Développement et déploiement d’une application OneM2M pour bâtiments intelligents

**Étudiant :**

* Abdelrazak El Ghazzaz
* Jean-Baptiste Lognon

**Organisme d’accueil – Lieu de travail** **:**

* UDN - University Of DA NANG
* UTE - University of Technology and Education
* DNIIT – Da Nang international Institue of Technology

**Enseignant référent :**

* NGUYEN Thanh Tuan
* Marie-Agnès PERALDI

Cahier des charges Mai 2019

Table des matières

[Résumé 2](#_Toc8311299)

[Abstract 2](#_Toc8311300)

[Contexte 3](#_Toc8311301)

[Besoins 4](#_Toc8311302)

[Grandes tâches à réaliser 6](#_Toc8311303)

[User stories 8](#_Toc8311304)

[Liste des tâches à réaliser 9](#_Toc8311305)

[Diagramme de GANTT 10](#_Toc8311306)

[Glossaire 10](#_Toc8311307)

## Résumé

Dans le cadre de notre DUT, nous devons effectuer un stage d’une durée minimale de 10 semaines. Nous avons ainsi rejoint le groupe SLEGO au sein du DNIIT, une université située à Da Nang au VIETNAM, en collaboration avec l’Université Nice Sophia Antipolis, afin de travailler sur le déploiement d’une plateforme qui utilise les standards du consortium OneM2M, pour gérer des objets connectés au sein de bâtiments intelligents. La finalité de notre projet consiste à réaliser des maquettes qui permettront d’automatiser le déploiement de cette plateforme, reliée à des objets connectés (lampe, chauffage, etc.…), en tenant compte des différentes contraintes qui sont leur localisation, les aspects de sécurité liés à l’accès de la plateforme de contrôle, la maintenance des objets, c’est-à-dire la gestion de leur état et enfin la gestion automatique de l’installation de nouveaux équipements. Cela comprend donc l’enregistrement des équipements et applications, le traitement des données, les communications, les protocoles au travers de l’appel de services web, tout cela via l’utilisation d’une version open source du standard oneM2M et de la programmation de carte Arduino en utilisant l’IDE Arduino pour gérer le fonctionnement des applications externes (capteurs/actionneurs).

## Abstract

As part of our our *wo-year university degree in technology*, we must complete an internship of a minimum duration of 10 weeks. We have thus joined the SLEGO group within the DNIIT, a university located in Da Nang, VIETNAM, in collaboration with the University of Nice Sophia Antipolis, to work on the deployment of a platform that uses the OneM2M consortium standards to manage objects connected within intelligent buildings. The purpose of our project consists in creating models that will automate the deployment of this platform, connected to connected objects (lamp, heating, etc …), taking into account the different constraints that are their locations, the security aspects related to access to the control platform, the maintenance of the objects, i.e. the management of their status and finally the automatic management of the installation of new equipment. This includes the abstraction of equipment and application registration, data processing, communications, protocols through the web services call all this through the use of an open source version of the oneM2M standard and Arduino card programming using the Arduino IDE to manage the operation of external applications (sensors / actuators).

## Contexte

À l'heure actuelle, aucune disposition ni mesure n'est prise concernant la gestion automatique des différents équipements installés dans l'université. Tout cela est géré manuellement, que ça soit l'éclairage, le chauffage ou bien la climatisation. L'automatisation de ces tâches représente un enjeu majeur en ce qui concerne la consommation d'énergie, notamment aujourd'hui où la question de l'impact énergétique mondiale sur l'environnement est au centre de toute attention. Nous pouvons également voir un aspect sécuritaire à l'aide de capteurs de présence qui permettront de voir si une personne est présente dans l'enseigne. 

