MQTT - DEFINICION

- *"MQTT is a Client Server publish/subscribe messaging transport protocol. It is light weight, open, simple, and designed so as to be easy to implement. These characteristics make it ideal for use in many situations, including constrained environments such as for communication in Machine to Machine (M2M) and Internet of Things (IoT) contexts where a small code footprint is required and/or network bandwidth is at a premium."*

  Citation from the official MQTT 3.1.1 specification

- En criollo:
  - Protocolo de comunicación basado en el concepto de publish/subscribe.
  - Se tiene un **broker**, al cual se conectan los clientes para intercambiar mensajes.
  - Orientado a comunicación M2M y para IoT, es decir, mensajes cortos y fugaces.

MQTT – ESQUEMA GENERAL



MQTT – BROKER Y CLIENT

Broker

- Se lo podría considerar al bróker como un servidor.
- Se encarga de **recibir** los mensajes, **filtrarlos**, determinar quién está suscrito a cada tópico, y **enrutar** el mensaje a los suscriptores.
- Puede contar o no con **autenticación** de acceso de los dispositivos conectados.
- El cliente siempre se conecta al bróker, nunca a otro cliente.

Client

- Todo dispositivo que se conecte con el bróker es un cliente, sin importar si publica un mensaje o si lo lee.
- Dicho dispositivo debe estar corriendo una API compatible con el protocolo MQTT.
- Disponibilidad de APIs:
  - Android, Arduino, C, C++, C#, Go, iOS, Java, JavaScript, y .NET, entre otras.
  - Más info en:

    https://github.com/mqtt/mqtt.github.io/wiki/libraries

MQTT – PUBLISH Y SUBSCRIBE

Publish

- **Publish = Publicar**
- Se utiliza para enviar un mensaje al bróker.
- Un publish se compone de:
  - **Topic**: campo donde se va a alojar el mensaje
  - **Payload**: mensaje a enviar
  - **QoS**: Quality of Service. 0 para nosotros
  - Entre otros…
- **¡El mensaje es siempre un string!**

Subscribe

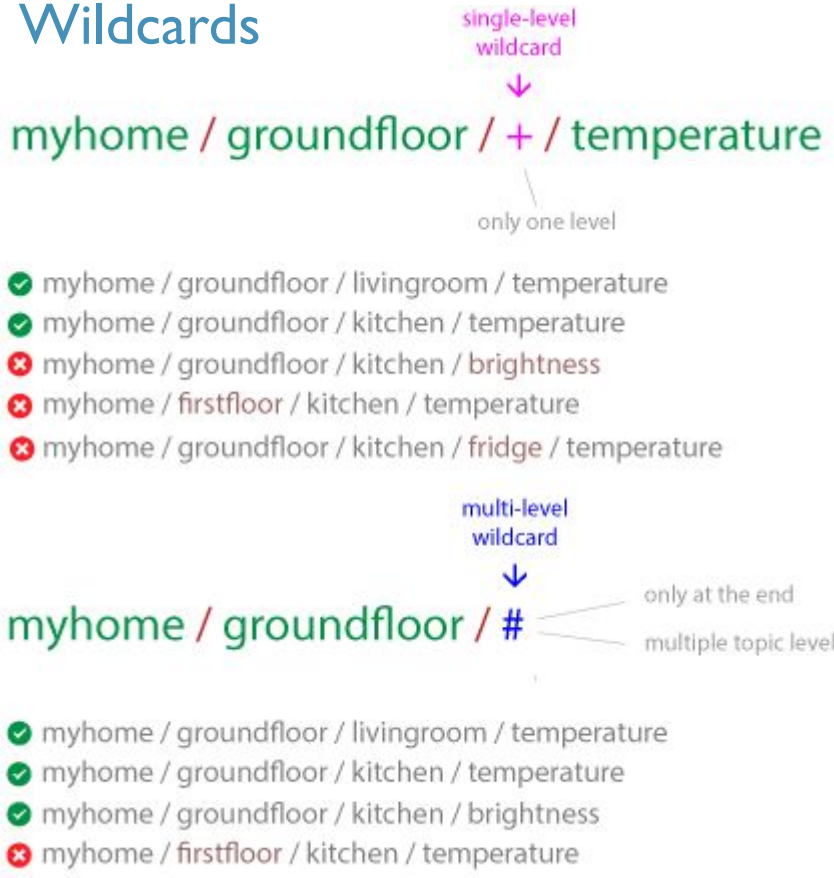
- **Subscribe = Suscribir**
- Se utiliza para recibir un mensaje del bróker.
- El cliente se suscribe a un tópico. Cada vez que llega un mensaje a dicho tópico, el bróker se lo envía a todos los clientes suscriptos.
- **¡El mensaje es siempre un string!**

