

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 : **TRAORÉ**

Prénom 1 : **Fayçal Samir Zégué**

Nom 2 :

Prénom 2 :

Groupe : A1

Encadrant :

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

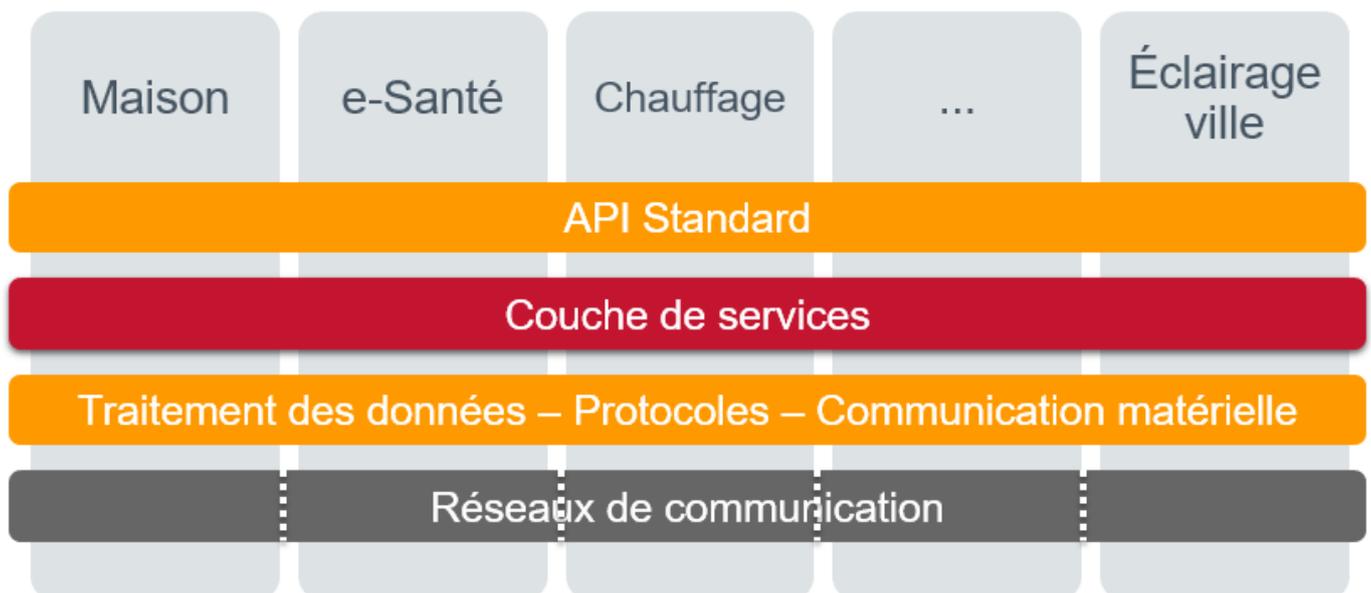
Expliquez brièvement le principe du standard **oneM2M** et comment il se positionne vis à vis des autres standards et technologies déjà existantes.

L'Internet des Objets (IOT) est un ensemble de réseaux physiques et virtuels qui communique souvent via des réseaux virtuels. C'est un domaine qui prend de plus en plus d'ampleur ces dernières années, notamment par le fait qu'il permet de lier monde réel et technologies, avec des capteurs permettant de donner en temps réel des informations sur le domaine physique et même de réagir directement face à cela.

Face à cette grande panoplie de capteurs et d'informations qui circulent, le besoin s'est présenté de mettre en place de façon standardisée un moyen générique de communication entre ces différents objets. Plusieurs standards ont vu le jour notamment, parmi lesquels :

- MQTT qui permet la Gestion et stockage de données échangées entre les objets via un système de publish subscribe
- CoAP qui est un protocole de communication pour les objets et équipements contraints basé sur un modèle de client serveur
- oneM2M, le standard créé par le consortium oneM2M composé de différents organismes de standardisation de par le monde (ETSI, ATSI, TIA, etc)

OneM2M propose une vision assez générique et horizontale des objets connectés, et fait de sorte que quel que soit le domaine dans lequel il est appliqué, il puisse être utilisé de la même façon, de façon transverse. Que ce soit dans le domaine des bâtiments et maisons connectées, dans l'industrie, l'énergie ou la santé il permet d'avoir une vision homogène du système grâce à une couche de services accessible en à travers une API REST et un arbre de ressources.



