

INTERGICIEL POUR L'INTERNET DES OBJETS

RAPPORT DE TP – 5 ISS INSA TOULOUSE

Ce rapport couvre l'ensemble des TPs Intergiciel pour l'Internet des Objets, il est individuel et sera intégré dans votre portfolio. Le but de ce rapport est de montrer les compétences acquises et les concepts compris et maîtrisés à la suite des cours et des différents TPs réalisés dans cette matière.

La structuration peut être modifiée, tant que les différents concepts sont abordés. Si des éléments supplémentaires vous semblent pertinents à détailler, libre à vous de les intégrer. La taille de ce rapport devra se limiter à **10 pages maximum** hors annexes. (N'hésitez pas à joindre des schémas en annexe si cela vous aide à clarifier votre discours)

Nom 1 : Gaigeot

Prénom 1 : Claire

1. SAVOIR POSITIONNER LES STANDARDS PRINCIPAUX DE L'INTERNET DES OBJETS

*Expliquez brièvement le principe du standard **oneM2M** et comment il se positionne vis à vis des autres standards et technologies déjà existantes.*

Le principe de OneM2M est de proposer un standard global pour le machine to machine et l'Internet des Objets. L'objectif de cette association est de simplifier l'implémentation de réseaux d'IoT complexe intégrant de nombreux dispositifs. Il s'agit d'un standard ouvert qui propose une vision assez générique et horizontale des systèmes connectés. Il répond au besoin d'une couche de service commune dans le domaine du machine to machine, sans regard du hardware ou du software. Ainsi avec son interface standardisé, oneM2M permet de faire abstraction des protocoles sous-jacent. Cette couche commune tend à intégrer différentes applications nécessitant par exemple leurs propres protocoles de communication. Ce standard peut s'appliquer à des secteurs très variés allant de la maison connectée, à la e-santé en passant par le transport intelligent ou l'automatique industriel.

Les dispositifs suivant le standard oneM2M doivent respecter de nombreuses caractéristiques tel que l'architecture de service, la sécurité ou les tests. Aujourd'hui, oneM2M est constitué de 8 organismes de standardisation et plus de 200 partenaires et membres participants.

Il existe d'autres technologies utilisées en machine to machine. Il y a de nombreux protocoles de communication comme les réseaux cellulaires, les réseaux basses consommation et autres réseaux sans fil. Il y a également de nombreux protocoles de transfert de données. Cela illustre la grande hétérogénéité des technologies disponibles. D'où l'importance d'un standard unifié qui soit capable d'interopérabilité. De nombreuses autres initiatives comme OneM2M existent. On peut citer ITU-T , MQTT, OASIS ou Open Mobile Alliance. Mais contrairement à OneM2M, ils ont tendance à se spécialiser à une application ou un domaine d'activité et offre une approche plutôt verticale.

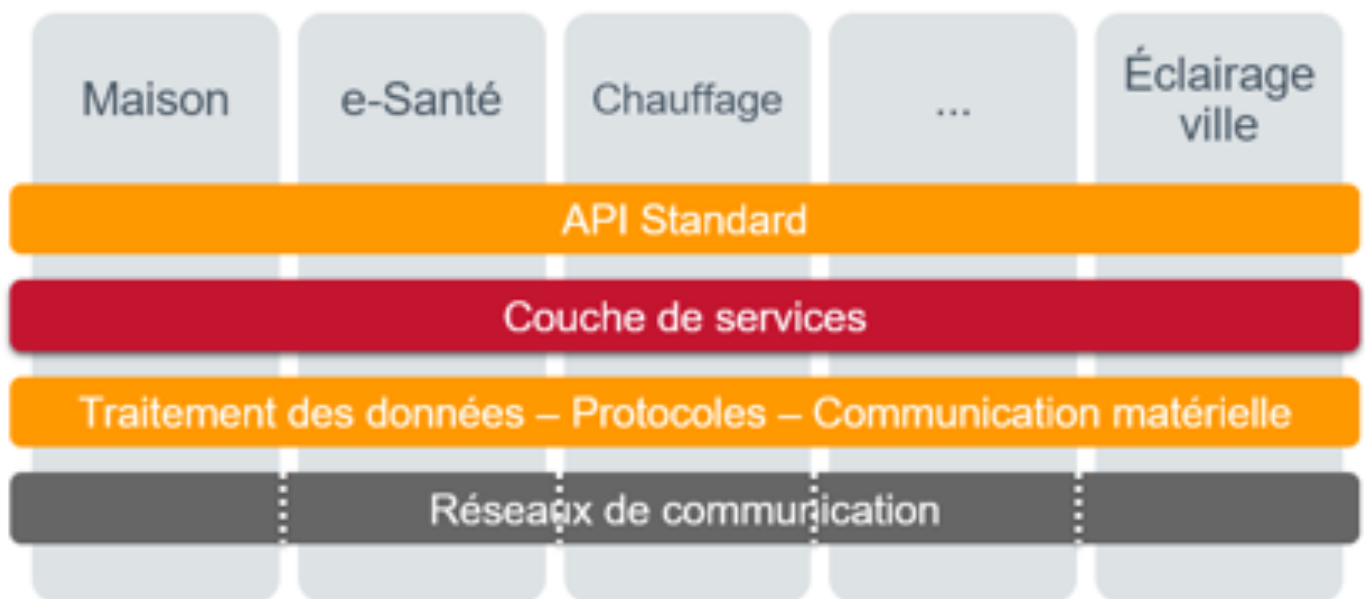


Fig 1 – La vision OneM2M (OneM2M MOOC OpenClassrooms – INSA)

2. DEPLOYER UNE ARCHITECTURE CONFORME A UN STANDARD ET METTRE EN PLACE UN SYSTEME DE RESEAU DE CAPTEURS AUX SERVICES

2.1. DEPLOYER ET CONFIGURER UNE ARCHITECTURE IOT EN UTILISANT OM2M

Expliquez comment vous avez déployé une **architecture IoT** avec **OM2M** : les types de nœuds utilisés, les entités en interaction avec l'intergiciel, à quel niveau les objets / capteurs interagissent avec le système ainsi que les différentes applications en interaction avec ce dernier, etc. (TP 1 + 2 +3)

OM2M est le logiciel Eclipse dédié au déploiement et à l'implémentation du standard OneM2M. C'est un logiciel open-source développé initialement par le LAAS-CNRS.

