



UNIVERSITÉ
TOULOUSE III
PAUL SABATIER



Université
de Toulouse



ID-PHYCS

Interactive Disseminated PHYsical Complex Sensors

Novembre - Décembre
2021

Clément Petit / Alexandre Baures
3A - SRI

Upssitech - Paul Sabatier

Sommaire

Introduction	3
Objectifs	3
Choix de conception	4
Disposition des capteurs dans la maison	4
Choix du hardware	5
Choix du protocole de communication	6
Architecture du systeme	7
Avantages, Inconvénients et limites de notre solution	8
Illustrations	9-10
Conclusion	11

Introduction

Un système réparti ou distribué est un système informatique dans lequel les ressources ne sont pas centralisées. Le but étant de permettre à des utilisateurs de manipuler leurs données sans contrainte sur les localisations respectives des éléments du système. Cela permet de créer des systèmes très versatiles qui s'adapte à chaque environnement et chaque application. Ce type de système nécessite tout de même la mise en place d'une architecture réseaux afin que chaque composant puisse communiquer entre eux et ainsi partager leurs données.

Objectifs

Nous allons concevoir un système permettant de collecter et de superviser un ensemble de capteurs (température et humidité) IP et non IP disséminé dans une maison. Les capteurs d'une pièce sont regroupés en agrégats. Un slot est composé de plusieurs pièces.

Lancement de l'application

Malheureusement la formule gratuite du broker MQTT sur Shiftr.io permet de maintenir un broker actif pour seulement 6h. Ainsi si vous voulez lancer l'application il faut aller renseigner dans le code les informations d'un nouveau broker lancé préalablement.

Une fois cela fait, il faut lancer:

Capteurs_virtuels.pde
TraitementHumi.pde
TraitementTemp.pde

De plus il faut aller renseigner les informations de WIFI dans le code de l'ESP32 pour que celui-ci aille se connecter au réseau en plus d'enregistrer un nouveau broker MQTT et enfin brancher le capteur de température DHT11 sur le pin D4 l'alimenter via son port USB

Le lancement de l'application étant un peu compliqué, je vous conseille d'aller voir la vidéo de démonstration disponible ici:

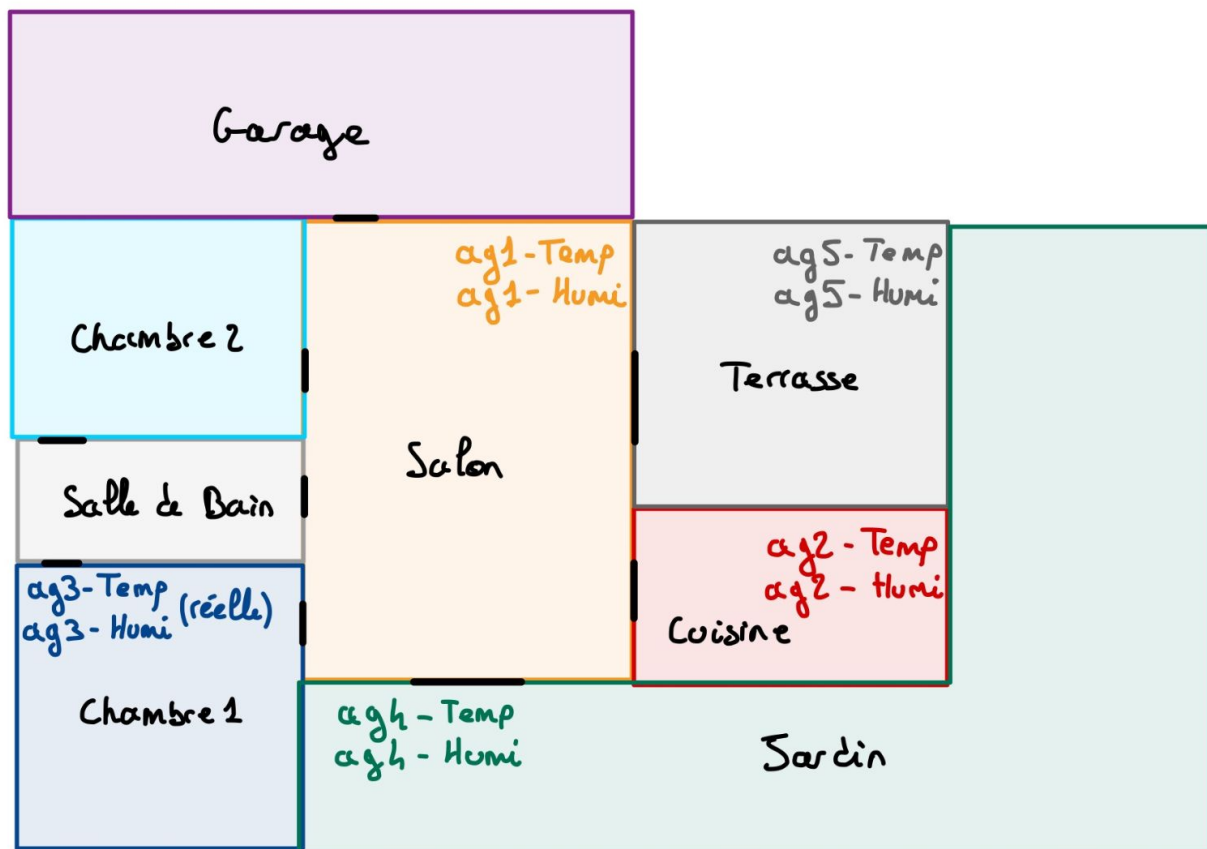
https://drive.google.com/file/d/1n_MUJGISluL71GSQf9KIAUYidzoLf_6i/view?usp=sharing