2. DÉPLOYER UNE ARCHITECTURE CONFORME À UN STANDARD ET METTRE EN PLACE UN SYSTÈME DE RÉSEAU DE CAPTEURS AUX SERVICES

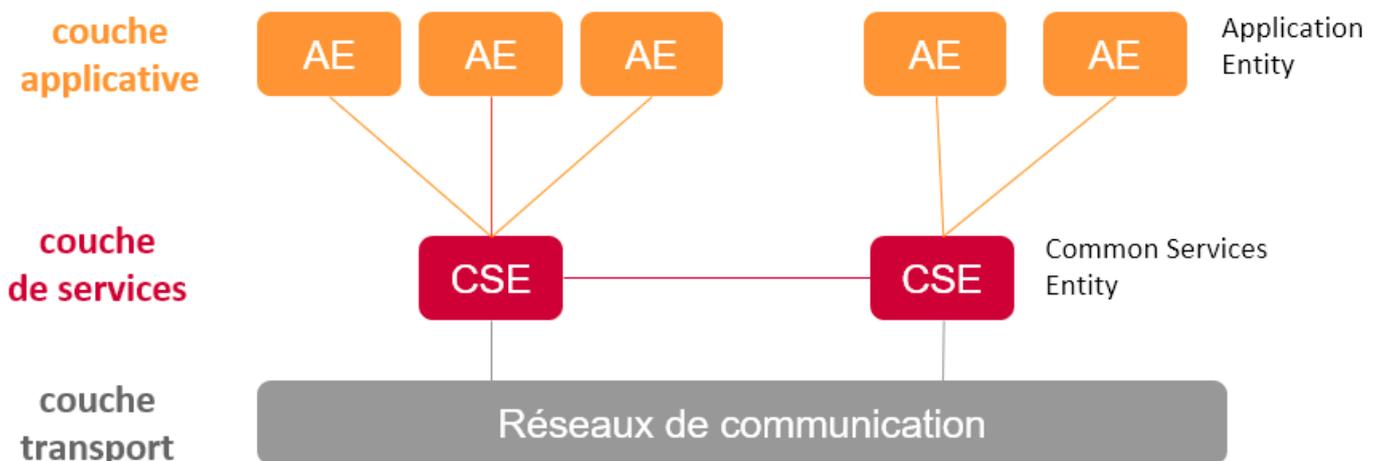
OM2M est le logiciel Eclipse dédié au déploiement et à la mise en œuvre de la norme oneM2M. C'est un logiciel open source développé initialement au LAAS-CNRS. Nous allons maintenant décrire l'architecture IoT du standard oneM2M en détails, et comment le déployer grâce à OM2M.

2.1. DÉPLOYER 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)

OneM2M possède une architecture composée de plusieurs entités regroupées principalement sous 3 couches :

- Une couche transport qui se charge de la communication réseaux, à savoir le transport physique des données. Une abstraction de cette couche est faite par les différentes couches de services
- Une couche de services ou Common Entity Layer (CSE). Ces entités fournissent les différents services liés au standard en permettant aux différentes applications de la couche applicative de s'enregistrer
- Une couche applicative ou Applicative Entities (AE) qui elle permet de représenter les différentes applications connectées au système. Cette couche est celle avec laquelle nous interagissons le plus car c'est au niveau de celle-ci que nous allons retrouver justement les différentes applications connectées, notamment les applications de l'utilisateur (monitoring etc.) et les objets connectés eux-mêmes.