Le standard oneM2M définit un modèle à trois couches :

- La couche applicative AL (Application Layer) possède des entités appelées les AE (Application Entities) qui représentent les entités hauts niveaux, c'est-à-dire les applications du côté utilisateur ou server, voire même des objets.
- La couche service CSL (Common Services Layer) a des entités que l'on nomme CSE (Common Services Entities).
- La couche transport NSL (Network Services Layer), qui est gérée par OM2M, s'occupe de la communication réseaux et est transparente aux applications.

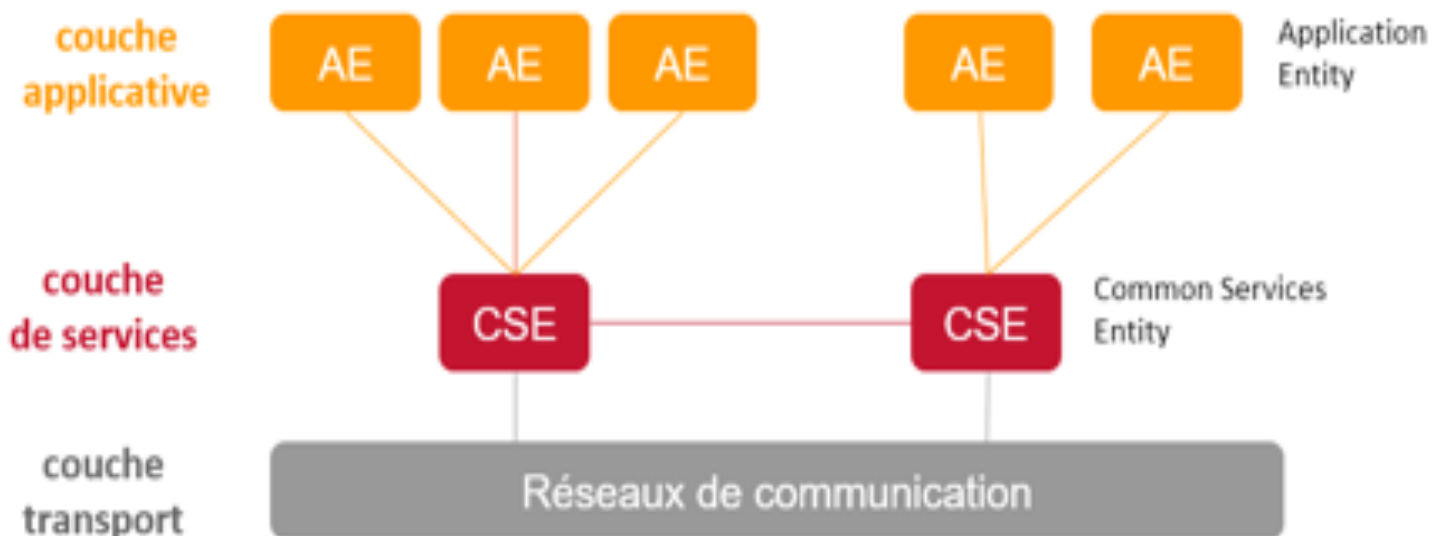


Fig 2 – Architecture proposée par OneM2M (OneM2M MOOC OpenClassrooms – INSA)

Les différents nœuds utilisés pour déployer l'architecture oneM2M :

- Les ADN (Application Dedicated Nodes) représentent les applications déployées sur les dispositifs
- Les MN (Middle Nodes) représentent les passerelles entre les dispositifs et les applications sur les réseaux locaux et sur l'infrastructure OneM2M.
- Les IN (Infrastructure Nodes) sont du côté server de l'architecture, et peuvent être déployés sur le cloud.

Les entités haut niveau, représentées par les AE, s'enregistrent sur un CSE libre du côté server (IN) et passerelle (MN). Ces deux CSE seront capable de communiquer de manière transparente grâce à la couche transport. Voici comment les nœuds interagissent les uns avec les autres.