Choix de conception

Pour la réalisation de notre système, nous avons effectué plusieurs choix de conception, les voici :

- Disposition des capteurs dans la maison

Nous avons disposé dans plusieurs pièces de la maison un capteur de température et d'humidité selon le schéma suivant :

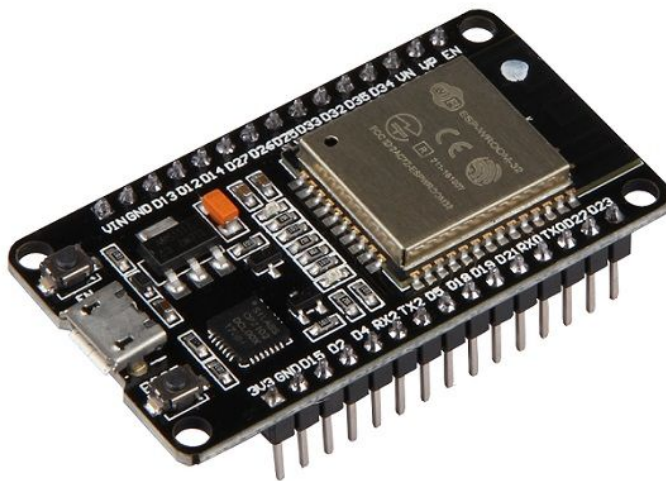


Chaque agx correspond à un capteur. Toutes les pièces n'ont pas été équipées. Nous avons donc 5 capteurs de température et 5 capteurs d'humidité.

NB: Pour faciliter le travail, seuls les capteurs de la chambre 1 sont réels. Les autres sont simulées à partir de valeurs générées aléatoirement.

- Choix du hardware

Pour réaliser nos capteurs nous avons choisi d'utiliser différentes cartes électroniques.