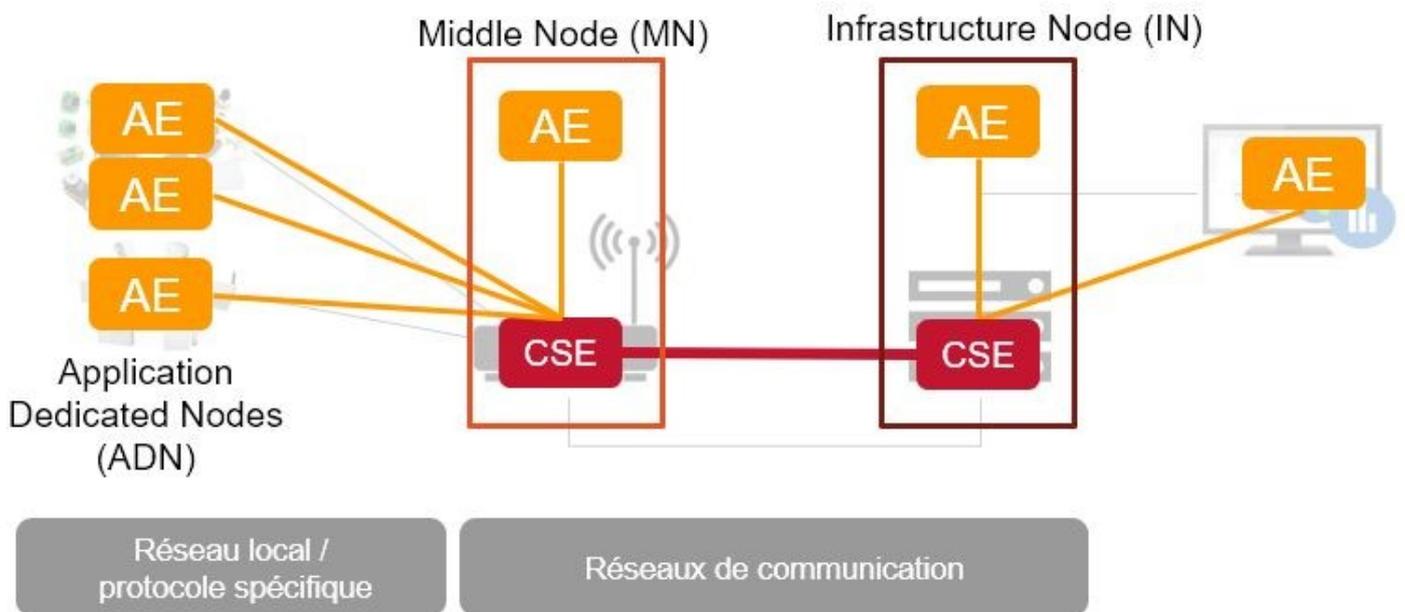


Structure en couche de oneM2M

Pour le déploiement d'une architecture oneM2M plusieurs types de nœuds entrent en compte.

Ces types de nœuds sont :

- L'**Application Dedicated Node (ADN)**: qui représentent les applications s'exécutant sur les objets
- Le **Middle Node (MN)** représentant la passerelle entre les objets et les applications s'exécutant en local sur l'infrastructure oneM2M
- L'**Infrastructure Node (IN)** qui représente le serveur s'exécutant sur le cloud



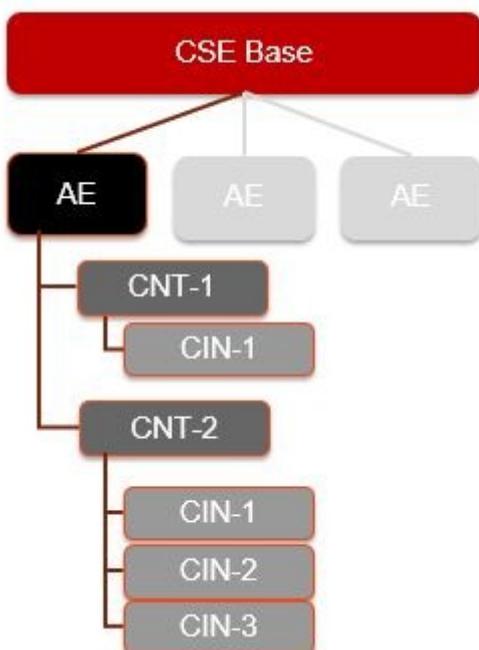
Architecture oneM2M

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)

