## L'intérêt du MQTT :

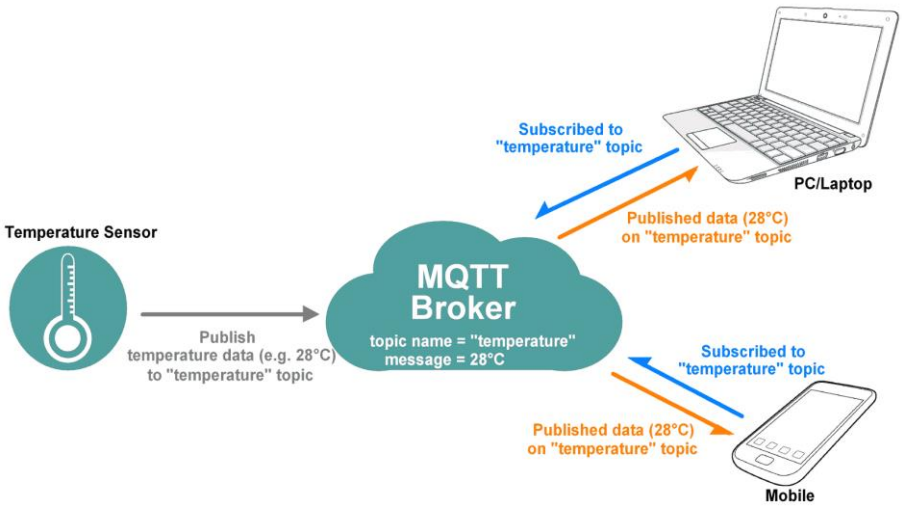
MQTT (Message Queuing Telemetry Transport) est un protocole ouvert, simple, léger et facile à mettre en œuvre. Ce protocole est idéal pour répondre aux besoins suivants :

- Utilisation d'une très faible bande passante.
- Utilisation sur les réseaux sans fil.
- Consommation en énergie réduite.
- Rapidité, avec un temps de réponse supérieur aux autres standards du web actuel.
- Fiabilité.
- Usage de faibles ressources processeurs et de mémoire.

## L'origine du MQTT :

MQTT est un standard ISO (ISO/IEC PRF 20922). Il a été développé par Andy Stanford-Clark (IBM) et Arlen Nipper (Eurotech) en 1999 pour le contrôle d'un pipeline situé dans le désert. L'objectif était d'avoir un protocole avec une faible bande passante (car les périphériques étaient connectés par liaison satellitaire) et utilisant peu d'énergie.

## Le principe de fonctionnement du MQTT :




Le protocole MQTT utilise une architecture « publish/subscribe » en contraste avec le protocole HTTP et son architecture « request/response ». Le point central de la communication est le broker MQTT en charge de relayer les messages des émetteurs vers les clients. Chaque client s'abonne une « boîte aux lettres » disponible sur un broker : on parle de « topic ». Le broker va alors réémettre les messages reçus des producteurs de données vers les clients. Les clients et les producteurs n'ont ainsi pas à se connaître. Ils ne communiquent qu'au travers des topics.

Chaque client MQTT a une connexion ouverte en permanence avec le broker. Si la connexion s'arrête, le broker bufférise les messages et les émet dès que la reconnexion avec le client est effectuée.

Un « topic MQTT » est une chaîne de caractères. Celle-ci peut posséder une hiérarchie de niveaux séparés par le caractère « / ». Exemple : une information de température du salon peut être envoyée sur le topic « maison/salon/temperature » et la température de la cuisine sur « maison/cuisine/temperature ».

Il est aussi possible d'utiliser la fonctionnalité « wildcards » (jokers MQTT) lors de l'abonnement à un topic en ajoutant les signes suivants :

+ : joker de niveau 1
# : joker multi-niveau

Exemple : si un souscripteur s'abonne au topic « maison/salon/# », il recevra toutes les données du salon. De même, s'il s'abonne au topic « maison/# », il collectera toutes les données des sondes de la maison. Les messages envoyés peuvent être de toutes sortes mais ne peuvent excéder une taille de 256 Mo.

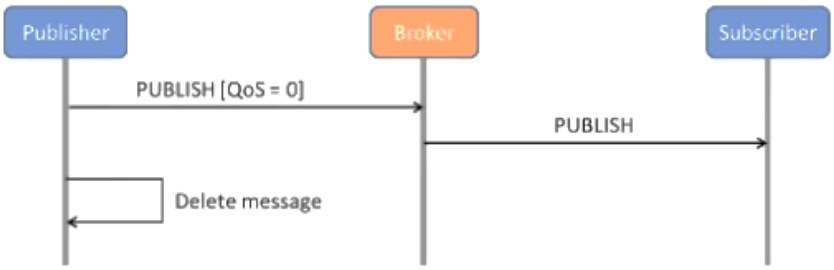
## La sécurité des données avec MQTT :

Les données IoT échangées peuvent s'avérer très critiques, c'est pourquoi il est aussi possible de sécuriser les échanges à plusieurs niveaux :
- Transport en SSL/TLS,
- Authentification par certificats SSL/TLS,
- Authentification par login/mot de passe.

