	<h1 style="margin: 0;">Protocole MQTT</h1>
<p><b>Sommaire :</b></p> <p>I - Présentation du protocole MQTT..... 1</p> <p>II - Principe de fonctionnement du protocole MQTT..... 1</p> <p>III - Mosquitto..... 2</p>	

## I - Présentation du protocole MQTT

Pour répondre à la problématique du nombre grandissant d’objets connectés, l’**IoT**, s’est doté d’un nouveau standard : le protocole **MQTT** (**M**essage **Q**ueuing **T**elemetry **T**ransport).

**MQTT** est ouvert, simple, léger et facile à mettre en œuvre. Il est idéal pour répondre aux besoins suivants :

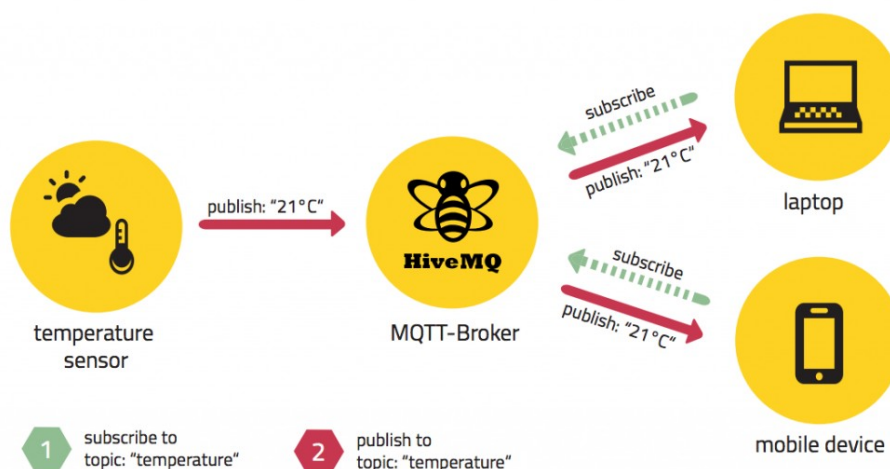
- faible consommateur en énergie ;
- très rapide, il permet un temps de réponse supérieur aux autres standards du web actuel ;
- permet une forte fiabilité si nécessaire ;
- nécessite peu de ressources processeurs et de mémoires ;
- légèreté : beaucoup moins verbeux que HTTP, avec un côté asynchrone natif.

**MQTT** est basé sur la suite de protocoles **TCP/IP**, utilisée par les protocoles internet dont **HTTP**. On peut donc le trouver sur n’importe quelle plateforme matérielle, que ce soit un microcontrôleur, un PC ou même un Cloud Microsoft Azure ou Amazon AWS.

## II - Principe de fonctionnement du protocole MQTT

**MQTT** est un **service de publication/abonnement TCP/IP** simple et extrêmement léger. Il fonctionne sur le principe **client/serveur**.

L’application réseau serveur, nommé **broker**, va collecter des informations que les **éditeurs (publishers)** vont lui transmettre. Certaines informations collectées par le broker seront renvoyées à certains **abonnés (subscribers)** en ayant préalablement fait la demande au **broker**. Un client peut être à la fois **éditeur** et **abonné**.



Les **messages** sont envoyés par les **éditeurs** sur un canal d'information appelé **topic**. Ces messages peuvent être lus par les **abonnés**. Les **topics** peuvent avoir une hiérarchie qui permet de sélectionner finement les informations que l'on désire.

Un **topic** est une **chaîne de caractère**. Il peut y avoir plusieurs niveaux de sujets séparés par un « / ».

