Intergiciel pour l'Internet des Objets

Rapport de TP – 5 ISS INSA Toulouse

Nom 1 : **Laurens**

Prénom 1 : **Pierre**

Groupe : **B2**

Nom 2 : **Ferrandi**

Prénom 2 : **Emmanuel**

Tuteur : **Guillaume Garzone**

Dans ce rapport, il vous sera présenté l'explication et la fonctionnement de l'architecture IoT OM2M créée par le LAAS-CNRS. Elle est basée sur le protocole OneM2M et permet de piloter et de monitorer tout type d'objet connectés afin de créer des environnements autonomes ou pilotables à distance. Node-red, une service web, permet quand à elle de faire le lien entre l'utilisateur et les objets connectés de façon simplifié avec une interface user-friendly.

Savoir positionner les standards principaux de l'Internet des Objets

Le concept de machine à machine (M2M) est l'une des principales caractéristiques d'Internet des objets (IoT). Il permet d'interconnecter des milliards d'appareils dans un avenir proche couvrant différents domaines. Cependant, le M2M souffre d'une fragmentation verticale importante et d'un manque de normes. Pour combler cette lacune, l'European Telecommunications Standards Institute (ETSI) a publié un ensemble de spécifications pour une plate-forme de service M2M commune. C'est la raison pour laquelle le projet Open Source OM2M, une plate-forme de service M2M autonome conforme à la norme ETSI a été créé. OM2M fournit une API RESTful pour améliorer l'interopérabilité. Il propose une architecture modulaire s'appuyant sur une couche OSGi, la rendant hautement extensible via des plugins. Il permet la liaison de plusieurs protocoles de communication, la réutilisation de mécanismes de gestion de périphériques distants existants et l'interfonctionnement avec des périphériques existants. Pour résoudre le problème de la complexité du M2M, un nouveau service M2M basé sur le paradigme de l'informatique autonome et des modèles sémantiques est défini pour fournir des mécanismes de découverte et de reconfiguration dynamiques.

Déployer une architecture conforme à un standard et mettre en place un système de réseau de capteurs aux services

Déployer et configurer une architecture IoT en utilisant OM2M

Dans un premier temps, afin d'utiliser OM2M, il est nécessaire d'installer un package disponible sur ce lien : https://www.eclipse.org/om2m/. Grâce à ça vous allez pouvoir créer une architecture en local avec laquelle vous allez pouvoir interagir. Le standard oneM2M se compose de quatre entités un ou plusieurs CNT (container), un ou plusieurs AEs (Application Entity), un ou plusieurs MN-CSE (Middle Node Common Services Entity) et un IN-CSE (Infrastructure Node).

Voici un exemple d'implémentation qui peut exister si l'on utilise ce standard dans le cadre d'une salle connectée.

