

# INTERGICIEL POUR L'INTERNET DES OBJETS

RAPPORT DE TP – 5 ISS INSA TOULOUSE



---

Ce document retrace l'ensemble des Travaux pratiques réalisés afin de définir qu'est-ce que le standard oneM2M, sa place dans le monde de l'Internet des Objets et afin d'apprendre à le mettre en oeuvre dans un cas d'utilisation concret.

Étudiants - Groupe A1:

**Mathilde Cornille**  
**Vincent Laurens**

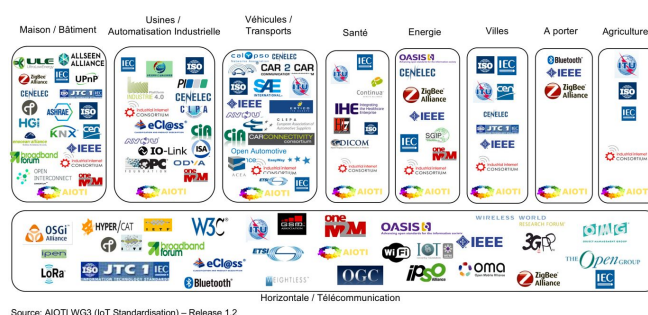
**Encadrant :** Guillaume Garzone

<b>1. Savoir positionner les standards principaux de l'Internet des Objets</b>	<b>2</b>
<b>2. Déployer une architecture conforme à un standard et mettre en place un système de réseau de capteurs aux services</b>	<b>2</b>
2.1. Déployer et configurer une architecture IoT en utilisant OM2M	3
2.2. Interagir avec les objets en utilisant une architecture REST	3
2.3. Intégrer une nouvelle / autre technologie dans une architecture IoT	3
<b>3. Déployer une application composite entre divers technologies grâce à NODE-RED en se basant sur un intergiciel standardisé</b>	<b>3</b>
<b>Bibliographie/Webographie</b>	<b>4</b>

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

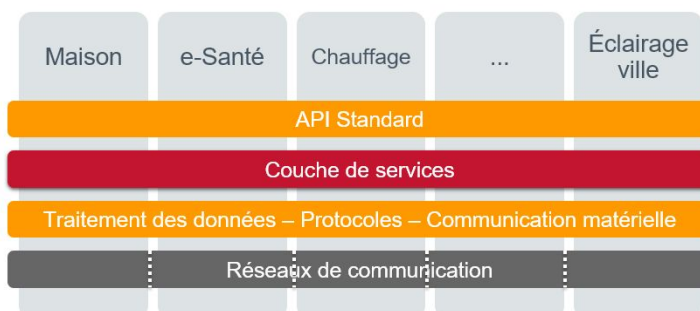
OneM2M est “un consortium mixte composé d’organismes de standardisation et d’industriels”. Mais ce qui fait la force du standard oneM2M est sûrement son ouverture pour la plateforme horizontale. Ceci a encouragé plus de 200 organismes à le soutenir, dont divers organismes de standardisations européens, américains et asiatiques, ainsi que des industriels comme Amazon, Cisco, Huawei, Intel, Interdigital, LG, ou même Orange en France.

L’Internet des Objets constitue un écosystème que l’on pourrait apparenter à une jungle de standards et protocoles tel que l’illustre la figure 1.



**Figure 1 : Jungle des standards et industriels.**

Le standard oneM2M puise sa force dans la vision générique qu’il propose pour le domaine de l’Internet des Objets. Ainsi il se positionne de manière transverse peu importe le domaine d’application. Donc il permet d’interconnecter des systèmes hétérogènes entre-eux et d’offrir par la même occasion une vision homogène de ces systèmes quelque soit le domaine d’application. L’om2m offre une couche de services diversifiés accessible via une interface standardisée comme le montre la figure ci-après.