MQTT – TOPIC

Topic

- **Topic = Tópico**
- Es un filtro de mensajes dentro del bróker.
- Su nombre debe estar compuesto por 1 o más caracteres, y es *case-sensitive*.
- Puede ser una sola palabra, o puede tener niveles:

Wildcards



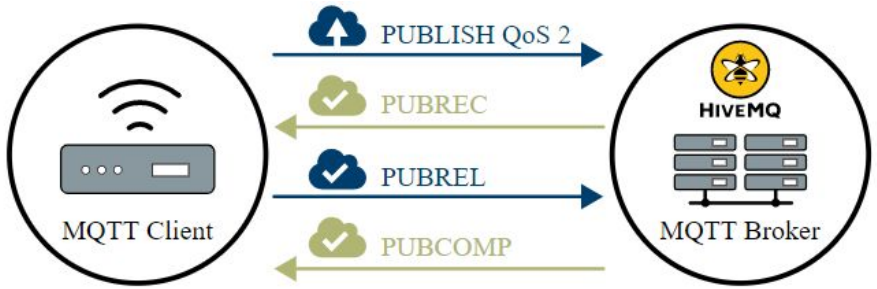
MQTT – QOS

QoS = Quality of Service

- Es la forma de gestionar la robustez del envío de mensajes al cliente ante fallos.
- **QoS 0 (at most one):** El mensaje se envía una única vez. En caso de fallo por lo que puede que alguno no se entregue.
- **QoS 1 (at least one):** El mensaje se envía hasta que se garantiza la entrega. En caso de fallo, el suscriptor puede recibir algún mensaje duplicados.
- **QoS 2 (exactly one):** Se garantiza que cada mensaje se entrega al suscriptor, y únicamente una vez.

