

Projet 2023

Plateforme de gestion des informations IoT

Version 5 mai 2023 (10:08)

Cahier des charges

On désire concevoir par nos propres moyens une plateforme de gestion des informations en IoT (*Internet of Things*, c'est-à-dire les objets connectés).

L'architecture générale de la solution qu'il vous est demandé de mettre en place est représentée sur la figure 1. Vous pouvez remarquer que les connexions IoT se feront au travers du protocole MQTT (*Message Queuing Telemetry Transport*). Soit directement depuis le device, soit par l'intermédiaire d'un programme (ex : Python) qui collecte les données et les retransmet à la plateforme via MQTT (*publish*). D'un autre côté vous avez un ou des programmes (ex : Python) qui récupèrent ces données depuis MQTT (*subscribe*) et les stockent (ex : base de données SQL). D'autres programmes pourraient également enregistrer directement des données dans la BdD. Finalement, à partir de cette BdD, différents programmes peuvent présenter les données à l'utilisateur : des scripts PHP qui génèrent des pages HTML, des programmes Python (ou autre) qui génèrent des documents PDF, des images PNG, etc.

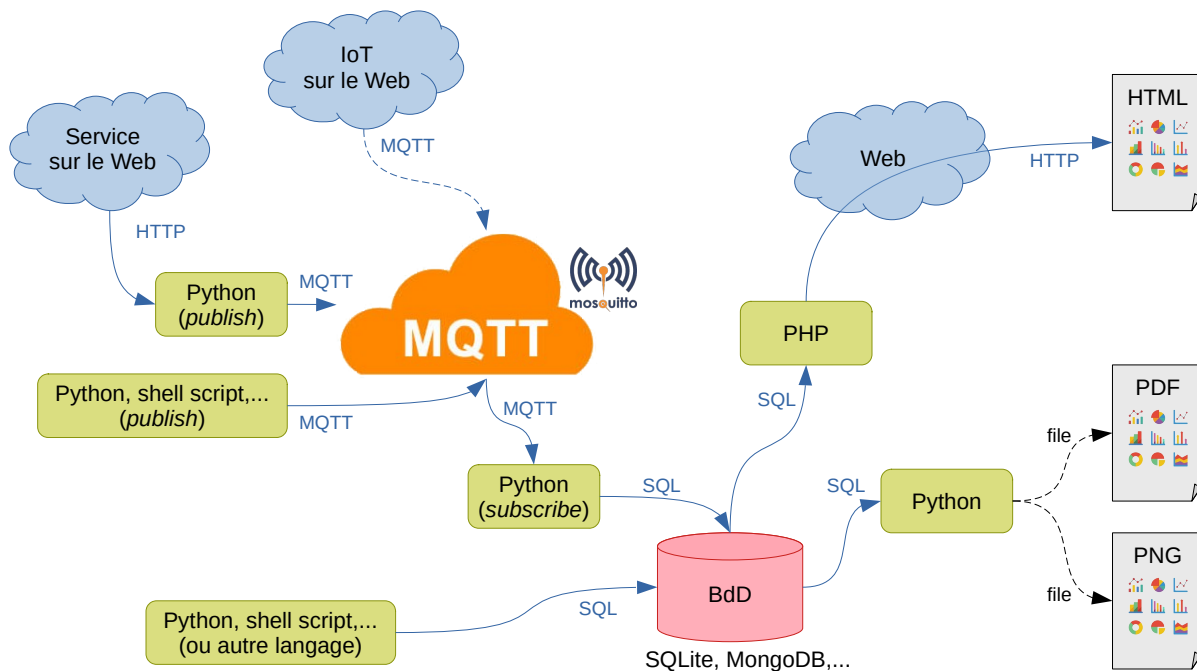


FIGURE 1 – Architecture générale

Travail à réaliser

- **Ce qu'il faut absolument faire :**
 - programme Python qui envoie des données via MQTT
 - programme Python qui reçoit des données via MQTT et les stocke dans une Bdd
 - site web dynamique (aka scripts PHP) qui présente les données via une interface web avec la génération de graphiques
- **Ce qui est fortement conseillé :**
 - dans les messages MQTT, la "charge utile" (le *payload*) est une chaîne de caractères → pour envoyer des données structurées on peut utiliser du JSON par exemple
 - écrire des programmes qui vont chercher des données sur le web (ex : Open Data) et les envoient sur notre plateforme en MQTT
- **Ce qui serait un plus :**
 - écrire un programme qui génère un document PDF à partir des données
 - écrire un programme qui génère une image PNG à partir des données; cette image pourrait d'ailleurs ensuite être incluse dans une page web (NB : moins dynamique que si c'est un script PHP qui génère cette image à la volée)

Pour vous aider

Histoire de vous aider un petit peu, j'ai écrit un programme Python qui récupère les informations météo sur OpenWeather (<https://openweathermap.org/>) pour l'emplacement "IUT MdM" et les republie sur des topics MQTT. J'ai mis ce programme dans la `crontab` d'un Linux afin qu'il soit automatiquement exécuté toutes les 2mn. Voici une trace d'un programme qui écouterait sur ce(s) topic(s) (sur le broker `test.mosquitto.org`).

```
Received '17.74' from '/rtmdm/mm/sae203/g553dtQJS7yk2G/weather/temp' topic
Received '17.4' from '/rtmdm/mm/sae203/g553dtQJS7yk2G/weather/feels_like' topic
Received '16.73' from '/rtmdm/mm/sae203/g553dtQJS7yk2G/weather/temp_min' topic
Received '19.42' from '/rtmdm/mm/sae203/g553dtQJS7yk2G/weather/temp_max' topic
Received '1020' from '/rtmdm/mm/sae203/g553dtQJS7yk2G/weather/pressure' topic
Received '70' from '/rtmdm/mm/sae203/g553dtQJS7yk2G/weather/humidity' topic
Received '{"temp": 17.74, "feels_like": 17.4, "temp_min": 16.73, "temp_max": 19.42,
"pressure": 1020, "humidity": 70}' from '/rtmdm/mm/sae203/g553dtQJS7yk2G/weather/all' topic
```

Évaluation

Pour ce projet vous travaillerez par binôme, ce qui évitera certains "problèmes" entre séances de TD et séances de TP. Votre évaluation pour ce projet en SAE203 se fera en deux parties : une évaluation finale du projet terminé (lors de la dernière séance) et un suivi individualisé de l'avancement de votre projet au fur et à mesure des séances.