



Projet « My Smart Quiet Locality »

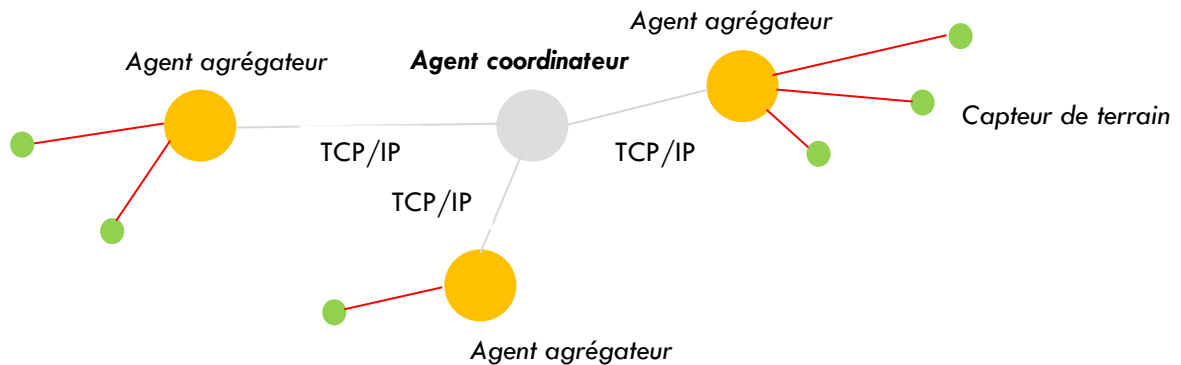
http://www.irt-systemx.fr/wp-content/uploads/2016/02/Picto_TerritoiresIntelligents-300x300.png

Le sujet du projet

Nous voulons concevoir un système permettant de gérer une série de capteurs disséminés dans un environnement ouvert (jardin et mobilier urbain) et fermé (domiciles).

Le réseau est composé de différents types d'**agents**, notamment les **capteurs de terrain** qui envoient leurs données à un **agent agrégateur** qui se connecte via TCP/IP à l'**agent coordinateur** (qui joue le rôle du « serveur » central).

Les agents capteurs peuvent être reliés aux agents agrégateurs par différents protocoles matériels et/ou logiciels (liaison série, USB, radio, Zigbee, BLE, etc.).



L'objectif est de permettre de gérer *facilement* le réseau des capteurs d'une part et de **proposer une API REST** permettant d'interroger les données au travers d'internet d'autre part.

Etape 1 : proposition d'une architecture

Dans cette étape, vous devez choisir l'architecture de votre application s'appuyant sur **une** ou **plusieurs** technologies vues en TP.

Le réseau des capteurs consiste pour ce projet en un « *certain nombre* » de capteurs de **température**, **pression atmosphérique** et **humidité** disséminés qui renvoient périodiquement leurs mesures à un agent agrégateur auquel ils sont connectés.

L'**agent agrégateur** possède de 0 à n capteurs « *locaux* » dont il stocke et traite les informations reçues (localisation (Ex : cuisine, salon, jardin ...), calcul de moyenne de T° par exemple).

De manière nominale, chaque agent agrégateur envoie ses informations périodiquement (1 fois par minute par exemple) à un **agent coordonnateur** qui reçoit des données provenant de 0 à m agents agrégateurs.

Le coordonnateur peut quant à lui récupérer sur demande ces données auprès des agrégateurs à tout moment.

Proposer une architecture réseau supportant ce cahier des charges.

Etape 2 : développement initial

Cette étape permet de mettre en œuvre une version initiale du réseau de capteurs. Le réseau est ici composé d'un agent coordinateur, de deux agents agrégateurs, d'un capteur pour le premier

agrégateur et d'au moins deux capteurs pour le second.

Développez une première version du réseau de capteurs correspondant.

Vous pouvez utiliser des agents capteurs « réels » branchés sur arduino, NodeMCU, RPi, etc (c'est un plus pour le projet) ou simulés logiciellement.

Étape 3 : configuration des agents agrégateurs

Dans cette étape, l'agent coordinateur souhaite pouvoir accéder à la configuration des différents agents agrégateurs (combien de capteurs sont connectés par agrégateur, leur type, leur localisation, etc.)

Modifiez votre système afin de supporter cette nouvelle contrainte. Utilisez si possible un échange de données JSON.

Étape 4 : supervision

Dans cette dernière étape, on souhaite **développer une API REST** permettant de récupérer facilement soit les données des agents agrégateurs, soit les données individualisées par capteur via le réseau internet.

Développez une interface graphique (avec Processing par exemple) illustrant cette requête.

Étape 5 : ajout/suppression dynamique des agrégateurs (optionnel).

Dans cette dernière étape, on souhaite gérer l'arrivée et le départ des agents agrégateurs de manière dynamique. Un agrégateur peut être mobile (cas d'un smartphone par exemple) ou tomber en panne et donc disparaître de la liste de l'agent coordonnateur.

Chaque agent coordonnateur possède sa liste des agrégateurs connectés à lui qu'il interroge périodiquement si aucune donnée ne lui parvient.

Au bout de trois demandes infructueuses de reconnexion, l'agrégateur concerné est éliminé définitivement de sa liste et une alerte est remontée.

A l'inverse, quand un nouvel agrégateur sur le réseau, il demande une connexion à l'agent coordonnateur qui le rajoute dans sa liste.

Modifiez votre système en conséquence.

Étape 6 : conclusion

Faites un bilan des avantages/inconvénients/limites de votre solution. Tâchez d'être le plus objectif possible.

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.
- Le **code source** de vos développements (archive zip) ainsi que l'**exécutable éventuel**

L'ensemble des documents devra être envoyé par mail (archive en pièce jointe ou lien de téléchargement) à Philippe Truillet (Philippe.Truillet@irit.fr) **avant le vendredi 18 janvier 2019 23h55 UTC.** ;

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

Vous pouvez aussi convenir d'un rendez-vous pour exposer oralement votre travail.