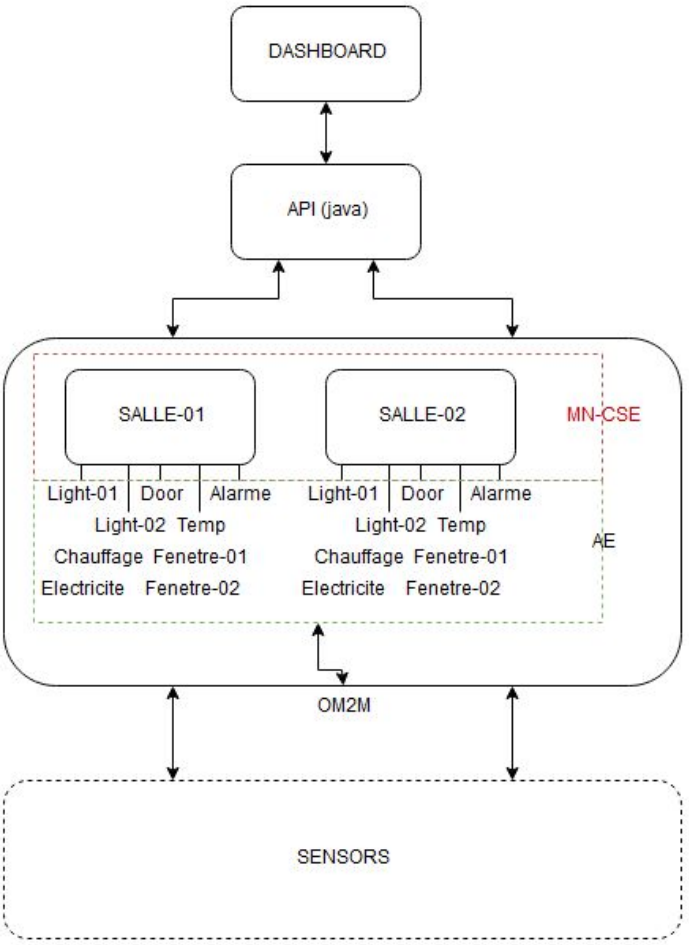


Image 1 : *Représentation d'une architecture basée sur le standard oneM2M*

Ainsi, il est possible de voir qu'une salle est gérée par un MN-CSE qui correspond à une gateway (Raspberry par exemple) avec en dessous différent capteurs qui sont tous des AEs. Grâce à l'application OM2M, il est possible d'interagir avec ses différents objets en interrogeant juste l'adresse souhaitée correspondante. Pour se faire, il est possible d'envoyer un fichier XML à l'aide du protocole http.

Il est possible d'avoir plusieurs types d'interactions avec les objets comme la création d'AE, la récupération de donnée, la mise à jour de donnée, la suppression de donnée ou d'AE ou encore souscrire à un changement de donnée se passant dans un AE. Cette partie est expliquée plus en détails par la suite. Vous pouvez donc créer une application dans le langage de programmation de votre choix pour réaliser une architecture M2M en utilisant OM2M.

Interagir avec les objets en utilisant une architecture REST

Comme expliqué un peu plus haut, oneM2M est composé d'une hiérarchie permettant la création d'entité ainsi que de leur gestion plus souple.

Nous avons donc :

1. CSE (Common Service Entity) : correspond au sommet de l'arborescence qui permet de stocker les nouvelles ressources telles que les AEs
2. AE (Application Entity) : permet d'enregistrer une application distante ou locale au CSE (capteurs, actionneurs ...)
3. CNT (Container) : permet de structurer l'arborescence par exemple pour un capteur, on peut stocker les données géographiques mais aussi les données qu'il capture dans deux CNT différents
4. CIN (Content Instance) : permet de stocker la donnée en lien avec le CNT

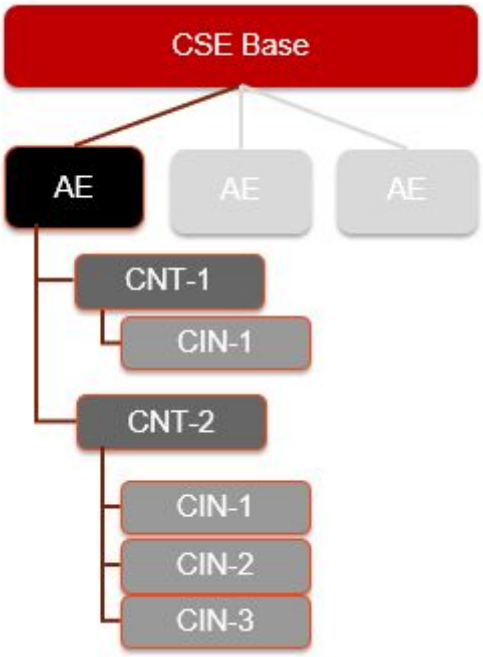


Image 2 : *Les ressources de base de oneM2M*

Cette hiérarchie est basée sur un service Web Rest avec laquelle on peut communiquer en utilisant le protocole http.