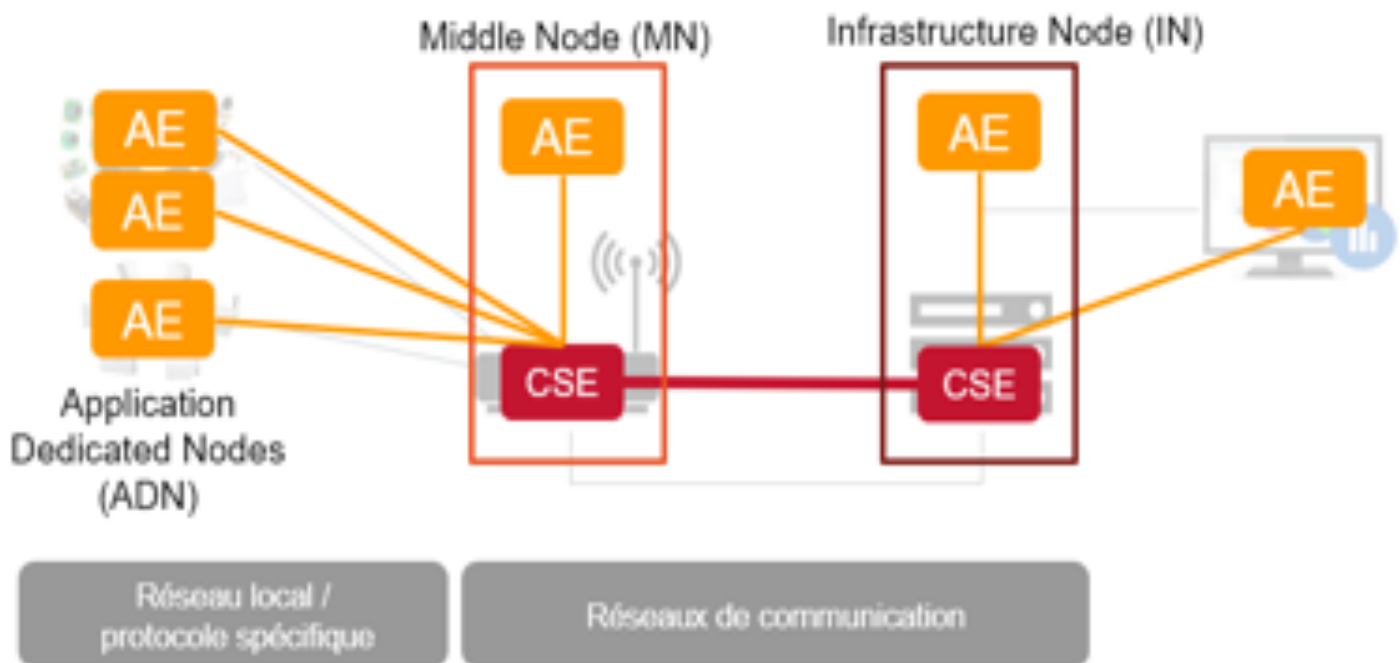


Fig 3 – Architecture proposée par OneM2M (OneM2M MOOC OpenClassrooms – INSA)

Au début des TP, on apprend comment configurer et lancer la plateforme OM2M. On apprend également à gérer le « resource tree » avec l'interface OM2M. Pour déployer une architecture IoT avec OM2M, un server IN-CSE et une passerelle applicative MN-CSE sont nécessaires pour pouvoir communiquer avec les dispositifs. Premièrement, il est nécessaire de configurer le server et la passerelle avec le fichier config.ini et ensuite lancer le server pour enregistrer la passerelle. La passerelle enregistre ses dispositifs et chaque AE possède des entités CNT qui possèdent des mesures (CIN) de l'AE. Grâce à l'interface utilisateur OM2M on peut envoyer des requêtes send/post pour commander les dispositifs.

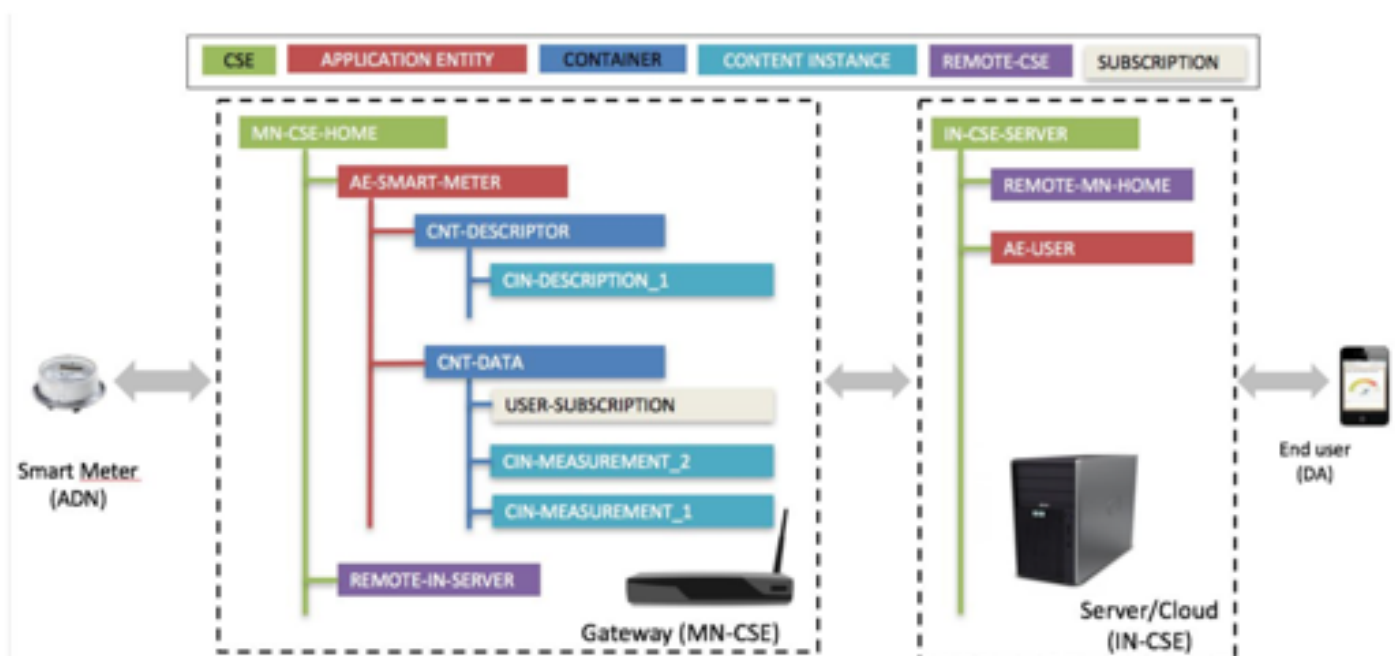


Fig 4 – Resource Tree (TP Machine To Machine – INSA)

2.2. INTERAGIR AVEC LES OBJETS EN UTILISANT UNE ARCHITECTURE REST

*Expliquez comment vous avez pu interagir avec des objets et visualiser les données envoyées par ces derniers (notamment les ressources générées par IPE Sample, lampes simulées). Expliquer les différentes ressources oneM2M que vous avez manipulées (**CSE-BASE**, **AE**, **CNT**, **CIN**, **SUB**, **REMOTE CSE**) et la façon d'interagir avec ces dernières (client HTTP + serveur). (TP 1 + 2)*

Qu'est ce que le CSE-BASE:

Le Common Service Entity Base est considéré comme la ressource parente de toutes les autres ressources. Il rassemble des informations sur le CSE correspondant et peut y stocker de nouvelles ressources.

Pour un CSE-BASE on va retrouver la structure suivante :

- Les AE, que j'ai présenté précédemment, sont des nœuds enfants du CSE-BASE. On peut avoir plusieurs AE associé à un CSE-BASE.
- Les CNT (Containers), structure l'arborescence des données et des ressources. Ils permettent l'échange d'informations entre les applications et les CSE.
- Les CIN (Content), permettent de stocker des données dans la plateforme.