Comme nous l'avons expliqué au niveau de la première partie, oneM2M est un standard permettant d'avoir une vision homogène du système grâce à une couche de services accessible en à travers une API REST et un arbre de ressources.

Cet arbre de ressource se présente de la façon suivante.



LE CSE BASE

Le CSE Base (Common Service Entity de Base) est la ressource la plus importante car c'est la ressource racine. Elle représente le début de l'arborescence et permet de regrouper différentes informations sur le CSE pris en compte, et de rajouter de nouvelles ressources sous ce dernier. Cela permet donc d'avoir une structure hiérarchisée avec comme ressource parent le CSE de base et comme ressources filles toute les autres ressources en dessous de celle-ci.

LES APPLICATIONS ENTITIES (AE)

Il y a plusieurs AE sous le même CSE. Les AE représentent le lien logique entre une application et un CSE. Elles permettent notamment d'enregistrer une application distante ou locale à la couche de services.

LES CNT

Ils jouent le rôle de conteneurs de données et permettent de structurer l'arborescence de données et de ressources.

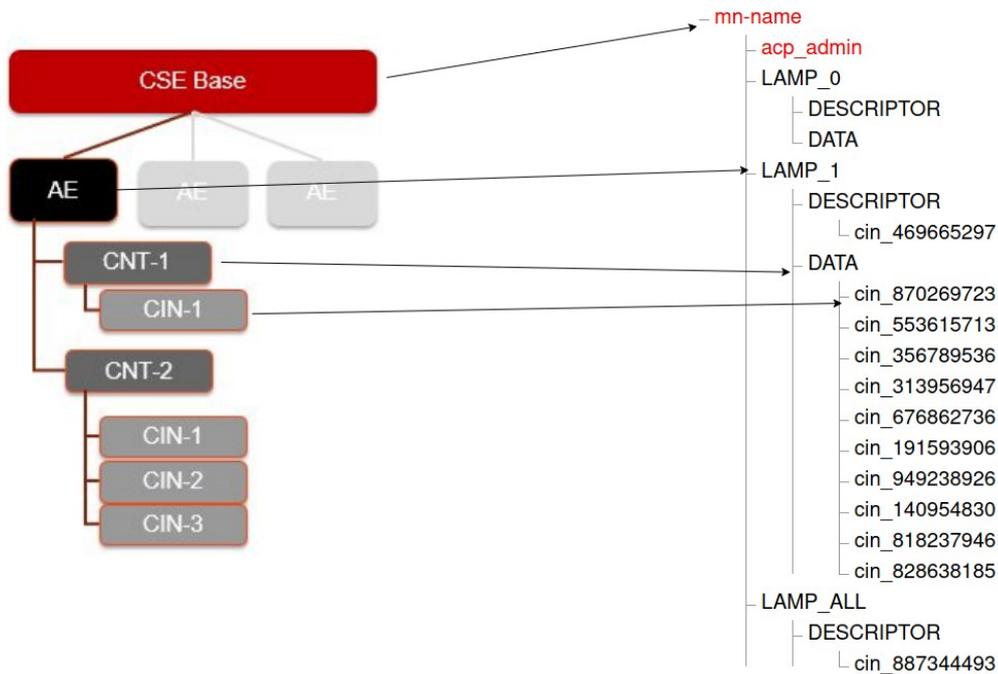
LES CIN

Ce sont à proprement parler les ressources qui permettent de stocker les valeurs sur la plateforme.