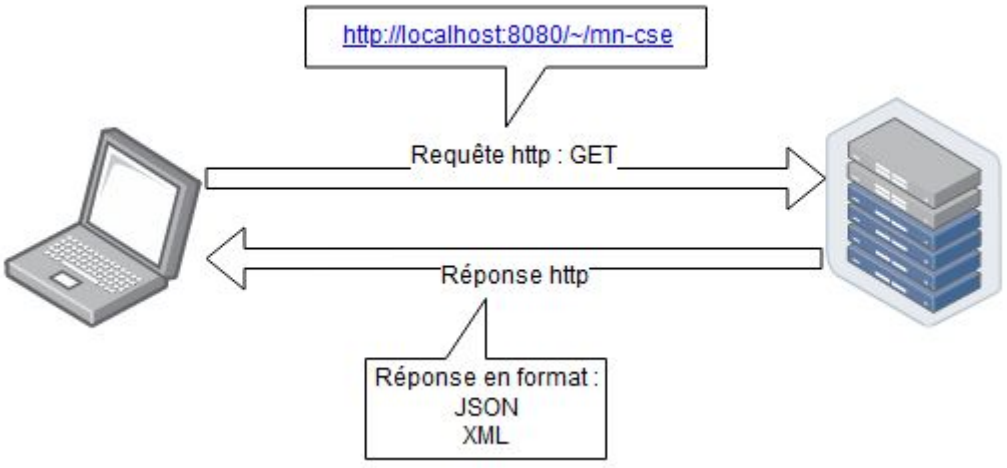


Image 3 : *Echange avec le service Web REST*

Dans le TP 1 et 2, nous avons pu mettre en place ce service et communiquer avec en utilisant un client REST Postman mais aussi en créant notre propre version de client. On a pu créer des AE avec leurs containers et leurs Contents. Ainsi que de faire un abonnement où un serveur est à l'écoute d'un changement faire sur l'AE pour informer après le client / serveur qui l'a demandé. Cette fonctionnalité peut être utilisée si un AE est une alarme et lors de son déclenchement le serveur reçoit cette notification pour ensuite prévenir l'utilisateur.

Intégrer une nouvelle / autre technologie dans une architecture IoT

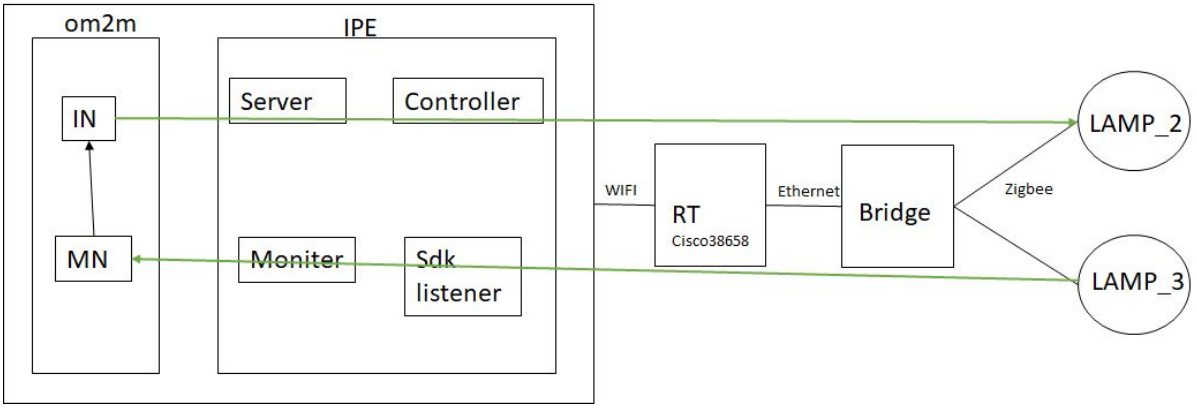


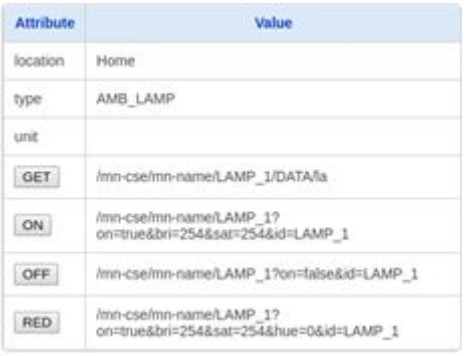
Image 4 : *Architecture IoT avec des lamps Philips Hue*

Comme le montre le schéma ci-dessus **(image 4)**, nous avons déployé une "Interworking Proxy Entity" (IPE) qui permet à n'importe quel objet (ici: la lampe HUE de philips) de dialoguer, interagir avec notre architecture OM2M.

Pour permettre cet interopérabilité et l'utilisation d'objets tel que les lampes HUE, nous avons codé en java un module IPE contenant 4 classes **(décrites ci-dessus sur l'image 4)** qui sont:

- le model (SDK listener) qui écoute et interagit directement avec le dispositif (HUE).
- le monitor qui récupère les données du dispositif et les envoie vers le MN-CSE.
- le contrôleur qui appelle les classes du serveur et exécute toutes les commandes envoyés par le middleware aux lampes.
- le serveur qui permet d'appeler et gérer toutes les classes pour les modules définis avant.

La figure **(image 4)** ci-dessus montre le sens de traitement de l'information.



Attribute	Value
location	Home
type	AMB_LAMP
unit	
GET	/mn-cse/mn-name/LAMP_1/DATA/la
ON	/mn-cse/mn-name/LAMP_1?on=true&bri=254&sat=254&id=LAMP_1
OFF	/mn-cse/mn-name/LAMP_1?on=false&id=LAMP_1
RED	/mn-cse/mn-name/LAMP_1?on=true&bri=254&sat=254&hue=0&id=LAMP_1

Image 5 : *Interface présente sur OM2M en lien avec une lampe Philips Hue*

L'**image 5** ci-dessus a été prise sur l'interface Om2m au niveau de l'objet lampe sur le mn-cse dans le "Application Entity" qui correspondait à l'objet "LAMP_2" ou "LAMP_3" sur le schéma ci-dessus.