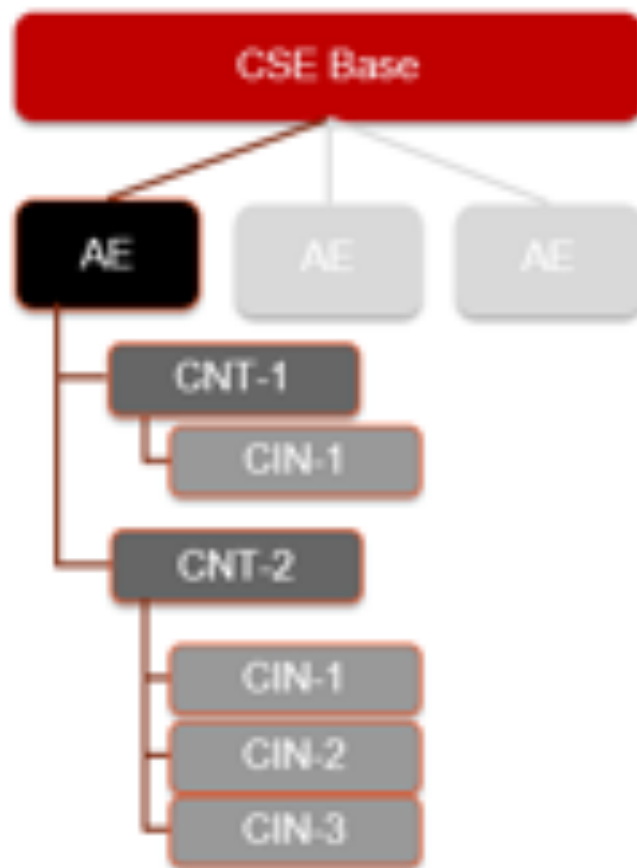


Fig 5 – Arbre de ressource (OneM2M MOOC OpenClassrooms – INSA)

Lors des travaux pratiques, nous avons pu interagir avec des objets et visualiser les données envoyées par ces derniers. Pour contrôler les lampes simulées, nous avons utilisé une architecture similaire à la figure 5. Il s'agit d'une application java lancée sur l'ordinateur de l'utilisateur et qui simule des lampes. L'application peut recevoir des commandes java et retourner l'état des lampes. Sur le CSE-Base, on peut créer des AE correspondant aux lampes. Et déclarer des nouveaux containers de description CNT avec leurs informations et CIN avec la requête http correspondante. On peut alors lancer notre CSE distant pour nous connecter à l'interface web et contrôler les lampes.

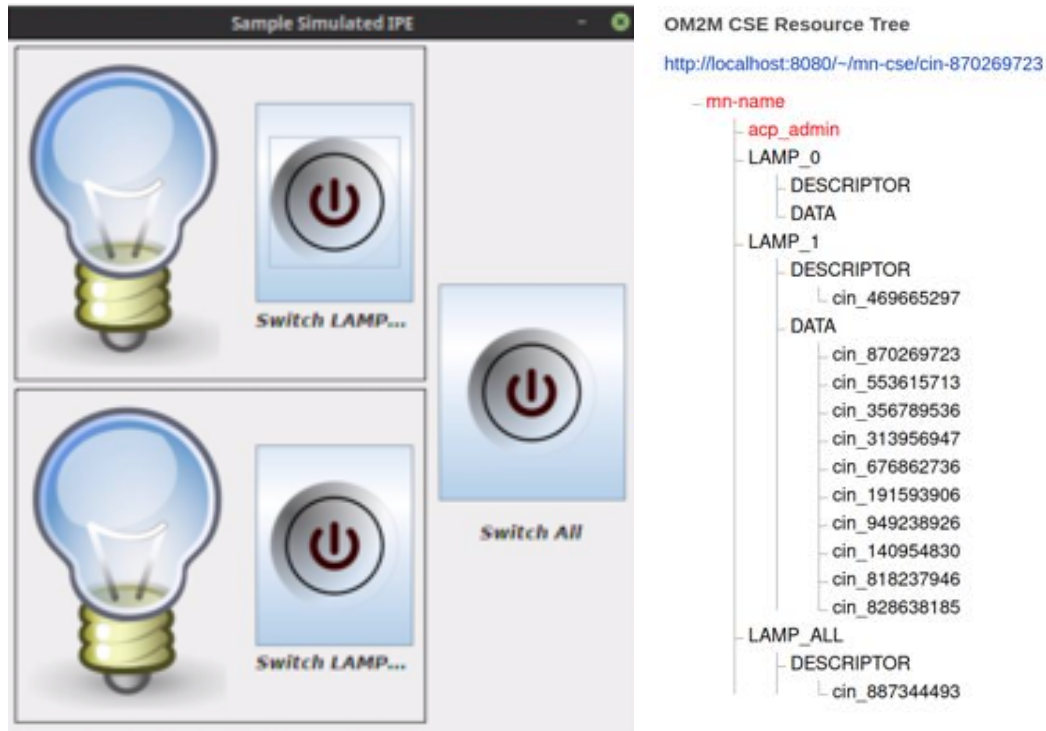


Fig 6 – Lampes simulées

Sur la figure 6, on peut voir à gauche les lampes simulées grâce au GUI. Sur la partie de droite on peut voir l'arbre de ressource associé. Nous avons le CSE-BASE (mn-name). Puis il y a trois AE (LAMP_0, LAMP1, LAMP_ALL). Chaque AE est composé de deux CNT (DESCRIPTOR et DATA). Le CNT DESCRIPTOR contient un CIN avec des informations sur les lampes et fonctions associées (*getState()*, *switchON()*, *switchOFF()*, *toggle()*...). Le CNT DATA contient plusieurs CIN avec des informations sur l'état des lampes.

Pour manipuler les ressources j'ai utilisé une API REST. Et pour facilement envoyer mes requêtes, j'ai utilisé Postman (voir figure 7).

Qu'est ce que REST :

Rest, pour representational state transfer, est apparue ces dernières années comme une alternative à l'architecture SOA. Rest est un style architectural pour le développement d'applications! Il présente un ensemble de conventions et de bonnes pratiques pour le développement de services Web, en se basant sur la notion de ressource. Les services Web Rest se basent principalement sur le protocole HTTP qui fournit les opérations nécessaires pour manipuler des ressources. Peu importe le langage dans lequel le service est développé et le type de plateforme d'exécution, un service Web Rest offre obligatoirement des interfaces basées sur les verbes HTTP qui sont GET, POST, PUT, ou DELETE. Ce sont ces requêtes que j'ai pu utiliser pour manipuler les ressources.