**Figure 2 : Vue générale des applications du oneM2M**

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

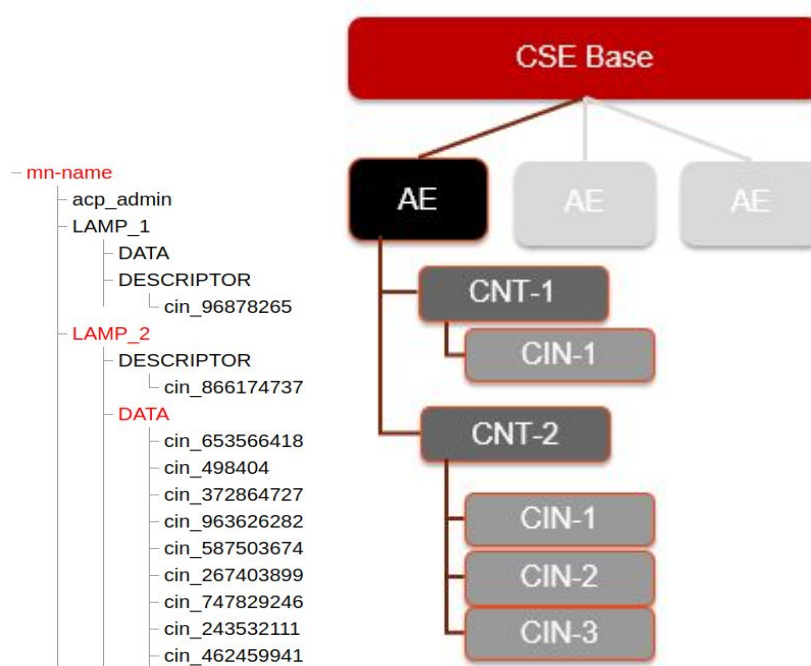
### 2.1. DÉPLOYER ET CONFIGURER UNE ARCHITECTURE IoT EN UTILISANT OM2M

Lors des premières séances de travaux pratiques nous avons déployé une architecture IoT avec OM2M. Cette architecture est constituée d'un serveur centralisé (IN pour Infrastructure Node) qui sert de point d'entrée dans le système IoT, et d'une ou plusieurs passerelles (MN pour Middle Node).

Les IN et MN sont les nœuds principaux de l'architecture OM2M. Ils sont à la base de l'arborescence des ressources. Au cours de ces TP, nous avons utilisés plusieurs types de ressources :

- des Application Entities (AE) qui représentent les différentes applications ou objets connectés
- des Containers (CNT) qui sont des dossiers imbriqués permettant de classifier les données. Généralement nous retrouvons deux CNT caractéristiques : DESCRIPTOR pour stocker les informations générales sur l'AE et DATA pour stocker les valeurs d'intérêt.
- des Content Instance (CIN) qui sont des instances du contenu que l'on souhaite stocker sur la plateforme.

L'objectif de ces TP était d'effectuer le contrôle de lampes Philips Hue (allumer, éteindre, changer l'intensité et la couleur) à travers la plateforme OM2M. L'architecture que nous avons déployée pour ce système peut être trouvée sur la figure n°3



**Figure 3 : Description des ressources**

Les lampes Philips Hue sont définies comme des AE. Pour chacune d'entre elles nous avons défini un CNT DESCRIPTOR qui recense les différentes opérations exécutables (GET, RED, ON, OFF) et un CNT DATA dans lequel seront stockées les informations sur l'état de la lampe. A chaque modification de l'état d'une lampe un CIN est créé sous le container DATA via une requête POST. Afin de récupérer le dernier état il est possible d'utiliser l'attribut "la" qui signifie "latest".

Lors du troisième TP nous avons travaillé sur la réalisation d'un IPE (Interworking Proxy Entity) Java utilisant les bibliothèques Philips Hue. Un IPE est une interface qui sert de traducteur entre un protocole particulier et oneM2M dans les deux sens. Il est possible de faire la traduction entre des objets Java et des ressources OM2M (bottom-up) ou l'inverse (top-down). Au travers de cet IPE nous avons pu interfacer Hue et la plateforme OM2M.

## 2.2. INTERAGIR AVEC LES OBJETS EN UTILISANT UNE ARCHITECTURE REST

Afin d'interagir avec les lampes nous avons utilisé une architecture REST. Nous avons alors pu effectuer différentes requêtes HTTP telles que GET pour récupérer une ressource, POST pour en créer une, PUT pour la modifier et DELETE pour la supprimer. Nous nous sommes servis des méthodes déjà implémentées telles que `retrieve()`, `create()`, `update()` ou `delete()` pour effectuer ces requêtes.