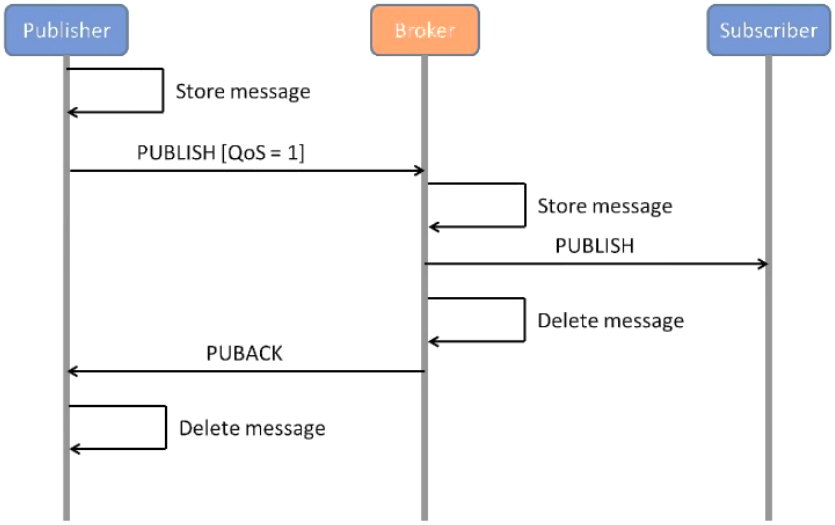
## La qualité de service (QoS) avec MQTT

MQTT intègre en natif la notion de QoS. En effet le publisher a la possibilité de définir la qualité de service. Trois niveaux sont possibles :

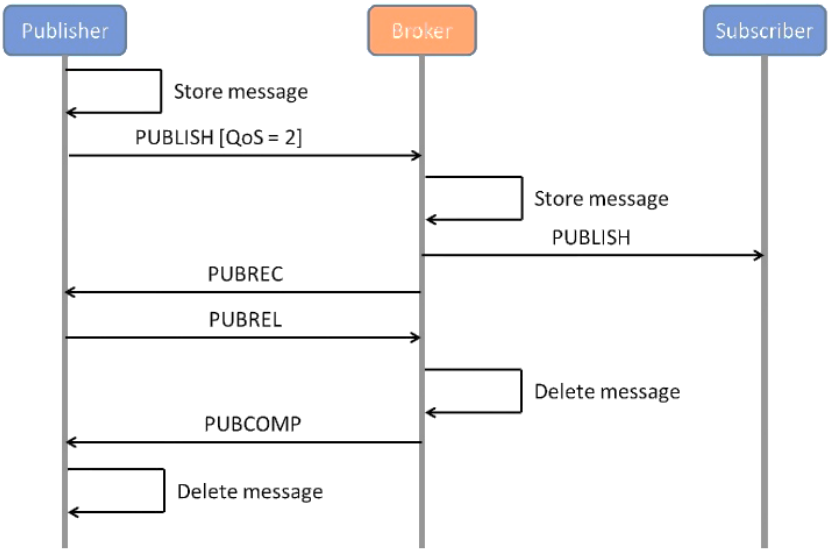
- Un message de **QoS niveau 0** « At most once » sera délivré au plus une fois. Ce qui signifie que le message est envoyé sans garantie de réception (le broker n'informe pas l'expéditeur qu'il a reçu le message). Le message envoyé n'est pas stocké par le Broker. Il n'y a pas d'accusé de réception. Le message sera perdu en cas d'arrêt du serveur ou du client. C'est le mode par défaut.




- Un message de **QoS niveau 1** « At least once » sera livré au moins une fois. Le Publisher le transmettra plusieurs fois s'il le faut jusqu'à ce que le Broker lui confirme qu'il a été transmis sur le réseau.




- Un message de **QoS niveau 2** « exactly once » sera obligatoirement sauvegardé par l'émetteur et le transmettra toujours tant que le récepteur ne confirme pas son envoi sur le réseau. La principale différence étant que l'émetteur utilise une phase de reconnaissance plus sophistiquée avec le broker pour éviter une duplication des messages (plus lent mais plus sûr).




## Mise en œuvre du MQTT

MQTT est multi-langage. On peut l'utiliser par exemple :

- en python avec une carte raspberry (https://pypi.python.org/pypi/paho-mqtt),
- en C/C++ depuis une carte Arduino ou le module Adafruit HUZZAH ESP8266 (https://github.com/256dpi/arduino-mqtt).

## Choix du broker MQTT

Il existe plusieurs brokers open-source :

- ActiveMQ qui permet d'ajouter MQTT à un serveur Web Apache (Développé par la fondation Apache)
- JoramMQ pour l'intégration de MQTT en Java
- Mosquitto, le broker open-source le plus utilisé dans les projets DIY soutenu par la fondation eclipse.org