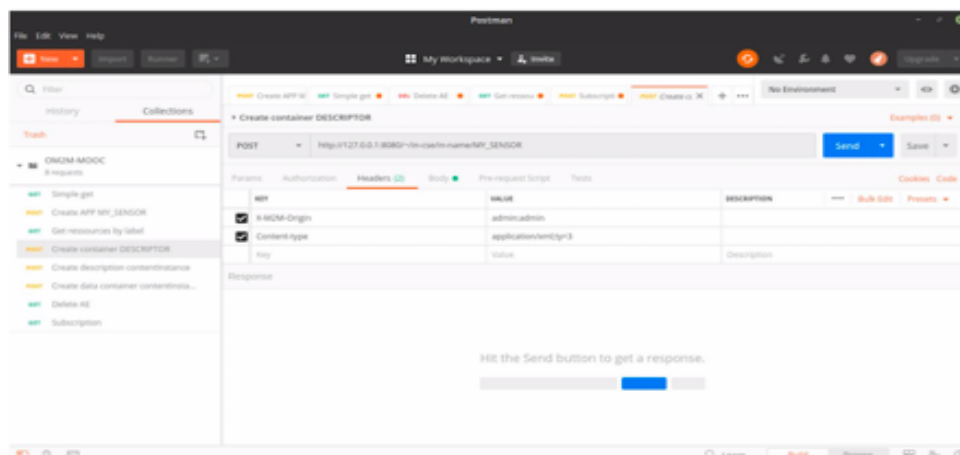


Fig 7 – Postman

2.3. INTEGRER UNE NOUVELLE / AUTRE TECHNOLOGIE DANS UNE ARCHITECTURE IOT

Expliquez le principe d'Interworking Proxy Entity de oneM2M ainsi que l'architecture utilisée pour intégrer dans l'intergiciel une nouvelle technologie non nativement compatible avec le standard. (Architecture logicielle, nœuds oneM2M utilisés, ressources oneM2M utilisées pour faire l'interface.) (TP 3)

OneM2M est un standard qui permet d'intégrer toute application peu importe le protocole de communication. C'est le OneM2M Interworking Proxy Entity qui, en prenant le rôle d'une entité proxy, va permettre d'interconnecter différents appareils et utilisateurs, aux technologies diverses, via un serveur cloud. Il rend transparent l'accès à la technologie intégrée, permettant à une application d'interagir avec elle à travers l'interface standard de OM2M. L'IPE permet également de contourner certaines limitations de la technologie native pour les remplacer par des fonctionnalités standards.

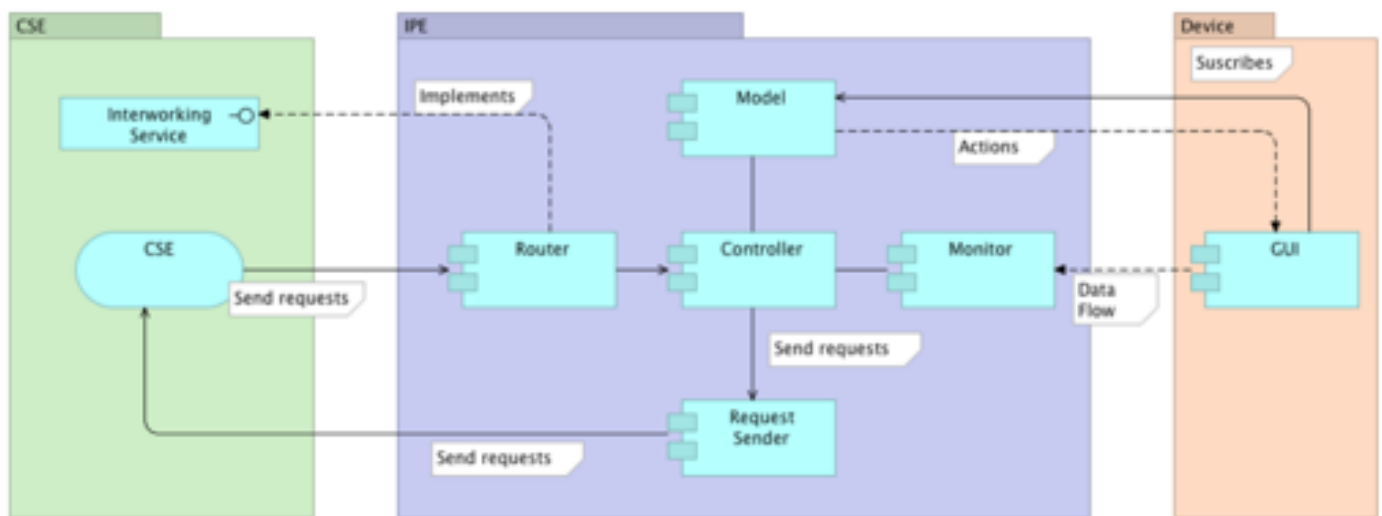


Fig 8 – Interworking Proxy Entity (OneM2M MOOC OpenClassrooms – INSA)

Au démarrage de l'IPE et lorsque de nouveaux capteurs sont ajoutés au réseau, l'IPE devra créer un container dans OM2M. Il devra également créer l'instance correspondante sous le capteur en question, à chaque nouvelle observation.

Finalement, pour implémenter une nouvelle technologie non compatible avec la norme oneM2M, il est nécessaire de créer un nouvel AE avec son conteneur et l'instance de contenu de l'appareil utilisant cette technologie. Nous pourrions ensuite traduire les requêtes http dans la technologie de l'appareil à l'aide des commandes et des données stockées. L'IPE joue le rôle d'intermédiaire entre le serveur et l'appareil, traduisant la requête http en commande compréhensible par l'appareil et inversement.