Quality of Service level 0: delivery at most once

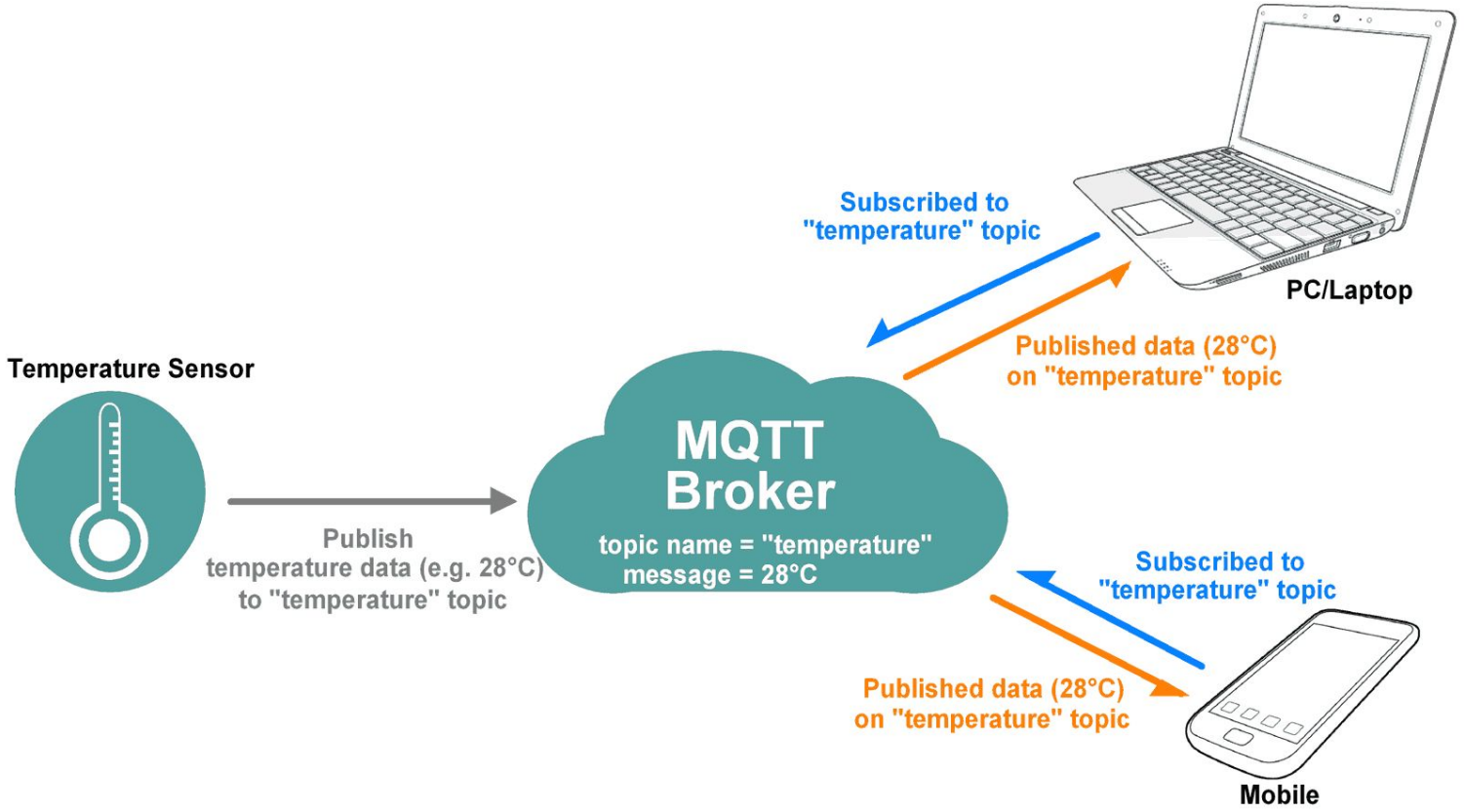
Quality of Service level 1: delivery at least once