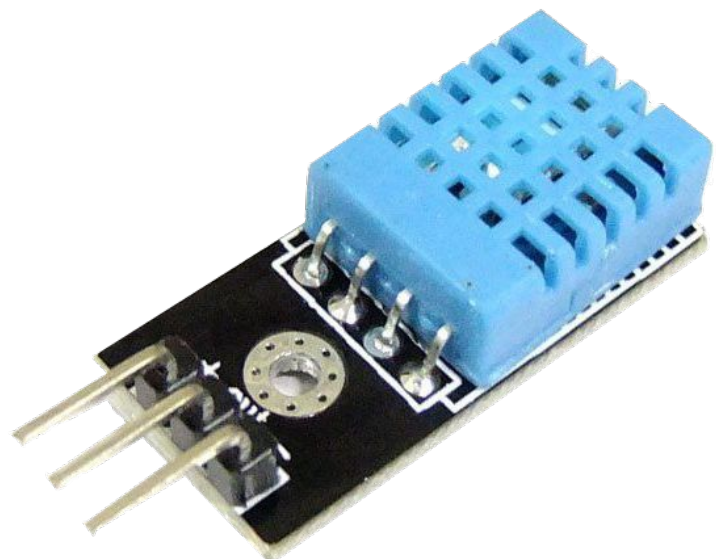
Une carte ESP32

Cette carte possède un cpu capable d'effectuer des traitements ainsi que de communiquer à l'aide de sa connectivité Wi-Fi et Bluetooth. Elle possède aussi un grand nombre d'entrées-sorties ce qui permet de lui connecter une large sélection de capteurs.

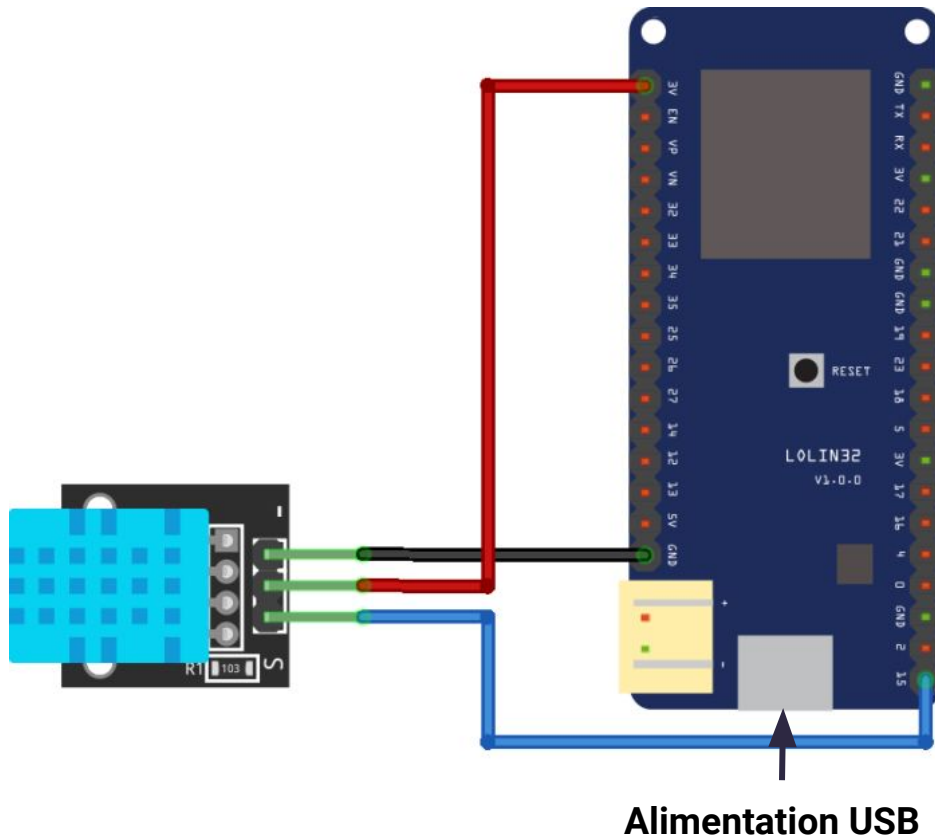
Dans notre système, elle est connectée à un capteur DHT11 ainsi qu'au Wi-Fi domestique pour envoyer les données du capteur.