Pour ce stage, nous disposons de deux emplacements de travail, la première au sein du bureau du DNIIT et ainsi qu’au niveau de l’UTE (université partenaire de l’UD). Le DNIIT est une structure installée dans le bâtiment VN-UK qui fait partie des 10 établissements affiliés à l’UD. Elle a été créée en 2017 suite à une collaboration entre l’UD, l’AUF et l’UNS. Celle-ci dispose de 3 centres :

* MIRE est un centre de recherche collaborative, d’échange d’informations scientifiques et de transfert de technologie qui vise à établir des projets à l’aide de partenaire nationaux et internationaux (chercheurs, étudiants, etc …)
* NiceCampus gère la formation et la communication scientifique dans la communauté, comportant l’intégration progressive de plusieurs parcours internationaux de formations aux niveaux licence, master et doctorat ainsi que des équipes mixtes de recherche
* CNFp gère la formation et la communication scientifique dans la communauté francophone

Donc le DNIIT a pour but de renforcer le développement de l'enseignement et de la recherche au niveau de l'enseignement supérieur de la ville de Da Nang ainsi que de ces partenaires nationaux et internationaux. Elle dispose ainsi de plusieurs unités de recherche telle que SLEGO, EMOTICA, SUSHA, POSCA, PAPIOT et SWAT. Toutes spécialisées dans des domaines différents (internet des objets, e-santé, l’étude des émotions, l’étude de la pollution, etc…).

Nous avons ainsi intégré le groupe de recherche SLEGO qui travaille sur la définition des langages pour la programmation des systèmes IOT (modélisation, conception, développement de système IOT y compris des aspect langages pour décrire les applications et les services pour l'IOT), dont les responsables sont Mme PERALDI-FRATI Marie-Agnès, M. Nhan LE THANH et NGUYEN Thanh Tuan, afin de réaliser des travaux sur les objets connectés. Pour ce faire, nous disposons d’un certains nombres de matériels, tel que des Raspberry PI, carte Arduino, carte Lora, etc...

## Besoins

La finalité du projet est donc de proposer une solution simple, fiable et sécurisée pour répondre aux besoins dans le domaine des bâtiments intelligents à savoir :

* L’automatisation du déploiement de la plateforme

En effet, nous devons rendre notre plateforme automatique/autonome. De ce fait nous devons faire en sorte, que lorsqu’un capteur est ajouté à la plateforme il s’enregistre auprès de cette dernière avec un nom et une localisation bien spécifique afin que ce dernier soit reconnaissable.

* L’interaction de la plateforme avec l’environnement

C’est-à-dire que lorsque la plateforme sera “mise en service” celle-ci recevra des données provenant des différents capteurs mis en place dans le bâtiment (exemple : température d’une pièce, niveau de luminosité, présence ou non d’un individu dans une zone, …). Ensuite, cette dernière effectuera des actions en fonction des données récoltées. Par exemple si une personne entre dans une pièce qui dispose de capteur de présence ainsi que de luminosité et que le niveau de luminosité dans la pièce est faible, la plateforme recevra des informations/donnés relatives à cette situation et devra donc allumer les lumières de la pièce (où se trouve les capteurs).

* L’optimisation du trafic

Dans un souci de surcharge, nous allons devoir optimiser le trafic des données envoyé par le capteur à la plateforme. En effet, si les capteurs envoient des données toutes les minutes par exemple, celles-ci risquent (pour certain capteur comme de présence) d’envoyer des flux d’informations redondant. Pour pallier ce problème, nous allons donc rendre les capteurs de l’infrastructure “intelligent”. Cela signifie donc que les capteurs devront envoyer des données à la plateforme seulement lorsqu’une condition est activée (par exemple, seuil de température atteint).