Pour faire une correspondance entre le schéma fait ci-dessus et le TP fait en utilisant la simulation de lampe IPE offerte par OM2M nous avons le schéma de correspondance ci-dessous:

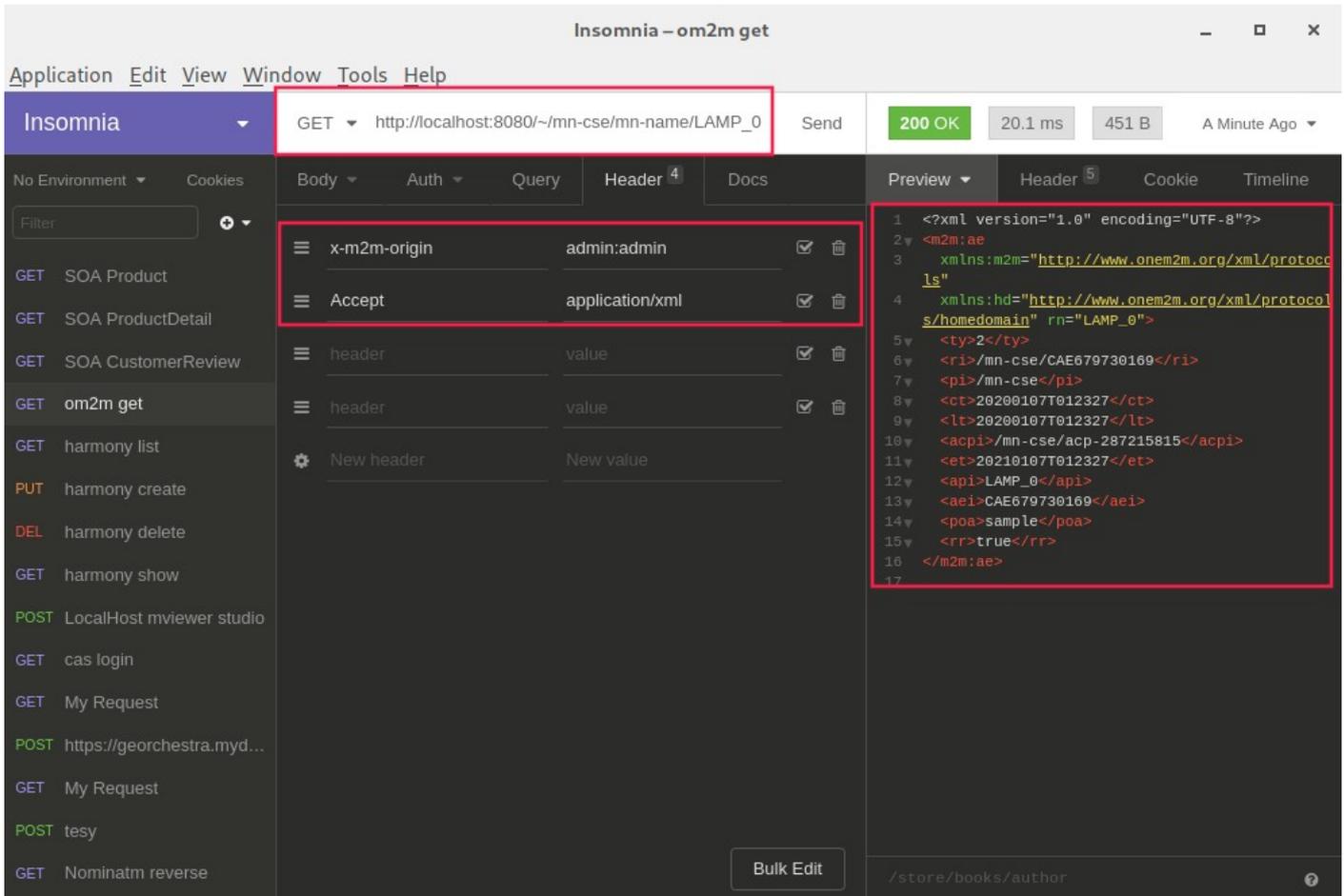
OM2M CSE Resource Tree

<http://localhost:8080/~mn-cse/cin-870269723>



Pour interagir avec chacune de ces ressources nous utilisons une API REST. REST permet d'accéder aux ressources à travers des URL que nous interrogeons avec les requêtes HTTP dont les plus utilisées sont GET, POST, PUT, DELETE. Et pour faire ces différentes requêtes nous avons besoin d'un client HTTP, c'est donc Insomnia (qui ressemble à Postman) que nous avons utilisé pour cela.