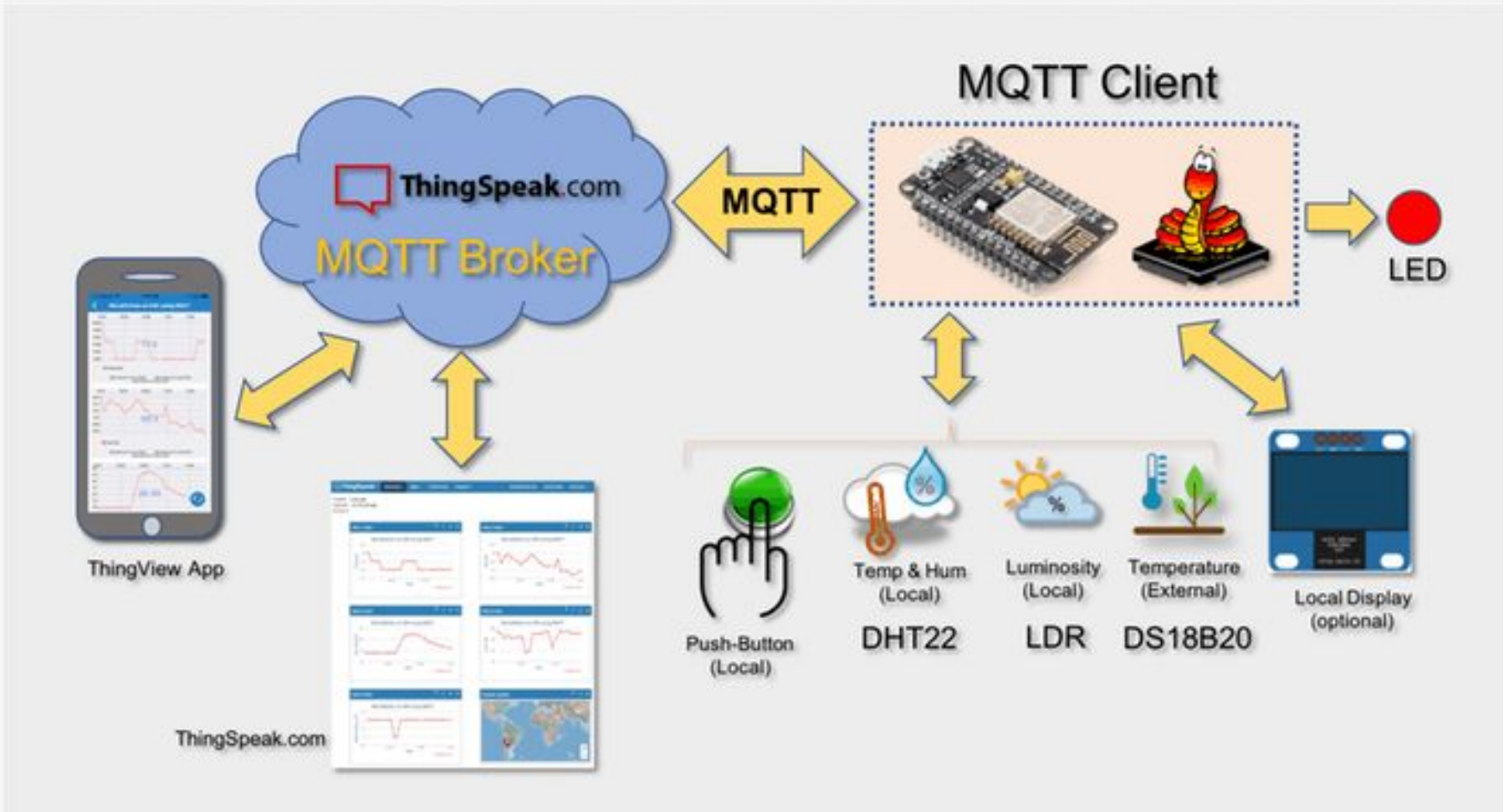
Quality of Service level 2: delivery exactly once

MQTT – EJEMPLO

- Un uC publica en el topic "temperature" un mensaje con la temperatura actual.
- Tanto la PC como el móvil están suscriptos al topic "temperature".
- En cuanto el uC publica el mensaje, a la PC y al móvil les llega el aviso, para que lean el dato del Broker.
- **Nota:** no alcanza con que sólo sea el mismo topic. El bróker debe ser el mismo para todos los dispositivos!

