



19/06/2023

Cahier des charges - système MQTT

Université d'été

Albanesi Nicolas

Kandiah Abivarman

Stirnemann Jonas

Dymarczyk Dawid

Contents

1. Descriptif détaillé du projet	3
2. Schéma illustrant le projet	3
3. Elements	3
3.1. Serveur	3
3.1.1. Broker MQTT	3
3.1.2. Publisher	4
3.2. Clusters	4
3.2.1. Topics	4
4. Exemples d'utilisation	4
4.1. Affichage de cours sur la salle	4
4.2. Dessin sur afficheur	5
4.3. Affichage menu cafeteriat	5
4.4. Games	5

1. Descriptif détaillé du projet

Projet : Système MQTT

Le système est composé d'un serveur contenant le broker MQTT (Mosquitto) et Node RED.

Chaque device (un esp32 et plusieurs hepialight2 par device) sera abonné à un topic MQTT, et va actualiser son affichage en fonction des données reçues par le publisher. Le but serait d'avoir par exemple un device par salle de classe, puis le publisher indique à chaque device les informations du cours qui ont lieu dans la salle de classe. On pourrait de cette manière savoir si la salle est libre ou non, avec le nom du prof et du cours affiché au dessus de la porte. Quelques devices pourraient également se situer aux alentours de la cafeteria afin d'afficher le menu disponible.

2. Schéma illustrant le projet

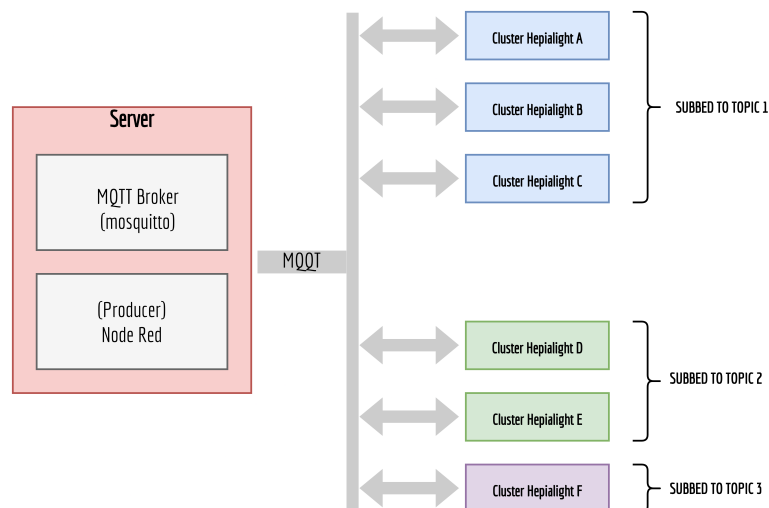


Figure 1: Architecture de l'entité Maître

3. Elements

3.1. Serveur

Nous avons besoin d'une unité pouvant accueillir un Broker MQTT et un gestionnaire de publications. Il nous faut donc un serveur.

3.1.1. Broker MQTT

Afin de gérer au mieux notre système, nous avons besoin d'une entité qui gère les différents Topics, souscriptions et Publishers. Il existe plusieurs broker MQTT, nous avons décidé de partir avec Mosquitto <https://mosquitto.org/> parcequ'il est gratuit, léger et open source.

3.1.2. Publisher

Afin de centraliser au mieux la gestion des publications, nous avons décidé d'utiliser Node RED <https://nodered.org/>. Il nous permet de générer des flots de données. et de publier sur les publics

3.2. Clusters

Un cluster est composé d'un ESP32 et plusieurs matrices Hepialight2 (modulaire).

Chaque cluster va pouvoir être actualisé par le serveur MQTT, en fonction du topic au quel le cluster est abonné.

3.2.1. Topics

Nous pensons avoir plusieurs topics pour différentes utilisations. Par exemple un topic pour les salles de classes avec en sous topic, le numéro de la salle. Un autre topic pour la cafétéria. On pourrait aussi dessiner sur un cluster a partir d'une interface WEB par exemple.

```
- /Classe
  - /Bat_I
    - /301
    - /302
    - /303
    ...
- /Cafeteria
  - /Menu1
  - /Menu2
  - /Menu3
```

4. Exemples d'utilisation

4.1. Affichage de cours sur la salle

Comme dit précédemment, une des utilité principale serait d'afficher des informations sur l'état d'une salle de classe. Au dessus de chaque porte de classe se trouverait une hepialight affichant le prof présent dans la salle ainsi que le nom du cours. Si la salle est vide, le device affiche que la salle est disponible.

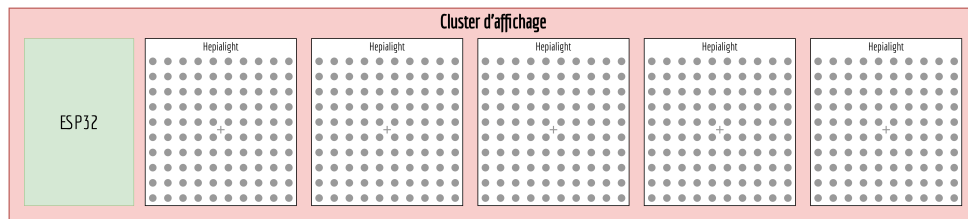


Figure 2: Architecture de l'entité Maître

4.2. Dessin sur afficheur

Nous pourrions dessiner sur des pixels d'une page web et les appuies seraient publiés sur un topic auquel un cluster de matrice serait souscrit.

4.3. Affichage menu cafeteriat

Il serait également possible de trouver des devices aux alentours de la cafeteria afin d'afficher le menu du jour pour les étudiants.

4.4. Games