Chacun possède 2 containers (CNT) un pour la description de la lampe qui se nomme "DESCRIPTOR" et l'autre pour toutes les données engendrées portant sur l'état de la lampe qui se nomme "DATA.

Dans le container "DESCRIPTOR" un content instance contient 4 fonctions(boutons) qui ont été codées dans le modèle:

- "GET" permet de récupérer l'état de la lampe,
- "ON" permet d'allumer la lampe,
- "OFF" permet d'éteindre la lampe,
- "RED" permet de changer la couleur de la lampe en rouge.

Ces boutons permettent de réaliser une action depuis om2m sur la lampe à distance.

Nous sommes maintenant capable de réaliser des applications comme par exemple piloter une lampe à distance en fonction d'un capteur de température. **(Conf ci-dessous TP4)**

Déployer une application composite entre divers technologies grâce à NODE-RED en se basant sur un intergiciel standardisé

Durant ce TP, nous nous sommes servis du logiciel NODE-RED.

NODE-RED est un outil de développement basé sur le flux permettant une programmation visuelle. Il permet de connecter plusieurs matériels, des APIs et des services en ligne. Il fournit un éditeur basé sur un navigateur. Il propose une large gamme de nœuds dans la "palette" qui peut être déployé et exécuté en un seul clic. Il propose également une interface

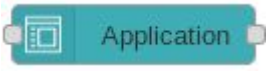
internet ce qui le rend facile d'accès. Il peut être lancé en local, sur le cloud et sur des devices (Raspberry,...).

Voici une description exhaustive des noeuds que nous avons utilisé pour déployer nos deux applications décrites ci-dessous.

Ce noeud est le premier de la chaîne car il permet d'insérer un timestamp dans notre application.

Ce noeud permet d'afficher sur une console le comportement de notre système.

Tous les noeuds suivant appartiennent au standard OM2M.

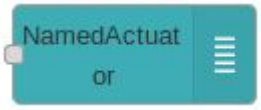


Permet de créer un "Application Entity"

Permet de créer un container (CNT).

Permet de créer un Content Instance (CIN) qui contient la donnée.

Permet de récupérer la valeur d'un capteur en y accédant grâce à un label connu.



Permet d'accéder un capteur dont son nom ou URI est connu.

Permet de simuler des conditions.

Utilisé dans un modèle "Publisher/Subscriber" , permet de créer une entité moniteur dont le rôle est d'envoyer des notifications aux noeuds qui souscrivent à une ressource, dès qu'une nouvelle valeur est disponible. Dans ce cas, les noeuds souscripteurs n'écoutent pas les canaux de communication et, par conséquent, consomment moins d'énergie.

Le schéma **(image 6)** est celui du TP3 auquel nous avons ajouté des briques comme le NODE-RED et le raspberry qui permet de récupérer les données du capteur de température (Fibaro).