* L’optimisation énergétique

Pour répondre aux problèmes liés à la consommation d’énergie et ainsi réduire son impact sur l’environnement, il est nécessaire et indispensable d’étudier les différents moyens de communications existants utilisés par les capteurs pour les envois d’informations. Avec l’internet des objets (IOT), plusieurs moyens de communication sont possibles pour la diffusion des flux telle que le Bluetooth, Zigbee/Zibwave, Wifi, Lora, etc... En fonction de la disposition, il faut adapter les moyens de communication. Par exemple, si la communication se fait dans une petite pièce, il n’est pas nécessaire d’utiliser le wifi pour transmettre des données bien qu’il soit bien déployé et ayant un débit élevé, cette technologie consomme beaucoup d’énergie ce qui pourrait vite réduire l’espérance de vie d’une batterie d’un capteur. Alors qu’il serait préférable dans notre cas, d’utiliser Lora ou bien le Bluetooth qui nécessite très peu d’énergie pour leur fonctionnement.

* Une maintenance prédictive

La plateforme doit être capable d’effectuer des scans afin de voir l’état des applications externes ou bien de programmer les cartes Arduino pour faire en sorte que les applications externes envoient des données à la plateforme lorsqu’elles subissent un changement d’état afin de localiser et prévoir un remplacement de celle-ci.

* Stockage/Gestion de données

Soit la mise en place d’une base de données qui serait nécessaire pour le stockage des informations si l’on veut garder les différents échantillons reçus ou bien faire en sorte de garder un nombre limité d’échantillon qui se met à jour à chaque fois que la plateforme reçoit des informations.

Afin de résoudre ces besoins, nous allons devoir utiliser et adapté différentes technologies déjà existantes.

Tout d’abord nous avons le standard OneM2M dont le but est de rendre interopérable le M2M (Machine To Machine) et les technologies IoT. L’architecture de OneM2M est composée de 3 “partie” :

* L’Application Layer qui est composée d’Application Entity (AE) qui utilise les services des CSE (voir ci-dessous) et qui fournit la partie applicative de la solution M2M.
* Le Common Service Layer qui lui est composée de Common Service Entity (CSE) qui propose un ensemble de service au travers d’interface standardisées et fournit les fonctions communes aux environnements M2M.
* Le Network Service Layer qui est composée de Network Service Entity (NSE) qui assure la communication entre les équipements et les réseaux. Les NSE ont donc pour objectif de fournir un service de transport de données aux CSE.

Ensuite nous avons les API REST (REpresentation State Transfer) qui sont dédié à la conception de web service. De plus les API REST ont une architecture orientée ressource ce qui signifie que chaque entité est une ressource et est accessible par un URI (Uniform Ressource Identifier). Enfin les requêtes REST sont basées sur HTTP qui utilise la couche réseau TCP. Celle-ci définit opérations GET, POST, PUT, DELETE pour accéder aux ressources stockées.

Nous avons également les différentes technologies de communications nécessaires aux fonctionnements de l’IOT tel que :

* Le Bluetooth qui a comme caractéristique une faible consommation et est sans-fil. Cependant il présente un débit moyen donc inadapté à une transmission rapide d’informations ou de grande portée
* Zigbee/zigwave qui propose une transmission de donnée facile, une faible consommation énergétique et un état dormant des objets
* Wifi qui propose un débit élevé et présente un avantage dû au fait qu’il est bien déployé, mais qui nécessite une consommation plus importante d’énergie
* Lora avec une basse consommation d’énergie mais qui propose une grande portée avec un faible débit.

Enfin il est existé un protocole permettant un abonnement a des données produites ou consommées par une entité dans le but d’obtenir leur valeur au fur et à mesure des nouvelles publications par l’entité. Le protocole se nomme MQTT.

## Grandes tâches à réaliser

Pour notre projet il y a différents types d’utilisateurs qui vont interagir avec la plateforme à savoir :

* Celui qui la conçoit
* Celui qui la déploie
* Celui qui la maintient
* Celui qui l’utilise

Et chaque utilisateur (ou groupe d’utilisateurs) aura différentes tâches à réaliser.

Tout d’abord nous avons celui qui conçoit la plateforme. Ce dernier, devra avoir la possibilité d’intégrer des objets paramétrables dans la plateforme (il pourra les modifier, en ajouter ou en retirer).