Ces méthodes prennent en argument l'URI de la ressource dans l'arborescence OM2M ainsi que l'originator. Ici nous avons utilisé l'originator "admin:admin" afin d'avoir tous les droits recensés dans la figure n°4 sur les ressources. En effet, il est possible de définir les droits d'un utilisateur en définissant l'ACP (Access Control Policy) comme la somme des droits utilisés. Par exemple, si un utilisateur a uniquement le droit de créer ou récupérer une ressource le code sera 3.

Opération	Code
CREATE	1
RETRIEVE	2
UPDATE	4
DELETE	8
NOTIFY	16
DISCOVERY	32

**Figure 4 : Droits de manipulation des ressources OM2M**

Pour certaines des méthodes il est également nécessaire de spécifier le type de ressource (ty=2 pour les AE, 3 pour les CNT et 4 pour les CIN) et le format (XML ou JSON).

Nous avons ainsi pu créer l'arborescence de notre système sur la plateforme OM2M. Nous avons un CSE-Base qui représente la racine de nos ressources. Le mn-name est un Remote-CSE c'est-à-dire un CSE distant. Nous avons ensuite utilisé, comme présenté précédemment, des AE pour représenter les lampes, des CNT (DESCRIPTOR et DATA) pour chacune des lampes et des CIN, stockés dans le CNT DATA, qui représentent la donnée mesurée.

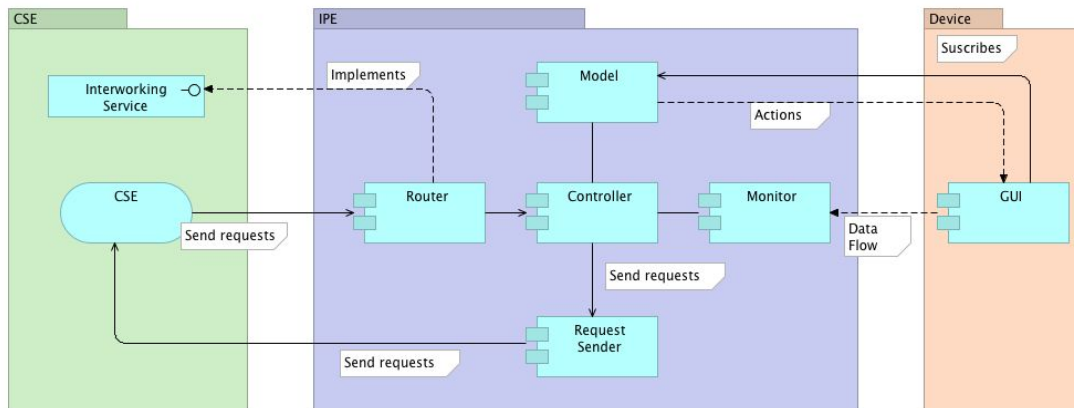
Il est également possible de s'abonner à une ressource en créant une subscription (SUB). Cela permet de faire de la notification asynchrone en étant informé lorsqu'un changement est effectué sur la ressource ou une branche inférieure. Il n'est alors pas nécessaire d'effectuer des requêtes GET périodiques qui peuvent être très consommatrices d'énergie.

### 2.3. INTÉGRER UNE NOUVELLE / AUTRE TECHNOLOGIE DANS UNE ARCHITECTURE IoT

Nous avons déployé un "Interworking Proxy Entity" dont le rôle est d'interconnecter n'importe quel module technologique à une plateforme OM2M. Dans notre cas des lampes directement connectées à un HUE bridge (partie "device" dans la figure suivante) qui envoie des données directement via une passerelle à notre ordinateur qui héberge un serveur IPE et la plateforme OM2M(CSE).

L'IPE et le HUE bridge ont été développés en java suivant le modèle d'architecture suivant.

L'"interworking service" est développé dans le module HUE qui s'assure de la bonne liaison entre le Hue Bridge et la passerelle. L'IPE lui écoute via un moniteur et s'il reçoit une requête il l'interprète et la pousse sur oneM2M.



**Figure 5** : Architecture IPE ([https://wiki.eclipse.org/OM2M/one/IPE\\_Sample](https://wiki.eclipse.org/OM2M/one/IPE_Sample))

Nous avons à la fin pu contrôler notre lampe en envoyant des requêtes HTTP directement via le descripteur du container

Attribute	Value
location	Home
type	AMB_LAMP
unit	
<input type="button" value="GET"/>	/mn-cse/mn-name/LAMP_1/DATA/a
<input type="button" value="ON"/>	/mn-cse/mn-name/LAMP_1? on=true&bri=254&sat=254&id=LAMP_1
<input type="button" value="OFF"/>	/mn-cse/mn-name/LAMP_1?on=false&id=LAMP_1
<input type="button" value="RED"/>	/mn-cse/mn-name/LAMP_1? on=true&bri=254&sat=254&hue=0&id=LAMP_1

“Lamp-1”.