Un capteur de température et d'humidité DHT11

Le capteur de température et d'humidité DHT11 renvoie des informations de température et d'humidité au Soc externe (ESP32) via un port série. C'est un des capteurs les plus populaires et les moins coûteux du marché.



Voici le schéma du montage



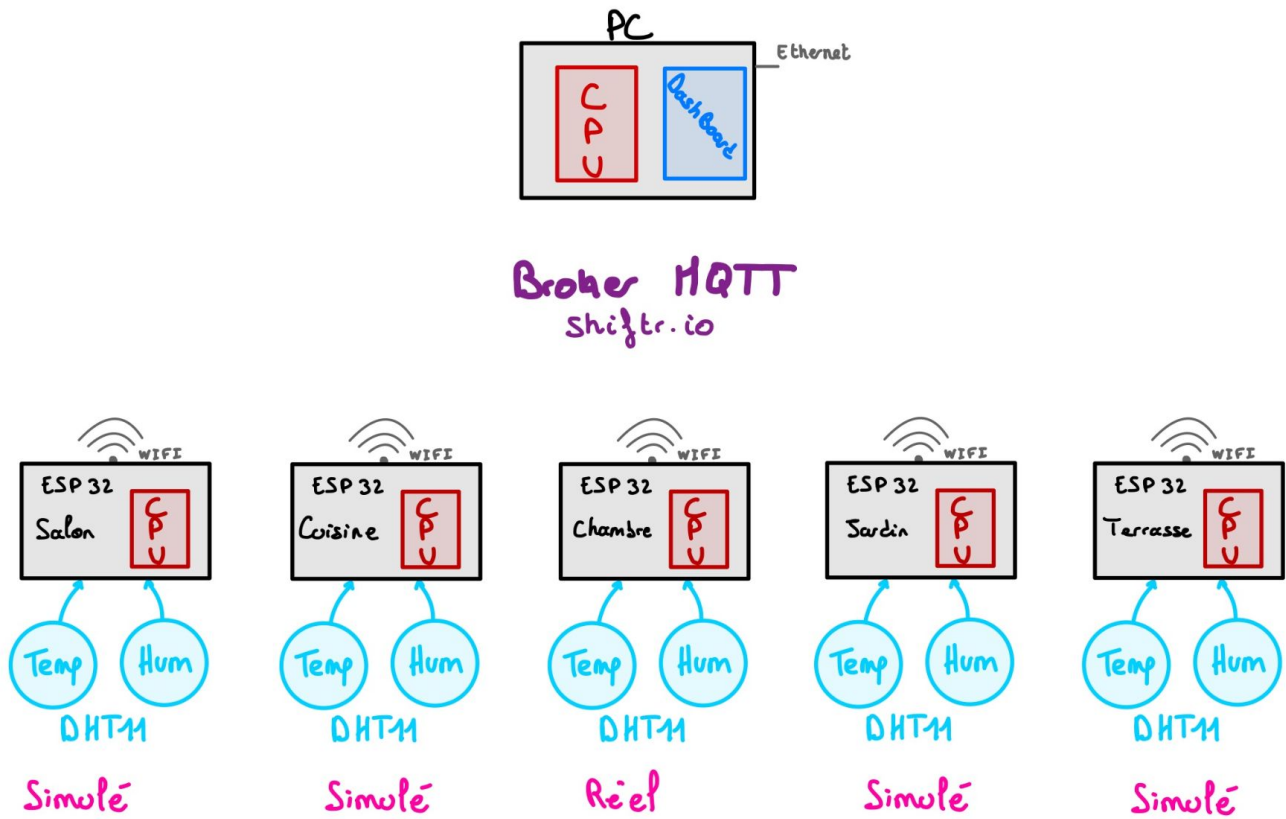
- Choix du protocole de communication

Pour que nos capteurs communiquent entre eux, il faut qu'ils utilisent un protocole de communication. Nous avons choisi d'utiliser le protocole MQTT avec le broker shiftr.io. MQTT, pour "Message Queuing Telemetry Transport", est un protocole open source de messagerie qui assure des communications non permanentes entre des appareils par le transport de leurs messages. Très utile dans notre système, c'est lui qui assure la liaison entre nos différentes entités.