Dans un premier temps nous allons devoir identifier les besoins en matière de données et de caractérisation des différents éléments de la plateforme, à savoir :

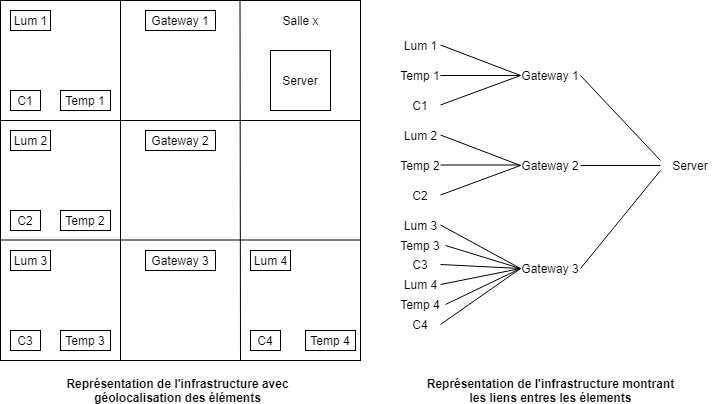
* De quoi a besoin la plateforme pour être développée ?
* Comment allons-nous devoir caractériser les capteurs ?
* Quels types de capteur/actionneur devrons-nous utiliser ?
* Où devrons-nous placer les capteurs/actionneurs ?
* Quelle période d’échantillonnage allons-nous utiliser pour ces capteurs/actionneurs ?
* Quels types de mesures (températures, luminosité, etc…) les capteurs devront-ils réaliser ?

À terme, cette identification nous permettra de savoir quel matériel (capteurs, actionneur, passerelle, etc.) est à quel endroit dans notre infrastructure.

Dans un même temps, nous devrons construire une structure de donnée qui va nous permettre de caractériser l’ensemble des objets. De plus cela servira de support à l’application en nous permettant d’obtenir des correspondances entre élément et ressource OM2M. Par exemple, un capteur donnera donc une AE avec “x” paramètres.

Enfin nous aurons la phase de développement. Dans cette phase, nous allons devoir déployer notre propre structure en nous basant sur le type de code qui a déjà été développer (comme dans les TPs) en JAVA. Ainsi nous allons devoir réaliser :

* Le déploiement de capteurs
* Le déploiement de nœuds



Légende :

* + Lum x : Capteurs de luminosités
  + Temp x : Capteurs de températures
  + Cx : Autres types de capteurs
  + Gateway x : Passerelle (un Raspberry)
  + Server : Machine qui va recevoir les données des capteurs
  + --------- : Communication entre deux éléments

Nous avons ensuite un deuxième type d‘utilisateur, celui va maintenir la plateforme. Il devra quant à lui pouvoir effectuer des modifications dans l’infrastructure (déjà existante), par exemple ce dernier, va devoir gérer le remplacement d’un capteur dans l’infrastructure en cas de panne de ce dernier. Dans ce cas de figure, nous allons devoir définir à partir de quel moment nous pouvons considérer qu’un capteur est en panne. Par exemple si aucune donnée n’a été reçu par l’infrastructure depuis 24h, alors on considère que le capteur est en panne et qu’il faut le remplacer. De ce fait, le nouveau capteur devra reprendre certains paramètres de l’ancien capteur à savoir : sa localisation, son nom, et tout autres paramètres.  En ce qui concerne les actionneurs, afin de connaitre leurs états, il faudrait quand on leurs demandent d’effectuer des actions, vérifier qu’ils les effectuent bien. En effet, si on demande à un actionneur d’allumer une lumière par exemple, et que celui-ci ne le fait pas (et ceci de manière récurrente) alors on pourra considérer cet actionneur comme étant en panne et le remplacer de la même manière qu’un capteur.

Par la suite nous allons devoir créer/coder diffèrent scenario à réaliser en fonction des données envoyées par les capteurs.

Exemple de scénario : s'il fait moins de 18 degrés Celsius dans une pièce entre 8h et 19h, alors nous devrons allumer le chauffage dans celle-ci.

