



ID-PHYCS - Interactive Disseminated PHYsical Complex Sensors

Le sujet du projet

Nous allons concevoir un système permettant de collecter et superviser un ensemble de capteurs IP et non IP disséminés dans le bâtiment U3 de l'UPS.

Les capteurs sont regroupés par **agrégats** (l'ensemble des capteurs d'une salle ou d'un bureau) et un **slot** est composé d'un ensemble de salles et/ou de bureaux. Il y a donc 4+1 **slots** dans le U3 (l'administration de l'UPSSITECH formant un **slot** à part). Nous allons spécifiquement nous intéresser au réseau de l'administration (cf. Figure 1) dans ce projet.

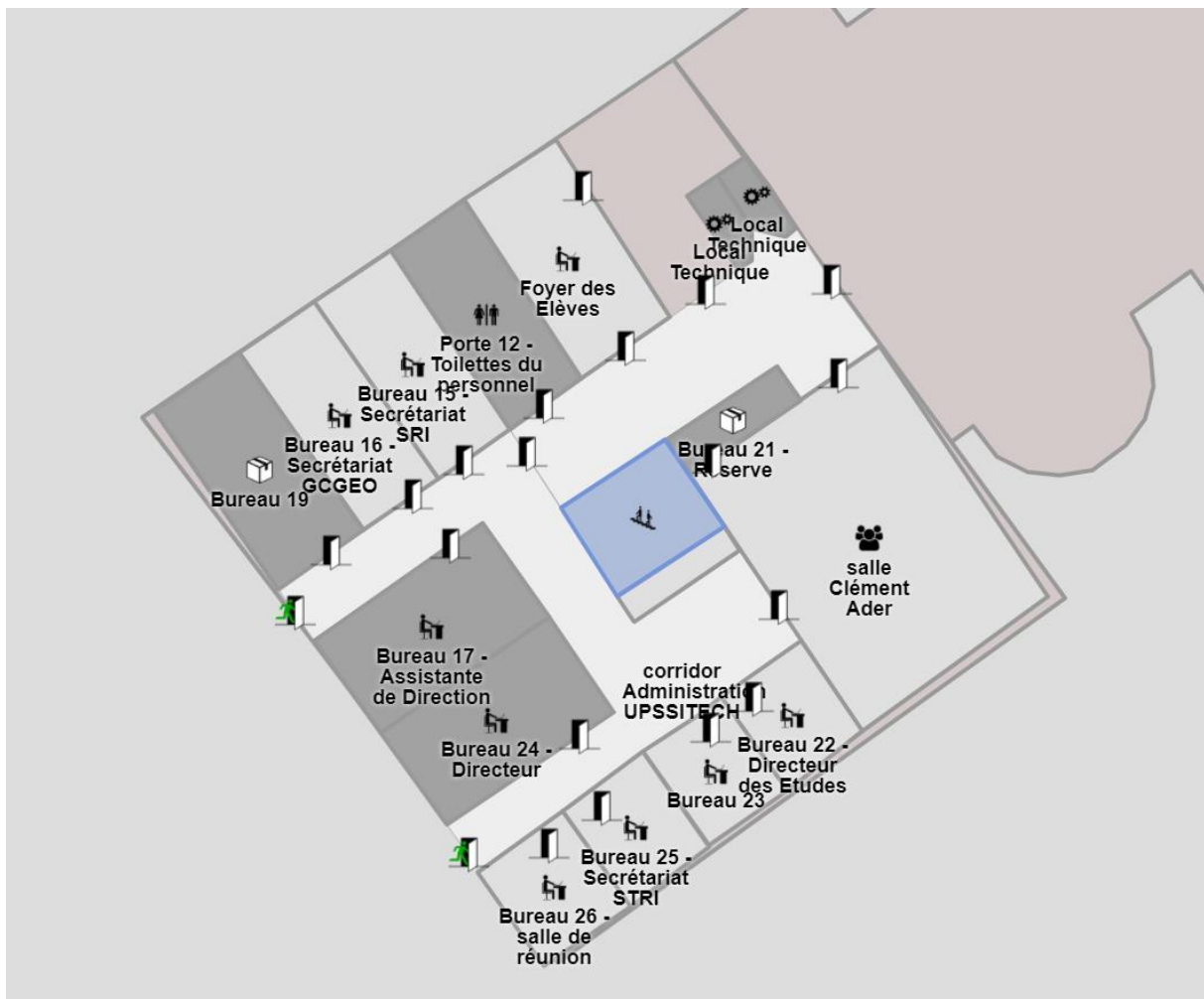


Figure 1 – le slot de l'UPSSITECH (<https://openlevelup.net/?l=0#18/43.56172/1.46952>)

Les capteurs sont soit des **capteurs de terrain** qui émettent régulièrement (fréquence propre à chaque capteur) vers un **agrégat**, soit des **capteurs « virtuels »** (accessibles via des API REST) qui sont connectés directement à un **slot**.