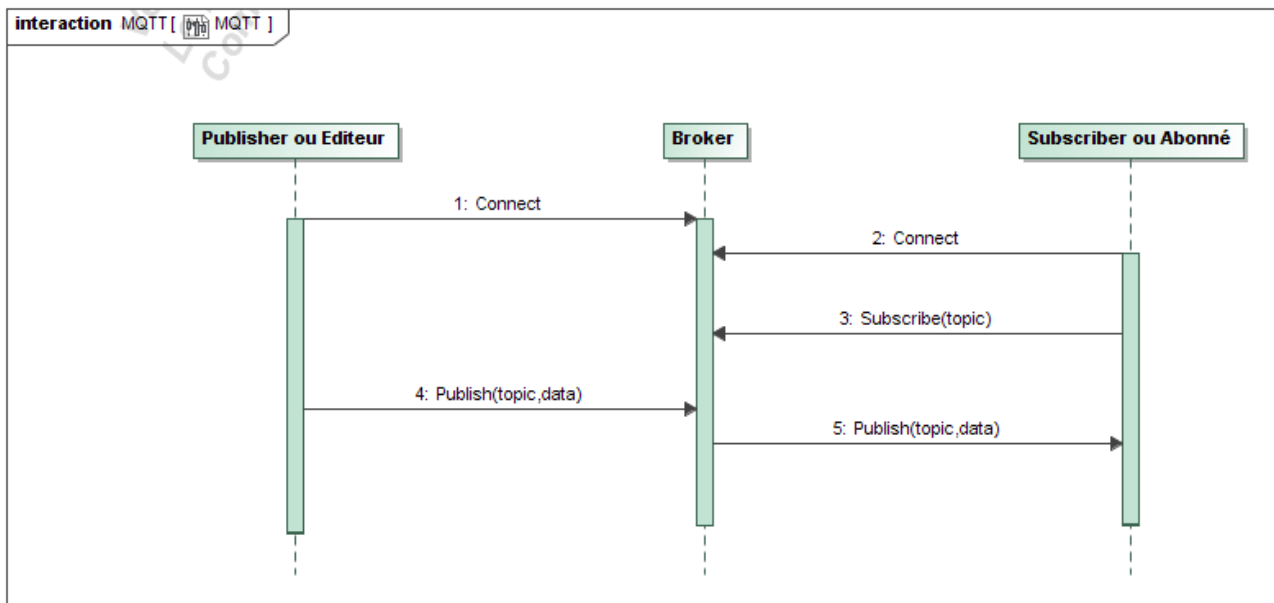
Exemples de topics :

**sensors/COMPUTER\_NAME/temperature/HARDDRIVE\_NAME**

**sensor/temperature/salon**

**sensor/temperature/#**

Le fonctionnement de **MQTT** peut être représenté par le diagramme de séquences suivant :



Les messages sont constitués de 2 éléments :

- un « **topic** » ou identifiant du message ;
- un « **body** » ou corps du message : les données du message qui peuvent se présenter sous différentes formes (binaire, chaîne de caractères ou chaîne avec encodage JSON par exemple). Les **messages** envoyés ne peuvent excéder une taille de **256 Mo**.

### III - Mosquitto

Le **serveur** ou **broker** pour les distributions **Linux** se nomme **mosquitto**. Il faut l'installer avec la commande **apt install mosquitto**.

Le **client** en ligne de commande pour les distributions **Linux** se nomme **mosquitto-clients**. Il faut l'installer avec la commande **apt install mosquitto-clients**. Afin de **souscrire** et de **publier** on utilisera respectivement les commandes **mosquitto\_sub** et **mosquitto\_pub**.

Par exemple, pour publier des données à destination du broker **192.168.X.4** et du topic **sensor/oc** :

**mosquitto\_pub -u iot -P iot -h 192.168.X.4 -t 'sensor/oc' -m '{"nom":"OC-INT-1","latitude":43.69,"longitude":-0.277,"temperature":22.1,"humidite":53}'**

Par exemple, pour lire les dernières données du topic **sensor/oc** du broker **192.168.X.4** :

**mosquitto\_sub -u iot -P iot -h 192.168.X.4 -t 'sensor/oc'**