- Architecture du système

L'architecture de notre système est la suivante:



Avantages, Inconvénients et limites de notre solution

- Avantages

Notre solution est rapide à mettre en place, elle ne nécessite pas de grandes infrastructures réseaux ou hardware notamment grâce aux capacités sans-fil des esp32. De plus, la transmission des données est rapide et sécurisé au moyen du protocole MQTT.

- Inconvénients

Notre solution fait appel au broker MQTT shiftr.io celui-ci est très bien pour le prototypage mais la formule gratuite ne permet l'ouverture d'un broker pour seulement 6h. Les formules avec un temps illimité sont payantes ce qui ajoute un coût à la solution. Il pourrait être bien d'essayer d'autre broker comme Mosquito ou d'autres solutions comme l'API Rest.

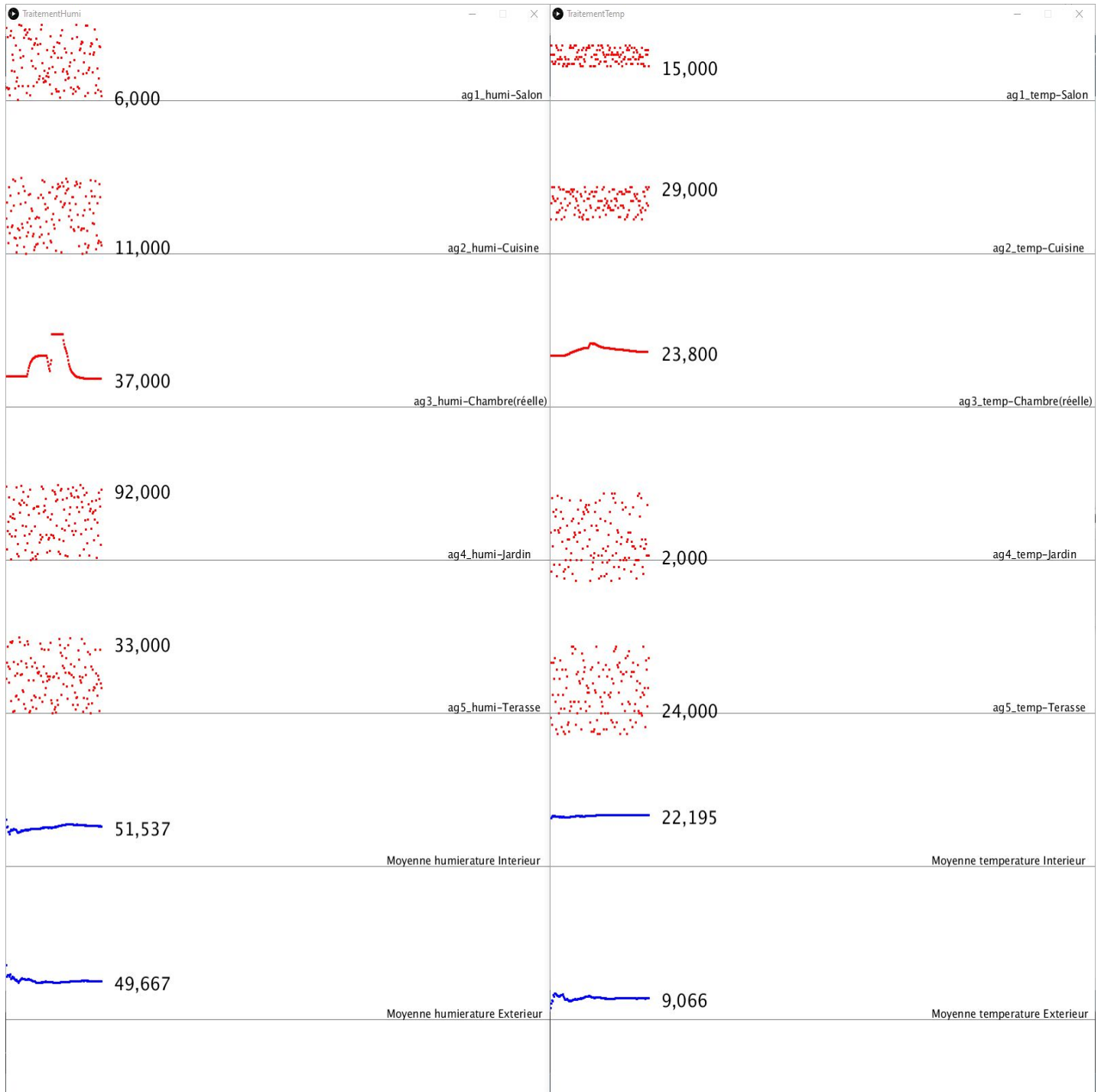
- Limites

L'ajout de capteurs dans notre solution peut être un peu compliqué pour les non initiés. En effet notre solution ne gère pas l'arrivée et le départ de nouveaux capteurs sur le réseau. C'est une voie d'amélioration.