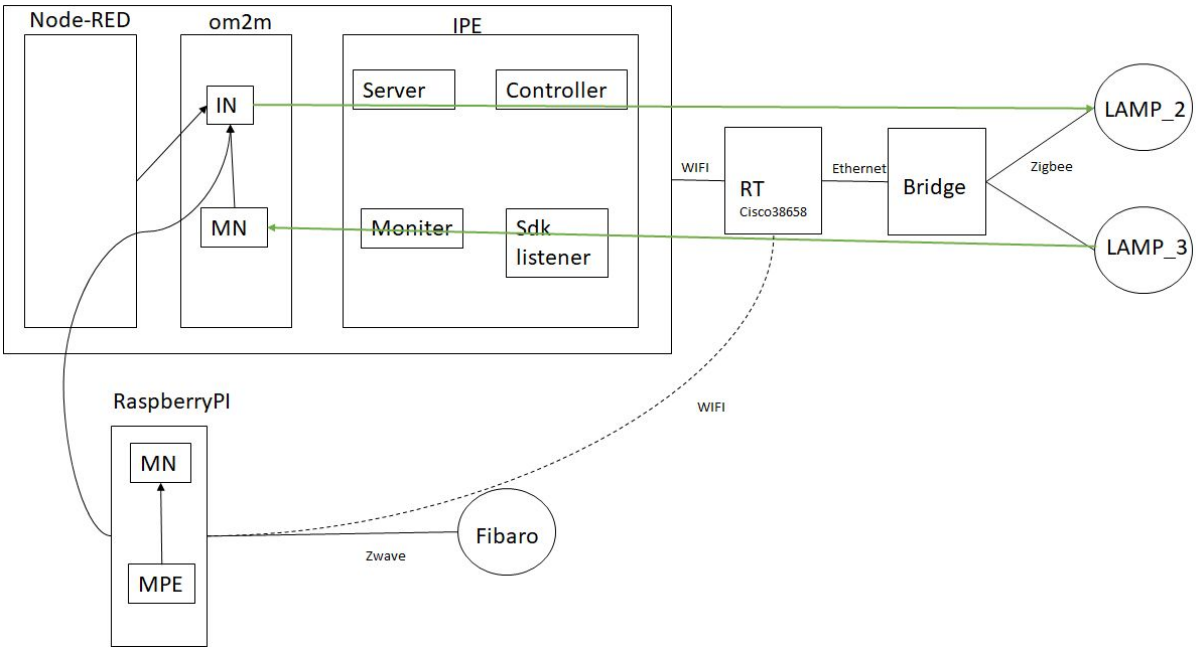


Image 6 : Architecture IoT dans le cadre de notre TP4

Application n°1 (code du flow en Annexe 1):

Gestion de l'état (allumer/éteindre) d'une lampe en fonction de la température de la pièce.

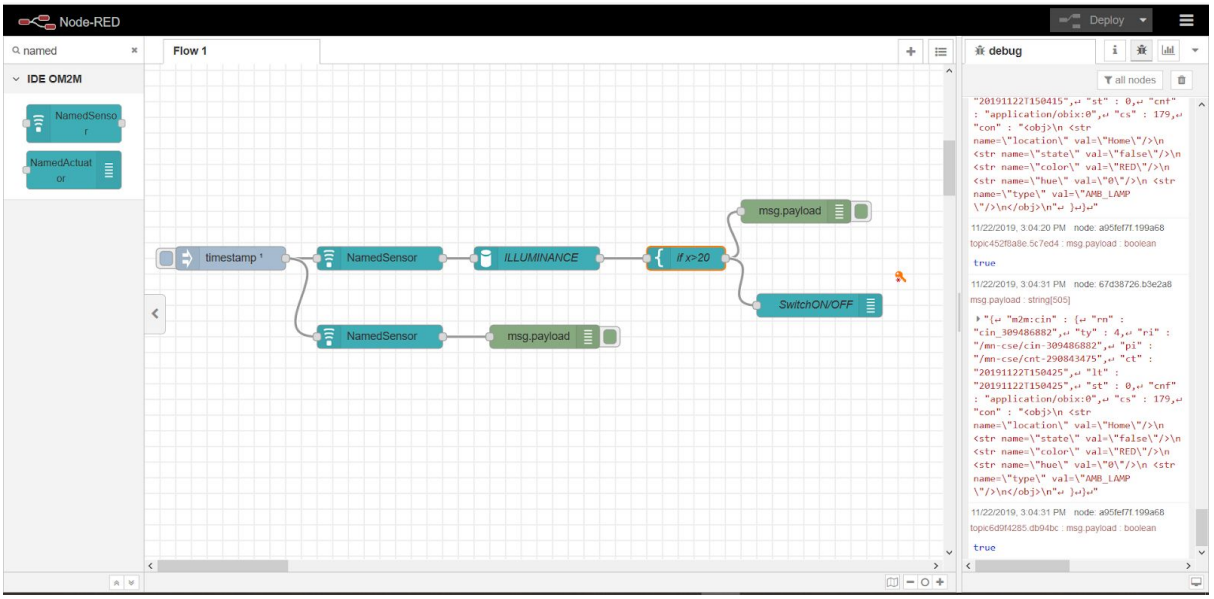


Image 7 : *Gestion d'une lampe Philips Hue avec l'outil Node-Red*

La température dans la pièce étant de 24,8°C au moment de la capture d'écran, la température de la salle dépassant les 20°C, le message "TRUE" nous est renvoyé **(image 7)** et la lampe s'allume.

Application N°2 (Annexe 2) :

Cette application a pour but de récupérer le dernier état de la lampe à chaque timestamp envoyé et d'afficher cet état sur un tableau de bord. Par exemple ci-dessous le dernier état de la lampe est "éteint" donc la valeur "false" est retournée **(image 8)** qui vérifie la condition donc la valeur est affiché sur le tableau de bord **(image 9)**.