Les capteurs de terrain peuvent être reliés via différents protocoles matériels et/ou logiciels (liaison série, USB, radio RF 433, Zigbee, BLE, etc.). Les **agrégats** stockent les données « localement ».

L'objectif de ce projet est de permettre de *gérer le plus finement possible* ce réseau des capteurs d'une part et de proposer un dashboard permettant de visualiser, gérer les données et prévoir de potentiels problèmes.

Etape 1 : une architecture générale

Dans cette étape, vous devez choisir l'architecture de votre application s'appuyant sur **une** ou **plusieurs** technologies vues en TP (ou autres) pour la liaison entre les capteurs, **agrégats** et **slot** d'étage.

Un **agrégat** consiste pour ce projet en un « *certain nombre* » (au moins 2) de capteurs de terrain (de **température, présence, consommation électrique, ...**) disséminés et localisés dans le bâtiment qui renvoient périodiquement leur mesure. Les **agrégats** stockent la configuration (fichier JSON) de la salle (nombre de capteurs, type, localisation, fréquence de remontée, ...) et traitent les informations reçues (calcul de moyenne lissée des différentes valeurs des capteurs par exemple).

Chaque **agrégat** renvoie ses données vers un **slot** exposé sur internet via une API REST toutes les 10 mn (6 fois par heure). N'importe quel client peut récupérer à tout moment ces données auprès du slot grâce à une requête web (le type de requête est à définir).

Proposez une architecture réseau supportant ce cahier des charges.

Etape 2 : une preuve de concept

Cette étape permet de mettre en œuvre la version initiale du réseau de capteurs. Le réseau est ici composé d'un **slot** et d'un **agrégat** composé d'au moins deux capteurs.

Vous devrez utiliser au moins un agent capteur « **réel** » (sur arduino, nodeMCU, RPi Zero ou 3), etc.) Les autres peuvent être simulés logiciellement.

Nota : le matériel pourra être prêté (micro-contrôleur ou Raspberry + capteurs) sur demande pour le développement du projet.

Développez une première version du réseau de capteurs.

Etape 3 : un peu plus sur des agrégats

Dans cette étape, le **slot** souhaite pouvoir récupérer la configuration courante des différents **agrégats** (son état au cours du temps, combien de capteurs sont connectés sur l'**agrégat**, leurs types, leur localisation dans la salle, etc.).

Les **agrégats** devront se faire connaître auprès du **slot** pour se faire enregistrer à leur lancement.

Ajoutez un second agrégat dans votre système.

Utilisez un échange de données JSON entre le slot et les agrégats pour permettre de récupérer la configuration.

Etape 4 : Un service opérationnel

Dans cette dernière étape, on souhaite **proposer un dashboard** permettant de récupérer et d'afficher facilement toutes les données du **slot** issues des **agrégats** avec une carte permettant la localisation des capteurs (vous pouvez utiliser par exemple l'API OpenStreetMap avec la couche OpenLevelUP!) et les données de chaque capteur ou de leur **agrégat**.

Proposez sur le dashboard une détection de toute valeur anormale dans les capteurs, codez les alertes correspondantes et proposez une prédiction sur l'évolution des valeurs. (Le dashboard peut être codé sur le web ou via une interface graphique « classique »).

Le projet s'effectue en binôme. Vous devrez remettre à l'issue du projet :

- **Un rapport** contenant la liste de vos choix de conception illustrés par des schémas, copies d'écran ou vidéo explicative.
Faites un bilan des avantages/inconvénients/limites de votre solution. Tâchez d'être le plus objectif possible !
- Le **code source** de vos développements (repository **git** ou archive **zip**) ainsi que l'**exécutable** éventuel.

L'ensemble des documents et liens devra être envoyé par **mail** (archive en pièce jointe ou lien de téléchargement) à Philippe Truillet (Philippe.Truillet@univ-tlse3.fr) **avant le dimanche 16 janvier 2022 23h55 UTC.**

0,25 pt de pénalité par jour de retard sera appliqué au-delà de cette date.

Vous pourrez aussi convenir d'un rendez-vous pour exposer oralement votre travail si vous le désirez.