Pour finir, nous allons devoir sécuriser notre infrastructure afin que seules les personnes autorisées (disposant d’un login et mots de passe) puissent interagir avec la plateforme (le login et mots de passe ne doivent pas être en dur, mais saisi par l'utilisateur).

## User stories

**Concepteur de la plateforme**

* En tant que concepteur de la plateforme, je veux pouvoir créer des objets paramétrables qui correspondent aux différentes caractéristiques des équipements déployés, afin de pouvoir les gérer et récupérer les informations essentielles de celles-ci.
* En tant que concepteur de la plateforme, je veux pouvoir sécuriser les canaux de communication, afin d’éviter qu’un tier ne puisse interpréter le message échangé entre un client/objet et le serveur. Le système répond à mes attentes, s’il chiffre la communication.

**Concepteur de service**

* En tant que concepteur de service, je veux pouvoir automatiser l’utilisation des services sur les objets afin que la plateforme soit autonome.
* En tant que concepteur de service, je veux pouvoir attribuer des droits d’accès aux ressources, afin d’éviter qu’un autre utilisateur ne puisse modifier une ressource. Le système répond à mes attentes s’il permet de spécifier les droits d’accès aux ressources par rapport aux utilisateurs.

**Mainteneur**

* En tant que mainteneur, je veux pouvoir accéder aux informations qui caractérise les équipements, afin de gérer le remplacement des équipements dans l’infrastructure en cas de panne.

**Utilisateur**

* En tant qu’utilisateur, je veux pouvoir avoir un accès à une interface qui représente la plateforme afin de pouvoir observer, contrôler les objets.

## Liste des tâches à réaliser

**T1** : Mise en place de passerelles via Raspberry Pi (Lora gateway) et d’un serveur.

**T2** : Relier la/les passerelle(s) au serveur et installer le(s) protocole(s) d’échange choisi.

**T3** : Configurer l’envoie des données des capteurs dans les programmes des cartes Arduino et aussi structurer l’envoi des données par rapport à la bande passante pour éviter de surcharger celle-ci (MQTT).

**T4** : Faire la correspondance des différents composants avec la plateforme OM2M :

* Les passerelles hébergent des Middle-Node
* Le serveur héberge l’Infrastructure Node qui fournit un ensemble de services tel que l’enregistrement, gestion des données collectées, le contrôle d’accès etc.
* Les capteurs/actionneurs hébergent un ADN (Application Dedicated Node) fournit avec des AE (ADN-AE) qui permettent la collecte de données.

**T5** : Développement d’une fonction d’enregistrement automatique des capteurs/actionneurs auprès de sa passerelle pour la création d’AE ressource auprès des MN-CSE en fonction de la localité et des différents paramètres.

**T6** : Développement d’une fonction d’enregistrement automatique des passerelles (MN) auprès du serveur (IN) pour la création d’une ressource représentant la MN-CSE sous l’IN-CSE et pareil dans l’autre sens.

**T7** : Développement d’une fonction d’enregistrement automatique l’enregistrement des ADN-AE directement auprès du serveur (définir les cas où on doit remonter l’info directement au serveur.

**T8** : Développement d’une fonction qui réalise une requête de souscription de l’AE de la passerelle auprès de l’AE des actionneurs.

**T9** : Développement d’une fonction qui réalise des requêtes de création de ressource(container) des capteurs et actionneurs auprès de la passerelle.

**T10** : Développement d’une fonction chaque qui permet à chaque ADN-AE de créer une requête de souscription auprès des containers des capteurs.

**T11** : Intégration des différentes fonctions et utilisation dans un cas d'étude de gestion de la luminosité dans plusieurs pièces d’un bâtiment.