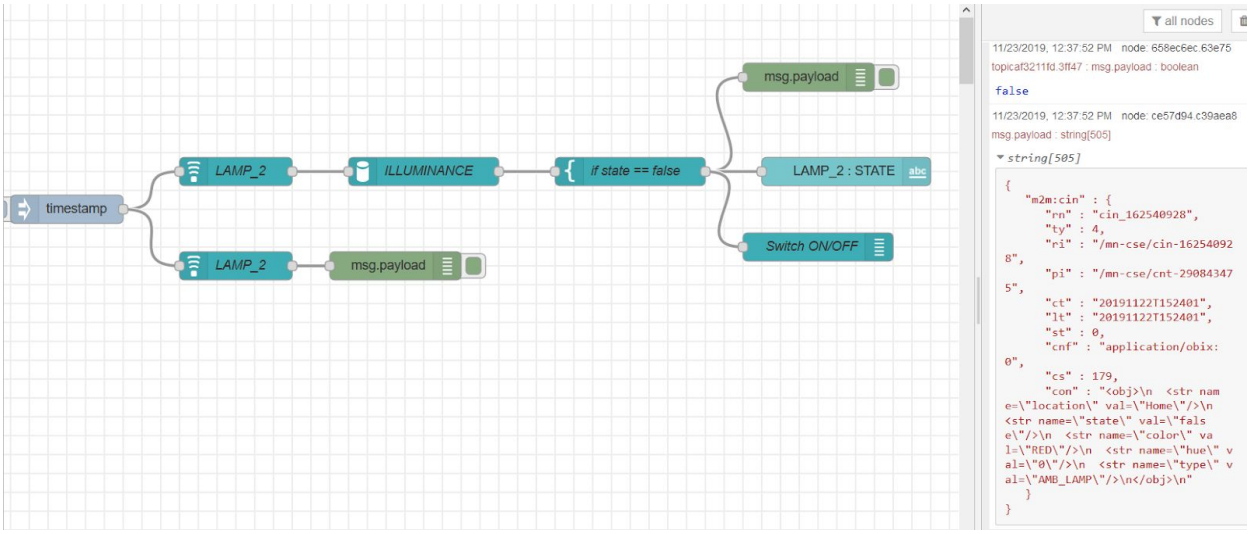


Image 8 : *Gestion d'une lumière Philips Hue en fonction de la température grâce à l'outil Node-Red*

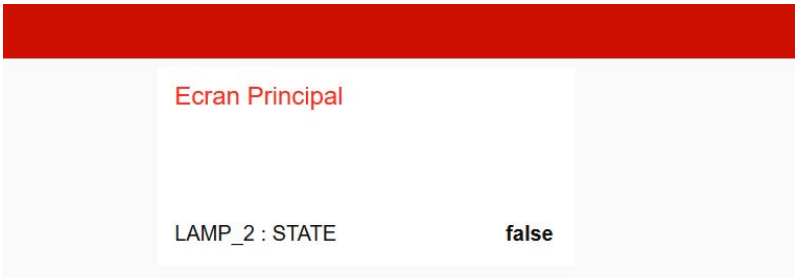


Image 9 : *Dashboard en lien avec les lumières Philips Hue*

Conclusion

Tout au long de ces 4 TPs, nous avons déployé une architecture OM2M qui permet de mettre en place une architecture IoT modulaire. Ainsi, nous avons codé et déployé une

architecture IoT afin de piloter plusieurs objets de manière automatique. ous avons alimentés nos connaissances dans la gestion d'objet IoT.

Le monde de l'IoT est un monde très cosmopolite avec beaucoup de solution propriétaire, ce qui nous a amené à réfléchir à comment rendre interopérable une architecture IoT pour interconnecter ces différentes solutions propriétaires. Enfin, nous avons utilisé un logiciel pour gérer plus facilement ces objets à l'aide d'une programmation visuelle.

Ce projet nous a permis :

- De découvrir une architecture open source pour le monde de l'IoT
- La gestion de différents types d'objets connectés (capteurs, actionneurs…)
- La prise en main de Node-Red pour la création de routine ou de dashboard

Références

https://www.eclipse.org/om2m/ : Site OM2M

https://nodered.org/ : site Node-Red

https://moodle.insa-toulouse.fr/course/view.php?id=785 Moodle INSA Toulouse: annoncé des Travaux Pratiques

Annexes

Annexe 1:

Flow : code Application1

[{"id":"156d9a0b.38546e","type":"tab","label":"TP4","disabled":true,"info":""},{"id":"1ac8b87b.efec4","type":"inject","z":"156d9a0b.38546e","name":"","topic":"","payload":"","payloadType":"date","repeat":"","crontab":"","once":false,"onceDelay":0.1,"x":220,"y":280,"wires":[["74b5b6c5.cb8c58","6660933.06531ec"]]},{"id":"74b5b6c5.cb8c58","type":"NamedSensor","z":"156d9a0b.38546e","name":"FIBARO","platform":"17321277.d5a246","sensor":"d10d4a20.a02218","container":"","cntInstance":"/la","x":520,"y":280,"wires":[["1df00afb.86ad35"]]},{"id":"1df00afb.86ad35","type":"DataExtractor","z":"156d9a0b.38546e","name":"ILLUMINANCE","viewtype":"data","viewunid1":"","viewunid2":"","x":780,"y":280,"wires":[["f74555a.b75f128"]]},{"id":"6660933.06531ec","type":"NamedSensor","z":"156d9a0b.38546e","name":"FIBARO","platform":"17321277.d5a246","sensor":"d10d4a20.a02218","container":"","cntInstance":"/la","x":520,"y":380,"wires":[["5b92a10f.d304"]]},{"id":"5b92a10f.d304","type":"debug","z":"156d9a0b.38546e","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","x":770,"y":380,"wires":[]},{"id":"f74555a.b75f128","type":"SimpleCondition","z":"156d9a0b.38546e",

"operator":">","value_c":"20","inputType":"msg","input":"","name":"if x>20","x":1010,"y":280,"wires":[["e9534afb.acdb7","6fbf6e1e.eb1b48","acd4de41.820018"]]},{"id":"e9534afb.acdb7","type":"debug","z":"156d9a0b.38546e","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","x":1170,"y":200,"wires":[]},{"id":"6fbf6e1e.eb1b48","type":"ui_text","z":"156d9a0b.38546e","group":"ea8f9f23.603cb","order":5,"width":0,"height":0,"name":"","label":"text","format":"{{msg.payload}}","layout":"row-spread","x":1170,"y":280,"wires":[]},{"id":"acd4de41.820018","type":"NamedActuator","z":"156d9a0b.38546e","platform":"17321277.d5a246","name":"Switch ON/OFF","actuator":"d10d4a20.a02218","command":"","x":1200,"y":380,"wires":[]},{"id":"17321277.d5a246","type":"xN_CSE","z":"","platform":"OM2M_1","URLBase":"http://192.168.1.119:8181/~/mn-zwave/mn-name","user":"admin","password":"admin"},{"id":"d10d4a20.a02218","type":"AE","z":"","appId":"Zw_FIBARO_MOTION_SENSOR_3492810500-3"},{"id":"ea8f9f23.603cb","type":"ui_group","z":"","name":"Ecran Principal","tab":"c8399cda.836b88","order":1,"disp":true,"width":"6","collapse":false,"info":""},{"id":"c8399cda.836b88","type":"ui_tab","z":"","name":"Home","icon":"dashboard","disabled":false,"hidden":true}]

Annexe 2:

Flow : code Application2

[{"id":"398b5a4b.7f7b4e","type":"tab","label":"TP4_TEST","disabled":false,"info":""},{"id":"421225b8.d0814c","type":"inject","z":"398b5a4b.7f7b4e","name":"","topic":"","payload":"","payloadType":"date","repeat":"","crontab":"","once":true,"onceDelay":0.1,"x":110,"y":920,"wires":[["6143eebf.e83f78","276aa7ce.675de"]]},{"id":"6143eebf.e83f78","type":"NamedSensor","z":"398b5a4b.7f7b4e","name":"","platform":"17321277.d5a246","sensor":"d10d4a20.a02218","container":"TEMPERATURE","cntInstance":"/la","x":300,"y":920,"wires":[["bff3e5ab.9b464"]]},{"id":"825c9f33.50454","type":"debug","z":"398b5a4b.7f7b4e","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","x":830,"y":860,"wires":[]},{"id":"bff3e5ab.9b464","type":"DataExtractor","z":"398b5a4b.7f7b4e","name":"ILLUMINANCE","viewtype":"data","viewunid1":"","viewunid2":"","x":500,"y":920,"wires":[["a4978dfe.5a169"]]},{"id":"a4978dfe.5a169","type":"SimpleCondition","z":"398b5a4b.7f7b4e","operator":">","value_c":"20","inputType":"msg","input":"payload","name":"if x>20","x":690,"y":920,"wires":[["825c9f33.50454","cb824334.bdda38"]]},{"id":"cb824334.bdda38","type":"NamedActuator","z":"398b5a4b.7f7b4e","platform":"2e9f6e8b.8dbf4a","name":"SwitchON/OFF","actuator":"31a06491.49f3fc","command":"on=true&bri=254&sat=254&id=LAMP_2","x":860,"y":980,"wires":[]},{"id":"276aa7ce.675de","type":"NamedSensor","z":"398b5a4b.7f7b4e","name":"","platform":"2e9f6e8b.8dbf4a","sensor":"31a06491.49f3fc","container":"DATA","cntInstance":"/la","