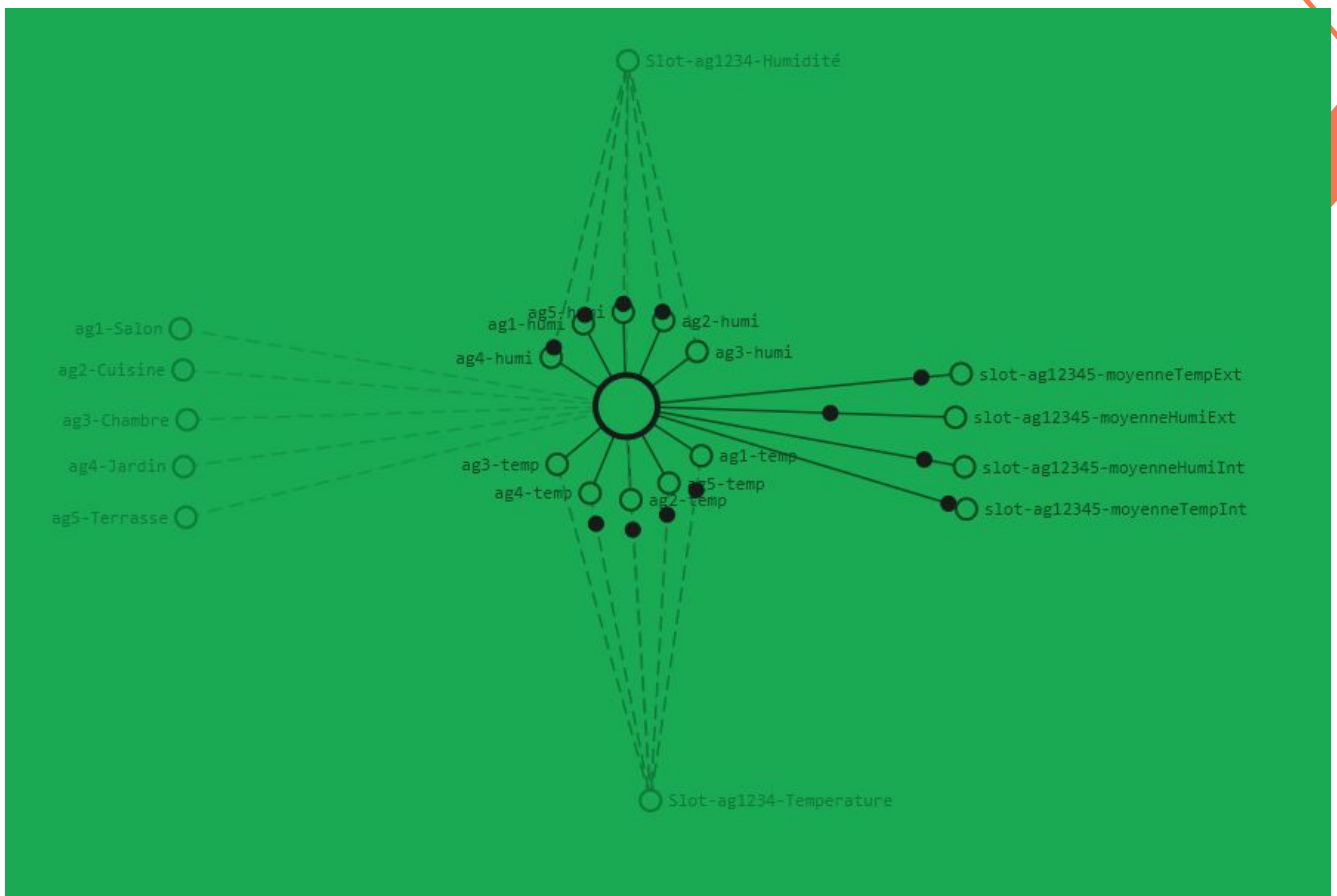
Illustrations

Voici quelques illustrations d'exemples de notre application. Voir aussi la vidéo ci-jointe:

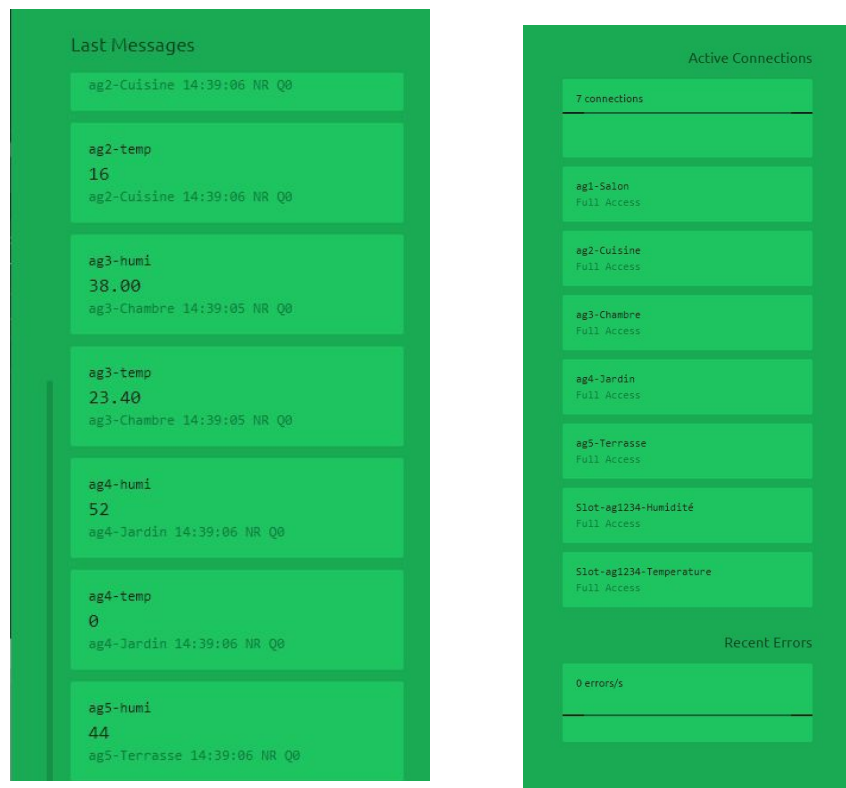
https://drive.google.com/file/d/1n_MUJGISluL71GSQf9KIAUYidzoLf_6i/view?usp=sharing



Les deux dashboards de suivi de l'évolution des données



Le broker MQTT sur Shiftr.io où toutes les données sont récupérées



Les messages reçus et les clients connectés

Conclusion

L'interaction distribuée est en voie de devenir de plus en plus utilisée avec l'arrivée de l'IoT (Internet of Things) où chaque objet physique sera connecté à un large réseau.

Ce projet nous a permis d'aborder quelques-unes des technologies qui constitueront cette révolution.