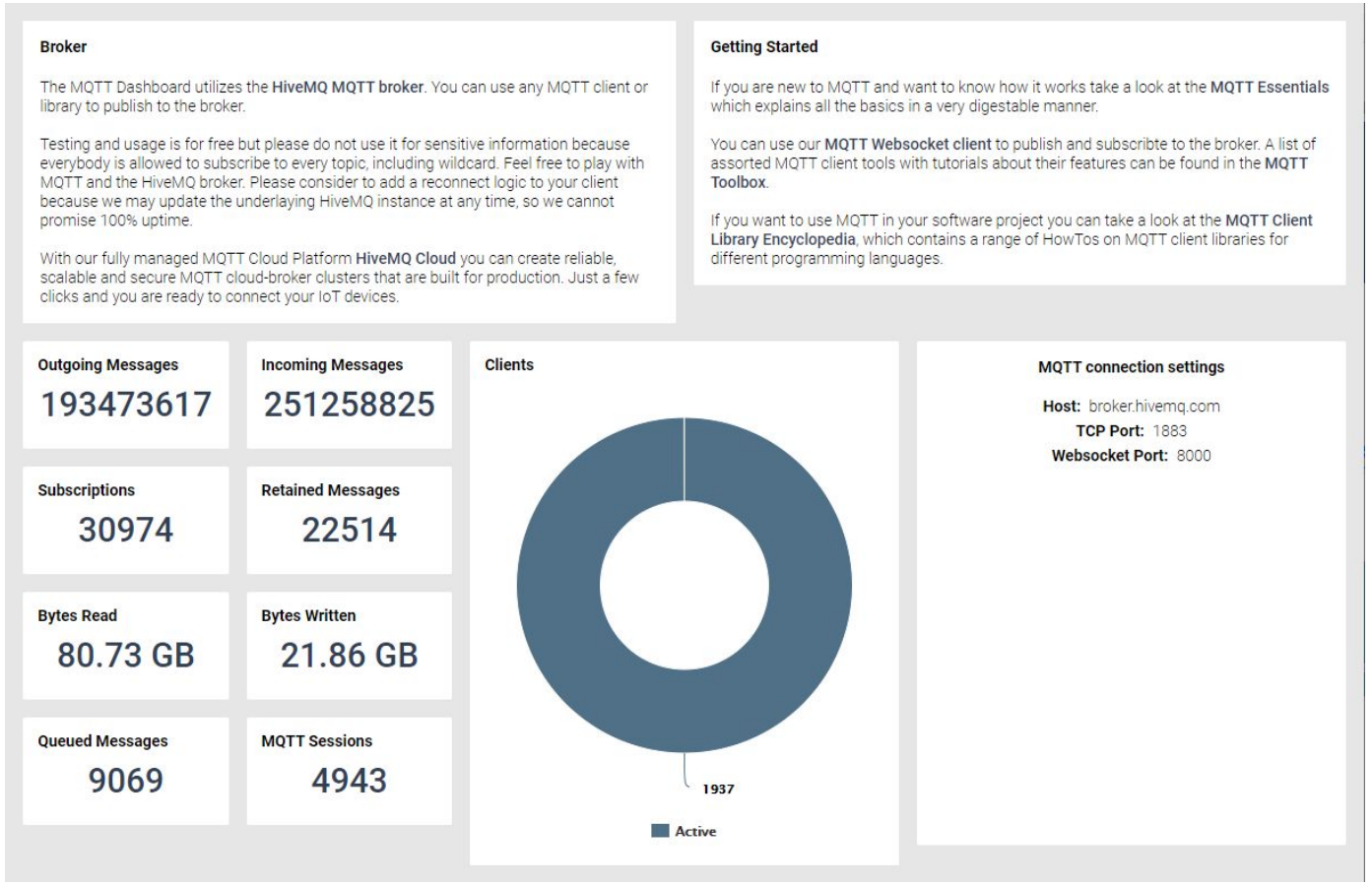


MQTT – EJEMPLO CON THINGSPEAK



HIVE MQTT BROKER

- Provee un Broker público gratuito.
- No garantizan seguridad en los datos (cualquiera se puede suscribir a cualquier tópico).
- Tampoco garantizan conexión constante (sugieren agregar lógica de reconexión al bróker).
- Brindan un WebSocket client para jugar desde la PC, en este link.



RESUMEN *FOR DUMMIES*

- **Broker** = Donde conecto el cable que viene de la calle.
- **Publish** = Acción de enviar un mensaje.
- **Subscribe** = Acción de quedarme escuchando mensajes de esa dirección.
- **Topic** = Dirección a la cual envío / de la cual espero mensajes.
- **QoS** = Qué tanto me importa si el mensaje llega bien o mal.
- **Cómo hacer esto con el ESP** = Lo vamos a ver la clase que viene.

¿PREGUNTAS, CONSULTAS HASTA ACÁ?

ENLACES ÚTILES

- List of MQTT Public Brokers: https://iotbyhvm.ooo/mqtt-public-brokers/
- What is MQTT and how it works: https://randomnerdtutorials.com/what-is-mqtt-and-how-it-works/
- The ultimate kickstart for MQTT beginners: https://www.hivemq.com/mqtt-essentials/
- Beginners guide to the MQTT Protocol: http://www.steves-internet-guide.com/mqtt/
- MQTT in depth: https://iotbyhvm.ooo/mqtt/
- HiveMQ API tutorials: https://www.hivemq.com/mqtt-client-library-encyclopedia/
- HiveMQ online dashboard: http://www.hivemq.com/demos/websocket-client/
- Qué es MQTT y la importancia en IoT: https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/

FIN