x":300,"y":1020,"wires":[["8148a626.b07e3"]]},{"id":"8148a626.b07e3","type":"debug","z":"398b5a
4b.7f7b4e","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"pa
yload","targetType":"msg","x":510,"y":1020,"wires":[]},{"id":"c105db42.d07fe8","type":"inject","z":"
398b5a4b.7f7b4e","name":"","topic":"","payload":"","payloadType":"date","repeat":"","crontab":"",
"once":false,"onceDelay":0.1,"x":340,"y":220,"wires":[["3438e48f.c6590c","5830ae9c.19e1"]]},{"id":
"3438e48f.c6590c","type":"NamedSensor","z":"398b5a4b.7f7b4e","name":"LAMP_2","platform":"2
10e7af3.fcd396","sensor":"31a06491.49f3fc","container":"DATA","cntInstance":"/la","x":520,"y":18
0,"wires":[["54566e0b.73a15"]]},{"id":"54566e0b.73a15","type":"DataExtractor","z":"398b5a4b.7f7b
4e","name":"ILLUMINANCE","viewtype":"data","viewunid1":"","viewunid2":"","x":720,"y":180,"wire
s":[["4495c611.d3f8b"]]},{"id":"5830ae9c.19e1","type":"NamedSensor","z":"398b5a4b.7f7b4e","na
me":"LAMP_2","platform":"210e7af3.fcd396","sensor":"31a06491.49f3fc","container":"DATA","cntI
nstance":"/la","x":520,"y":280,"wires":[["ce57d94.c39aea8"]]},{"id":"ce57d94.c39aea8","type":"deb
ug","z":"398b5a4b.7f7b4e","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,
"complete":"false","x":690,"y":280,"wires":[]},{"id":"4495c611.d3f8b","type":"SimpleCondition","z":
"398b5a4b.7f7b4e","operator":"=","value_c":"false","inputType":"msg","input":"","name":"if state
==
false","x":940,"y":180,"wires":[["658ec6ec.63e75","285717f2.69aa08","90533ae3.1105f"]]},{"id":"65
8ec6ec.63e75","type":"debug","z":"398b5a4b.7f7b4e","name":"","active":true,"tosidebar":true,"con
sole":false,"tostatus":false,"complete":"false","x":1130,"y":80,"wires":[]},{"id":"285717f2.69aa08","t
ype":"ui_text","z":"398b5a4b.7f7b4e","group":"ea8f9f23.603cb","order":5,"width":0,"height":0,"na
me":"","label":"LAMP_2 :
STATE","format":"{{msg.payload}}","layout":"row-spread","x":1170,"y":180,"wires":[]},{"id":"90533a
e3.1105f","type":"NamedActuator","z":"398b5a4b.7f7b4e","platform":"210e7af3.fcd396","name":"
Switch
ON/OFF","actuator":"31a06491.49f3fc","command":"","x":1140,"y":260,"wires":[]},{"id":"17321277.
d5a246","type":"xN_CSE","z":"","platform":"OM2M_1","URLBase":"http://192.168.1.119:8181/~/m
n-zwave/mn-name","user":"admin","password":"admin"},{"id":"d10d4a20.a02218","type":"AE","z":"
","appId":"Zw_FIBARO_MOTION_SENSOR_3492810500-3"},{"id":"2e9f6e8b.8dbf4a","type":"xN_CSE
","z":"","platform":"Hue_MN","URLBase":"http://127.0.0.1:8080/~/mn-cse/mn-name","user":"admi
n","password":"admin"},{"id":"31a06491.49f3fc","type":"AE","z":"","appId":"LAMP_2"},{"id":"210e7
af3.fcd396","type":"xN_CSE","z":"","platform":"OM2M-in","URLBase":"http://127.0.0.1:8080/~/mn-
cse/mn-name","user":"admin","password":"admin"},{"id":"ea8f9f23.603cb","type":"ui_group","z":""
,"name":"Ecran
Principal","tab":"c8399cda.836b88","order":1,"disp":true,"width":"6","collapse":false,"info":"#
"},{"id":"c8399cda.836b88","type":"ui_tab","z":"","name":"Home","icon":"dashboard","disabled":fal
se,"hidden":true}]