3. DEPLOYER UNE APPLICATION COMPOSITE ENTRE DIVERS TECHNOLOGIES GRACE A NODE-RED EN SE BASANT SUR UN INTERGICIEL STANDARDISE

Expliquez brièvement le principe de fonctionnement de NODE-RED.

Donnez les exemples d'application(s) que vous avez déployée(s) avec NODE-RED (capture écran du flux et explications). Faites également un export de votre application et joignez le(s) fichier(s) à votre rapport. (TP 4)

Node-Red est un éditeur visuel open-source d'IBM pour l'internet des objets. Il permet de lier aisément des sources de données à des composants de traitement, locaux ou distants, et de créer des chaînes de valeurs en quelques clics. Le TP4 consistait à travailler avec un fibaro z-wave. C'est un multi capteur connecté avec une technologie ZWave. Il était également demandé de travailler avec les lampes HUE de Philips. Cependant nous n'avions pas ce matériel à disposition et avons utilisé des lampes simulées et un « Luxmètre aléatoire » (qui génère aléatoirement un entier compris entre 0 et 400). Sur la figure 9 on peut voir comment l'intégration aurait dû être faite.

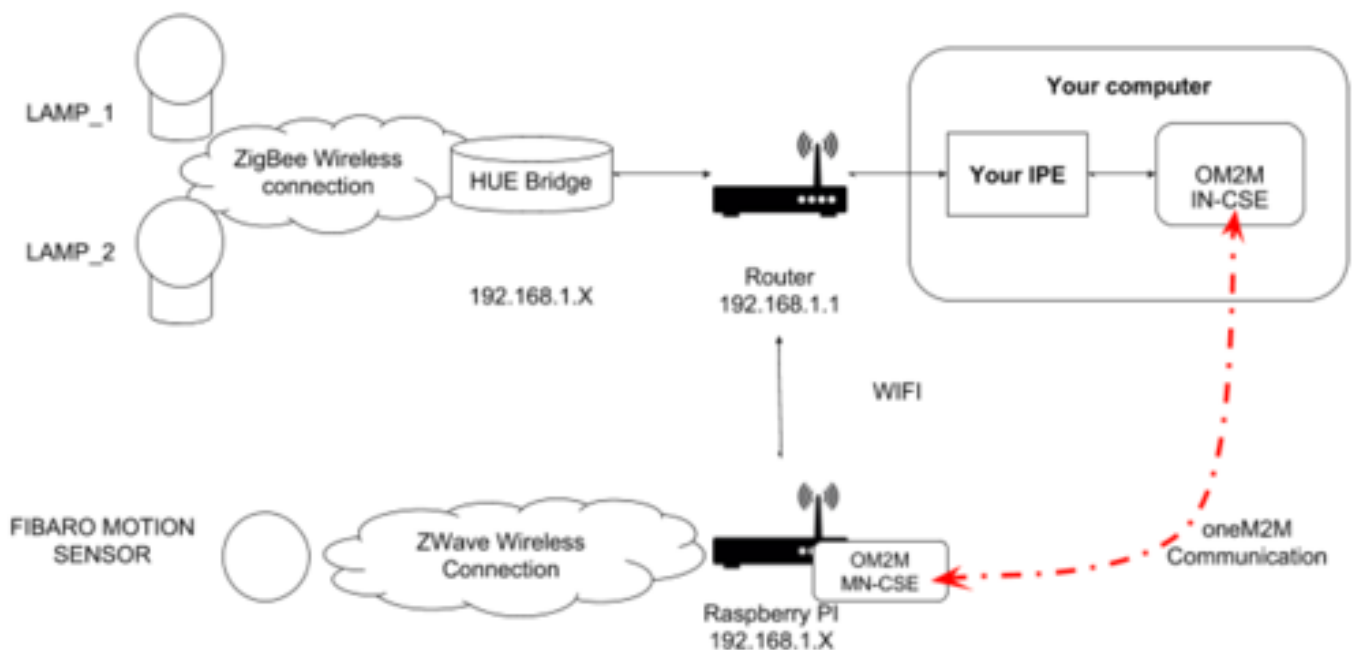


Fig 9 – Intégration du capteur et des lampes (TP Machine To Machine – INSA)

Comme mentionné précédemment, Node-Red nous permet de lier graphiquement différents dispositifs et de les programmer à faire différentes tâches. Pour déclarer un nouveau capteur, il faut spécifier l'URI de l'MN-CSE concerné, pour que Node-Red envoie une requête http au serveur. Il faut également spécifier le nom du capteur perçu par la passerelle applicative, à Node-Red. Quand le programme Node-Red est déployé les opérations sont traduites en requêtes http et envoyées au MN-CSE pour être ensuite traduites dans la technologie du dispositif.

Pour le TP nous avons créé une application qui : lorsque la lampe 0 est allumée et que la luminosité est supérieure à un certain seuil (ici 200 lux) alors la lampe 1 s'éteint. Lorsque la luminosité est inférieure à ce seuil, la lampe 1 s'allume. Pour les lumières nous avons utilisés les lampes simulées. Nous avons pu récupérer les informations des lampes avec l'interface Web et avons remplis ces données dans les actionneurs Node-Red.

Afin de récupérer l'état de la lampe 0, nous avons connecté un data extractor qui renvoie un booléen. Dans notre cas, « NamedSensor » nous permet d'avoir l'état de la lampe 0 via l'API Rest. Pour savoir si le seuil de luminosité a été dépassé, nous appliquons une condition sur la valeur renvoyée par le Luxmètre. Cette condition nous retourne ensuite un booléen. Une fois les deux booléens récupérés nous les plaçons dans un tableau avec le bloc « Join ». Ces booléens sont alors envoyés à un switch qui va renvoyer True si les deux valeurs sont True et False sinon. Le booléen généré va ensuite être récupéré par « NamedActuator » qui nous permet de contrôler l'état de la lampe 1. Vous trouverez ci-dessous le wireframe que nous avons implémenté :

