

FabLab Saint-Gély-du-Fesc

Le protocole M.Q.T.T.

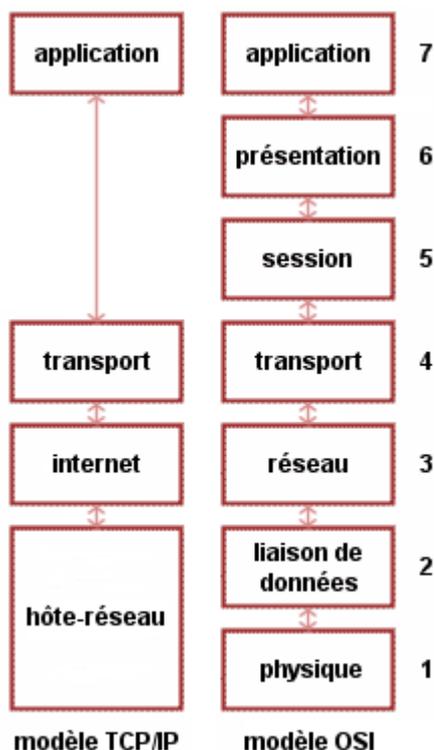
13 janvier 2022

# Protocole, vous avez dit protocole ?

On nomme protocole les conventions qui facilitent une communication ou des interactions sans faire directement partie du sujet de la communication elle-même. Ce terme est utilisé dans plusieurs domaines ...

Un protocole de communication est un ensemble de contraintes permettant d'établir une communication entre deux entités.

# Protocole, vous avez dit protocole ?



7 La couche hôte-réseau ... envoyer des paquets IP sur le réseau, une implémentation : Ethernet

6  
5 La couche internet : injection de paquets et acheminements (routage) des paquets indépendamment jusqu'à destination. Implémentation officielle : le protocole IP (*Internet Protocol*)

4  
3 La couche transport : permettre une conversation entre 2 entités paires – 2 implémentations : TCP et UDP

\* TCP (*Transmission Control Protocol*) : protocole fiable orienté connexion avec contrôle de flux.

\* UDP (*User Datagram Protocol*) : plus simple que TCP non fiable sans connexion temps de remise des paquets prédominant.

La couche application : TFTP (*Trivial File Transfer Protocol avec UDP*), FTP (*File Transfer Protocol avec TCP*), SMTP (*Simple Mail Transfer Protocol avec TCP*), HTTP (*Hypertext Transfer Protocol avec TCP*), HTTPS, ...

Référence : <https://www.frameip.com/tcpip/>.

# Mais revenons à MQTT ...

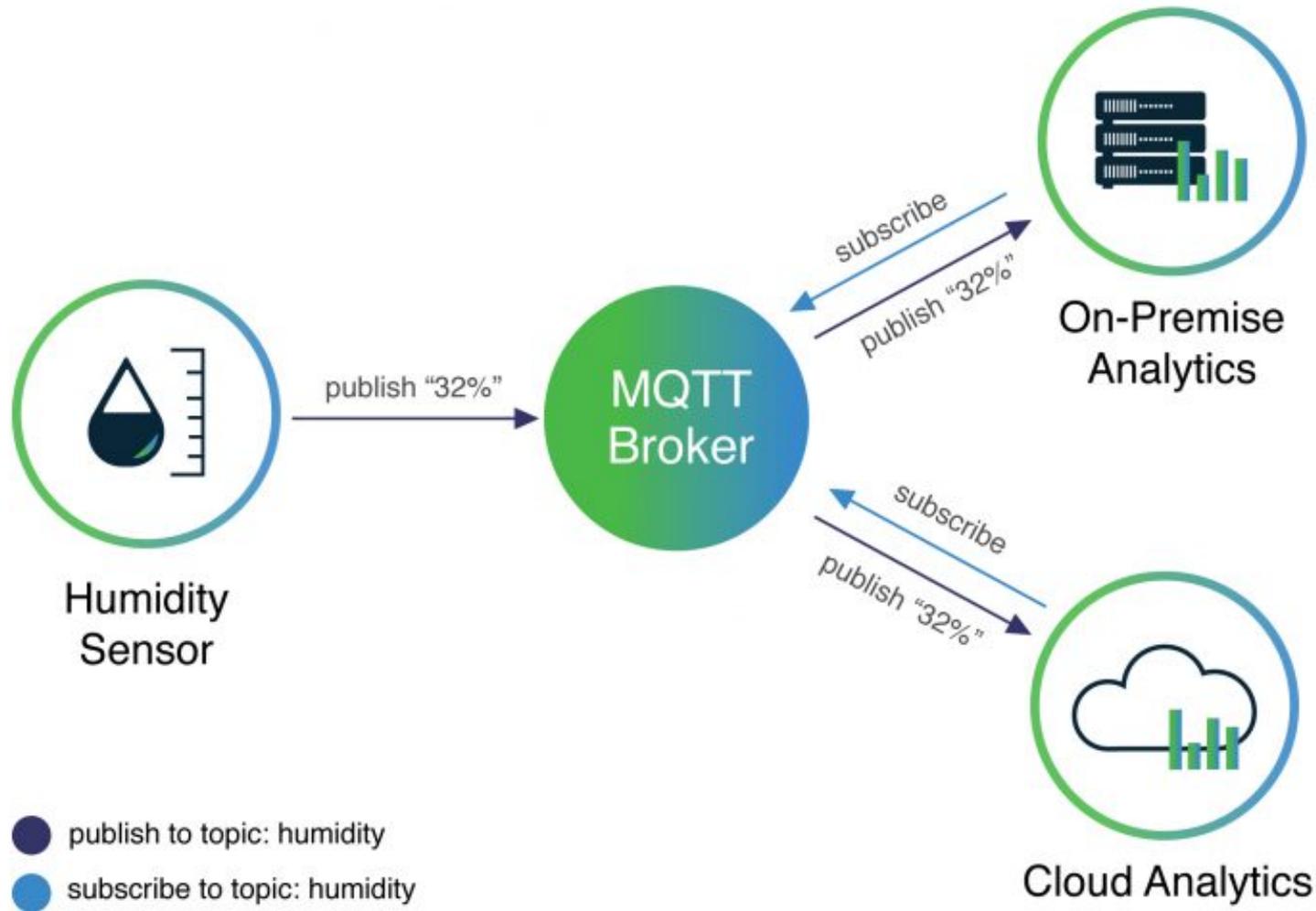
Créé en 1999 par Andy Stanford-Clark, ingénieur chez IBM, et Arlen Nipper, chez EuroTech