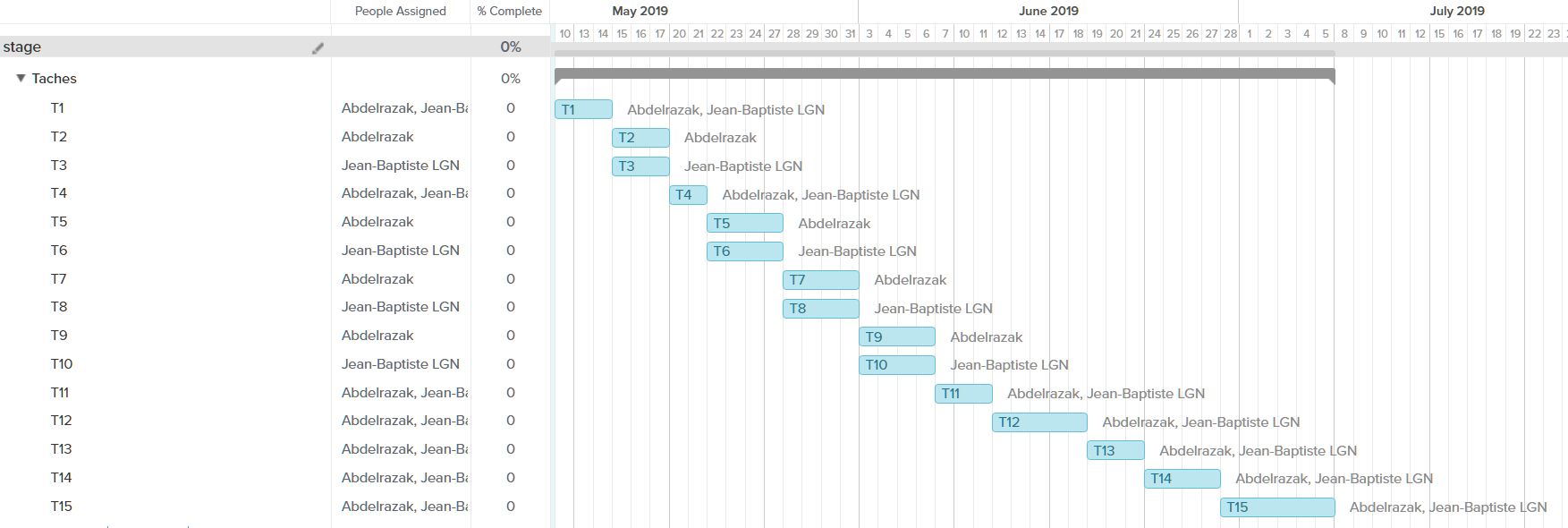
**T12** : Développement d’un langage de description de scénarios pour automatiser les actions à effectuer par le biais de l’interprétation des informations que les ADN-AE reçoivent grâce aux notifications des capteurs (dû à leurs souscriptions) lorsqu’il y a un changement (exemple : changement de température qui active une condition dans la programmation).

**T13** : Utilisation et démonstration du langage dans le même cas d'étude que T12.

**T14** : Programmation des capteurs pour la maintenance prédictive (renseignement de l’état en cas critique). Si le capteur ne le permet pas, mis en place d’un IPE pour contourner les limitations de technologie de celles afin de les remplacer par des fonctionnalités permettant de faire l’action voulue.

**T15** : Mettre en place la base donnée InfluxDB pour un stockage plus conséquent des données et ainsi pouvoir faire des analyses dans le temps (Grafana).

## Diagramme de GANTT



(Cf pièce jointe pour voir l’image si besoin)

## Glossaire

**Plateforme**

Une plateforme, est une “application” qui permet d’utiliser et de gérer des services.

**Infrastructure**

Une infrastructure est caractérisée par l’ensemble des éléments matériels qui compose un système.

**Capteur**Un capteur est un équipement utilisé pour prélever des informations/grandeurs physiques telle qu'une distance, température, etc...

**Actionneur**Un actionneur est un équipement qui produit une action physique (allumer une LED par exemple) à partir d’une énergie reçue (courant électrique, lumière, etc.).