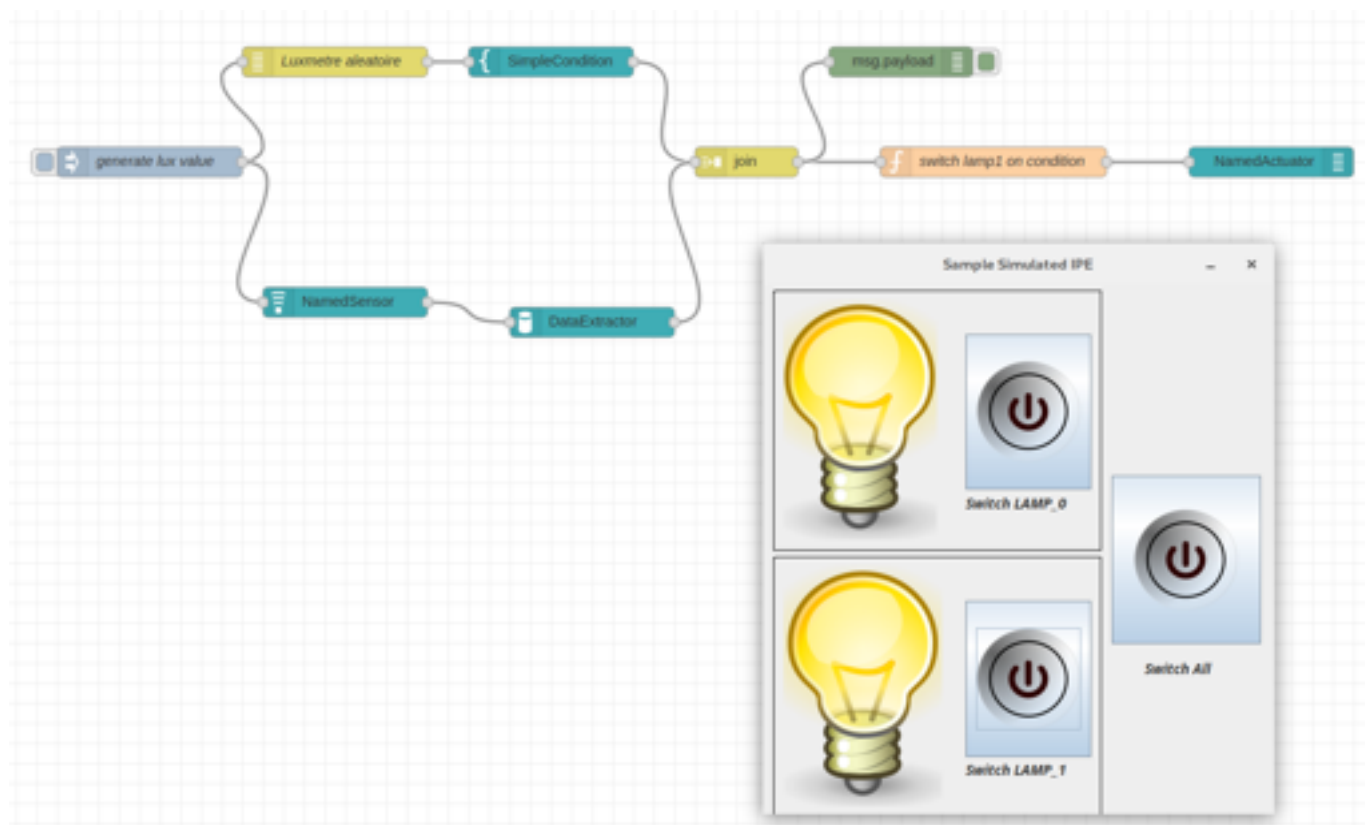


Fig 10 – Node-Red wireframe (lorsque la luminosité <200) (TP Machine To Machine – INSA)

Sur le screenshot précédent, on voit que les deux lumières sont allumées car la luminosité est inférieure au seuil (200 lux). Dans la figure suivante c'est l'inverse. Une seule lumière est allumée, ce qui signifie que la luminosité est supérieure au seuil.