Devenu une norme ISO (*International Organization for Standardization*) en 2016, MQTT 3.1.1 est un standard OASIS (*Organization for the Advancement of Structured Information Standards*), la version 5 de la spécification est maintenant publiée depuis le 7 mars 2019.

MQTT (*Message Queuing Telemetry Transport*) est un protocole open source de messagerie publish-subscribe basé sur le protocole TCP/IP.

Il assure des communications non permanentes entre des appareils par le transport de leurs messages. C'est donc du M2M où les appareils peuvent être de technologies différentes.

# Une image vaut mieux qu'un long discours ...



# QoS ... Quésako (occitan)

3 niveaux de qualité de service :

Au plus une fois, au moins une fois et exactement une fois.

Les deux côtés d'un message envoyé sont à prendre en compte : la remise du message du client au broker et le message du broker au client abonné. Le client qui publie le message sur le broker définit le niveau de QoS du message lorsqu'il envoie le message au broker. Le courtier transmet ce message aux clients abonnés, tout en utilisant le niveau de QoS que chaque client abonné définit au cours du processus d'abonnement. Si le client abonné définit une qualité de service "inférieure" à celle du client, le broker transmet le message avec un QoS inférieur. La QoS donne ainsi au client le pouvoir de choisir un niveau de service correspondant à la fiabilité de son réseau ainsi qu'à sa logique d'application.

# QoS ... Quésako (occitan)

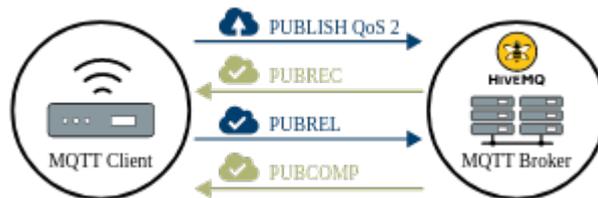
QoS 0 : Moindre effort – aucune garantie que le message est délivré.



QoS 1 : L'émetteur stocke le message tant que le récepteur ne l'a pas acquitté. Le message peut donc être émis plusieurs fois ... et donc reçu plusieurs fois.



QoS 2 : Une fois et une seule.



Référence : <https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/>

# Les logiciels brokers et clients

Brokers : ActiveMQ, JoramMQ, Mosquitto, RabbitMQ, ...

Mosquitto est disponible sur toutes les plateformes (MacOS, Windows XP-10, Linux).

Il est le broker le plus souvent utilisé pour les projets ESP8266, Arduino, Raspberry, ...

Les bibliothèques clientes : pour la plupart des langages (C, C++, Java, JavaScript, PHP, Python...) et sur la plupart des plates-formes (GNU/Linux, Windows, iOS, Android, Arduino...).

# Pourquoi utiliser MQTT ?

Sa légèreté donc utilisable sur microcontrôleurs

Sa souplesse et simplicité de mise en œuvre

Sa robustesse (QoS)

La possibilité de mettre une couche de sécurité pour l'authentification et la transmission des messages TLS

Dans l'IoT, avec les assistants vocaux, Facebook Messenger, le cloud des GAFAM, ...

# Un peu de pratique ... Linux Fedora Server

Installation du broker à partir des dépôts

```
$ sudo dnf install mosquitto
```

Création du fichier d'autorisation pour le broker

```
$ sudo mosquitto_passwd -c /etc/mosquitto/accesslist.txt user
```

Paramétrage du broker : adresse de bind – port d'écoute

Fichier : /etc/mosquitto/mosquitto.conf

S'assurer que le firewall de la machine autorise la configuration définie :

```
$ sudo firewall-cmd --zone=dmz --add-port=xxxx/tcp --permanent
```

Activation du broker au boot de la machine

```
$ sudo systemctl enable mosquitto
```

Les commandes basiques en console :

Publication :

```
mosquitto_pub -h addr_IP -u user -P password -t topic -m payload
```

Souscription :

```
mosquitto_sub -h addr_IP -u user -P password -t topic
```

Merci pour



votre attention