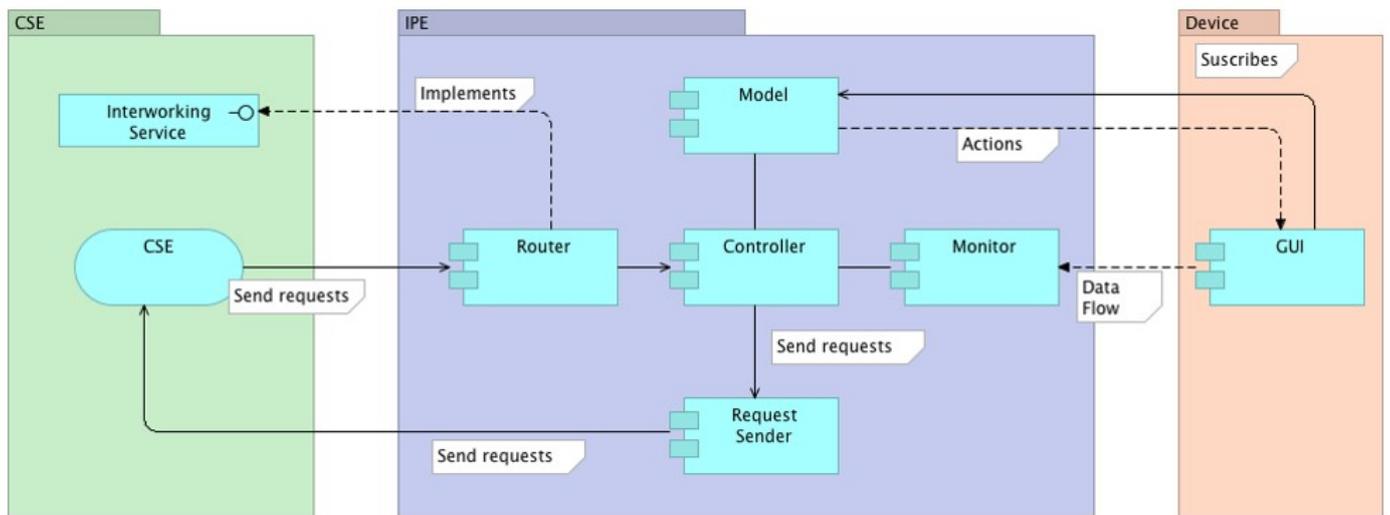
Au niveau de la requête faites ci-dessous nous devons tout d'abord rajouter les headers spécifiques à OM2M, et faire une requête de type GET sur l'URL http://localhost:8080/~mn-cse/mn-name/LAMP_0 pour avoir une réponse concernant les informations contenue sur le CSE à propos de l'AE LAMP_0.



2.3. INTÉGRER 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'integiciel une nouvelle technologie non nativement compatible avec le standard. (Architecture logicielle, nœuds oneM2M utilisés, ressources oneM2M utilisées pour faire l'interface.) (TP 3)

L'objectif de l'Interworking Proxy Entity(IPE) de oneM2M est de créer une interface à partir d'une technologie de périphérique / réseau spécifique vers la norme oneM2M. Dans OM2M, un IPE peut être intégré à la plateforme en en faisant un plugin.



À son démarrage, et à chaque fois que de nouveaux capteurs sont ajoutés au réseau, l'IPE devra créer un container dans OM2M. De plus, à chaque nouvelle observation, l'IPE devra créer l'instance correspondante sous le capteur idoine.