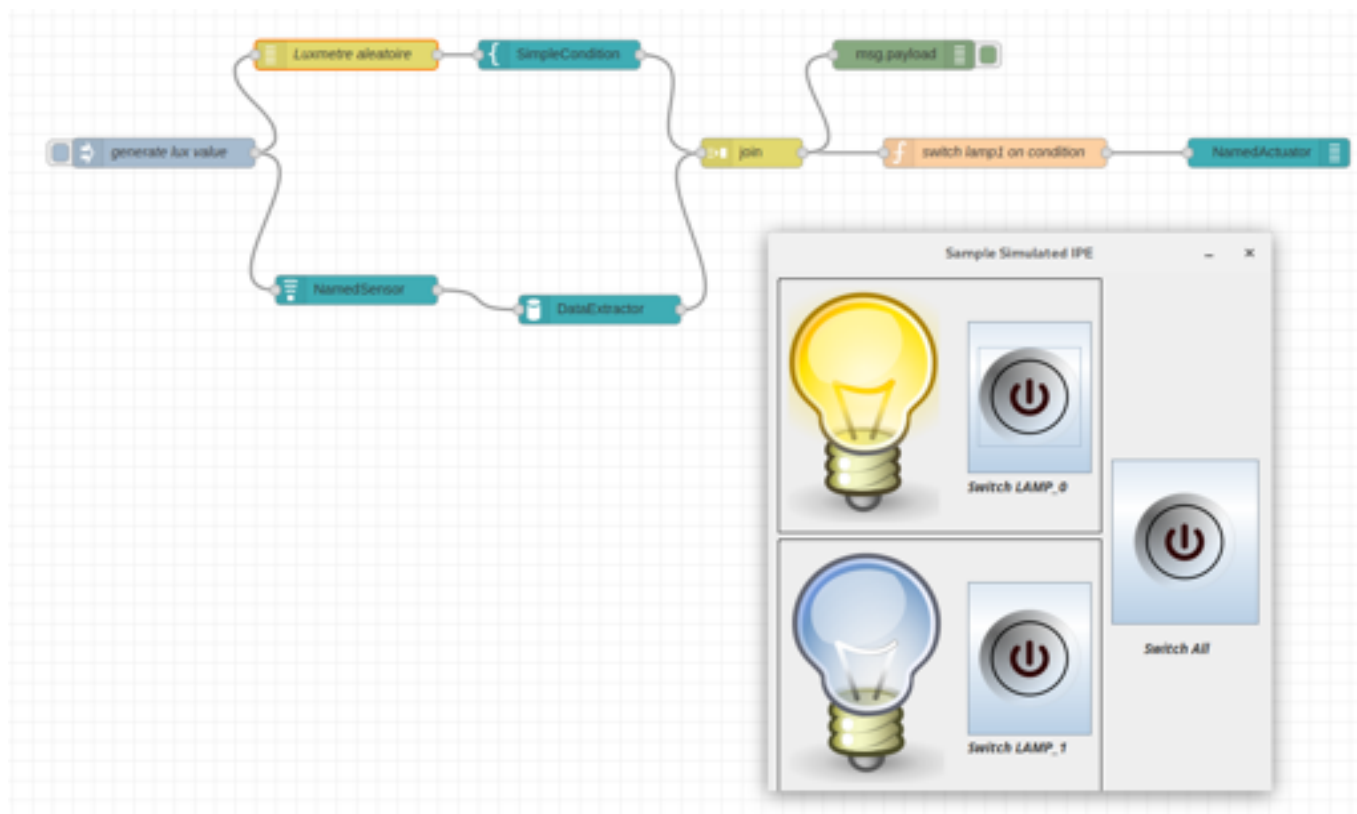


Fig 11 – Node-Red wireframe (lorsque luminosité >200) (TP Machine To Machine – INSA)

Voici l'export de notre flow Node-Red au format JSON :

```
[{"id":"d19c7b48.9804a","type":"tab","label":"OM2M","disabled":false,"info":"","","id":"6d4570c6.928bf","type":"NamedSensor","z":"d19c7b48.9804a","name":"","platform":"f82a51a7.c4c49","sensor":"2546e1df.932b5e","container":"DATA","cntInstance":"/la","x":280,"y":1540,"wires":[["3255f69a.18ed0a"]]}, {"id":"3255f69a.18ed0a","type":"DataExtractor","z":"d19c7b48.9804a","name":"","viewtype":"state","viewunid1":"","viewunid2":"$.bool.val[0]","x":480,"y":1540,"wires":[["ab2328b7.c6d69"]]}, {"id":"84fa11d1.34ec68","type":"random","z":"d19c7b48.9804a","name":"Luxmetre aleatoire","low":0,"high":400,"inte":"true","property":"payload","x":290,"y":1280,"wires":[["adfb111c.bdbf","f9236a2c.deba7"]]}, {"id":"6aa87b76.f3e65c","type":"inject","z":"d19c7b48.9804a","name":"trigger","topic":"","payload":"","payloadType":"date","repeat":"","crontab":"","once":false,"onceDelay":0.1,"x":130,"y":1420,"wires":[["84fa11d1.34ec68","6d4570c6.928bf"]]}, {"id":"adfb111c.bdbf","type":"SimpleCondition","z":"d19c7b48.9804a","operator":"<","value_c":200,"inputType":"msg","input":"payload","name":"","x":500,"y":1400,"wires":[["e9d23692.5c5af"]]}, {"id":"b5d045b.17a89b8","type":"NamedActuator","z":"d19c7b48.9804a","platform":"f82a51a7.c4c49","name":"","actuator":"14965642.de2e7a","command":"op=setOn&lampid=LAMP_1","x":1340,"y":1400,"wires":[]}, {"id":"e9d23692.5c5af","type":"join","z":"d19c7b48.9804a","name":"","mode":"custom","build":"array","property":"payload","propertyType":"msg","key":"topic","joiner":"\\n","joinerType":"str","accumulate":false,"timeout":"","count":2,"reduceRight":false,"reduceExp":"","reduceInit":"","reduceInitType":"","reduceFixup":"","x":770,"y":1440,"wires":[["8b6ba56b.580dc8","f9236a2c.deba7"]]}, {"id":"8b6ba56b.580dc8","type":"function","z":"d19c7b48.9804a","name":"switch lamp1 on condition","func":"var newMsg = { payload: msg.payload };\\nnewMsg.payload = (newMsg.payload[0] === newMsg.payload[1]) && (newMsg.payload[0] === true);\\nreturn newMsg;","outputs":1,"noerr":0,"x":1010,"y":1440,"wires":[["96b27d1c.5332a8","68e3f6ab.04ddc"]]}, {"id":"f9236a2c.deba7","type":"debug","z":"d19c7b48.9804a","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","x":1050,"y":1280,"wires":[]}, {"id":"96b27d1c.5332a8","type":"debug","z":"d19c7b48.9804a","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","x":1310,"y":1260,"wires":[]}, {"id":"ab2328b7.c6d69","type":"change","z":"d19c7b48.9804a","name":"string to bool","rules":[{"t":"change","p":"payload","pt":"msg","from":"true","fromt":"str","to":"true","tot":"bool"}],{"t":"change","p":"payload","pt":"msg","from":"false","fromt":"str","to":"false","tot":"bool"}], "action":"","property":"","from":"","to":"","reg":false,"x":650,"y":1520,"wires":[["e9d23692.5c5af"]]}, {"id":"70e61f77.8b1a28","type":"NamedActuator","z":"d19c7b48.9804a","platform":"f82a51a7.c4c49","name":"","actuator":"14965642.de2e7a","command":"op=setOff&lampid=LAMP_1","x":1340,"y":1500,"wires":[]}, {"id":"68e3f6ab.04ddc","type":"switch","z":"d19c7b48.9804a","name":"","property":"payload","propertyType":"msg","rules":[{"t":"eq","v":"true","vt":"jsonata"}, {"t":"eq","v":"false","vt":"jsonata"}], "checkall":"true","repair":false,"outputs":2,"x":1130,"y":1500,"wires":[["b5d045b.17a89b8"],["70e61f77.8b1a28"]]}, {"id":"f82a51a7.c4c49","type":"xN_CSE","z":"","platform":"OM2M_1","URLBase":"http://127.0.0.1:8080/~mn-cse/mn-name","user":"admin","password":"admin"}, {"id":"2546e1df.932b5e","type":"AE","z":"","appId":"LAMP_0"}, {"id":"14965642.de2e7a","type":"AE","z":"","appId":"LAMP_1"}]
```