L'intérêt de l'IPE est qu'il rend transparent l'accès à la technologie intégrée, permettant à une application d'interagir avec elle à travers l'interface standard d'OM2M.

L'IPE permet aussi éventuellement de contourner certaines limitations de la technologie native pour les remplacer par des fonctionnalités standards. Si on suppose que les capteurs intégrés ne sont pas capables de notifications, l'IPE pourra implémenter une boucle d'attente active qui créera un flux d'observation dans la plateforme OM2M. De ce fait, au lieu de devoir implémenter la boucle d'attente active elle-même, l'application cliente peut créer une souscription de manière standard et recevra des notifications à chaque nouvelle observation des capteurs.

3. DÉPLOYER UNE APPLICATION COMPOSITE ENTRE DIVERS TECHNOLOGIES GRÂCE À NODE-RED EN SE BASANT SUR UN INTERGICIEL STANDARDISÉ

*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 outil de programmation pour connecter des périphériques matériels, des API et des services en ligne de manière nouvelle et intéressante. Il fournit un éditeur basé sur un navigateur qui facilite le câblage des flux à l'aide de la large gamme de nœuds de la palette qui peut être déployée sur son runtime en un seul clic.

Nous avons donc utilisé NODE-RED pour mettre en place notre application.

L'application mise en place permet d'allumer la lampe 1 de IPE sample 39 de OM2M lorsque le luxmètre renvoie une valeur en deçà de 200 et que la lampe 0 est allumée.

Pour cela nous avons connecté un data extractor à une notification reliée à la lampe 0 permettant d'extraire la valeur reçue par le luxmètre et d'appliquer une condition sur la valeur (< 200) renvoyant true ou false. Au même moment nous récupérons l'état de la lampe 0 via un NameSensor puis un DataExtractor permettant de récupérer la valeur booléenne que l'on converti ensuite en bool.

Ces deux valeurs sont ensuite récupérées dans tableau sous la forme `[bool1, bool2]` que l'on passe à une fonction renvoyant true si les deux valeur sont true, false sinon. A travers un switch on allume ou éteint la lampe 1 via des NamedActuator permettant de contrôler via l'API REST l'état de la lampes

Sur cette première capture d'écran on peut voir que une valeur de 39 lux (qui est inférieure à 200) et la lampe 0 allumée, ce qui entraîne le renvoie d'une valeur true par notre fonction allumant ainsi donc la lampe 1.

The screenshot shows the Node-RED interface with a flow diagram and a debug console. The flow diagram includes the following nodes: NotificationsHandler, NamedSensor, DataExtractor, SimpleCondition, join, string to bool, switch lamp1 on condition, switch, and NamedActuator. The debug console shows a log entry for 19/01/2020 at 15:32:56 with a payload of `[true, true]`, indicating that both conditions were met and the lamps were turned on.

Sur cette capture d'écran par contre une valeur de 371 lux est renvoyée; ce qui entraîne l'éteignage de la lampe 1.


```
rator": "<", "value_c": "200", "inputType": "msg", "input": "payload", "name": "",
, "x": 500, "y": 1400, "wires": [[ "e9d23692.5c5af" ]]},
{"id": "b5d045b.17a89b8", "type": "NamedActuator", "z": "d19c7b48.9804a", "pla
tform": "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": "", "mod
e": "custom", "build": "array", "property": "payload", "propertyType": "msg", "k
ey": "topic", "joiner": "\\
n", "joinerType": "str", "accumulate": false, "timeout": "", "count": "2", "reduc
eRight": false, "reduceExp": "", "reduceInit": "", "reduceInitType": "", "reduce
Fixup": "", "x": 770, "y": 1440, "wires":
[[ "8b6ba56b.580dc8", "f9236a2c.deba7" ]]},
{"id": "8b6ba56b.580dc8", "type": "function", "z": "d19c7b48.9804a", "name": "s
witch 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": "", "ac
tive": 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": "", "a
ctive": 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": "stri
ng 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", "pla
tform": "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": "", "p
roperty": "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"}]
```