Le bouton “GET” permet de récupérer la dernière valeur de donnée enregistrée dans dernier container d’instance.

ON permet d’allumer la lumière en définissant un niveau de saturation et une couleur à la lumière.

OFF éteint en envoyant le paramètre “on” à faux.

Enfin “RED” autorise l’utilisateur à jouer avec l’intensité de la saturation et la couleur de la lampe.



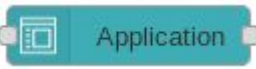
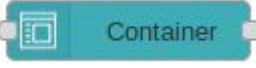

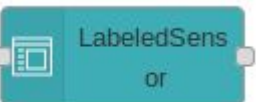
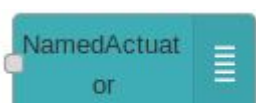


### 3. DÉPLOYER UNE APPLICATION COMPOSITE ENTRE DIVERS TECHNOLOGIES GRÂCE À **NODE-RED** EN SE BASANT SUR UN INTERGICIEL STANDARDISÉ

Node-red est un outil de programmation qui permet de décrire le comportement d’une application. En effet, Node-red donne accès à un nombre important de paquets qui contiennent un ensemble diversifié de noeuds. Ainsi, nous pouvons accéder à l’éditeur facilement au travers d’un navigateur internet et modéliser une application en interconnectant les noeuds ensembles. L’application Node-red s’appuie sur “Nodejs” ce qui la rend légère, facile à configurer et à utiliser, et peut donc être déployée sur des containers ou n’importe quel système d’exploitation. Elle est ainsi déployable sur tout type d’objets comme des Raspberry Pi (un avantage dans le monde de l’internet des objets).

Node-red, est d’autant plus facile à déployer qu’il possède une base de données complète pour tous types d’applications comme le standard oneM2M, les protocoles comme HTTP, COAP ou MQTT ou encore beaucoup d’autres cas d’applications.

Nous allons présenter les principaux noeuds que nous avons utilisé dans nos applications.

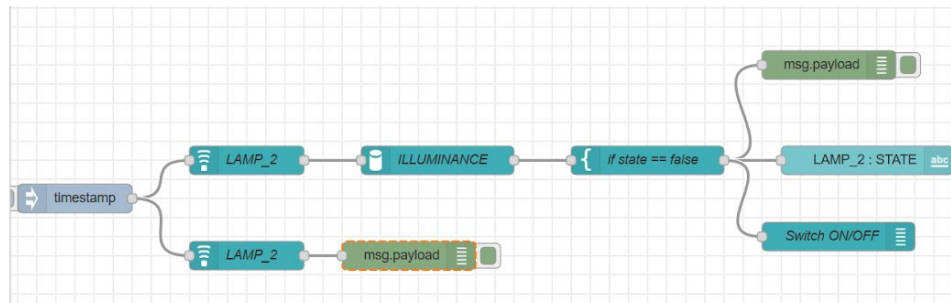
Nous allons décrire chaque noeud et expliquer brièvement pourquoi on les a utilisés.

	Ce noeud permet d'insérer de la données dans le système.
	Ce noeud permet d'afficher sur une console le résultat d'une opération afin d'étudier le bon fonctionnement de la solution, de la déboguer.
	Ce noeud entre dans le standard oneM2M et permet de créer un noeud "Application entity" (AE)
	Ce noeud permet de créer un container oneM2M (CNT)
	Ce noeud crée un container d'instance (CIN) hébergeant la données
	Ce noeud permet de récupérer la valeur d'un capteur en y accédant par un label connu.
	Ce noeud permet d'accéder à un capteur dont on connaît directement le nom ou l'URI.
	Ce noeud permet de poser des conditions dans le programme et ainsi réaliser des tests sur les valeurs de capteurs et réaliser une ou plusieurs actions en conséquence.
	Ce noeud peut être utilisé dans un modèle "Publisher/Subscriber" et il nous permet de créer une entité moniteur dont le rôle va être d'envoyer des notifications aux noeuds qui souscrivent à une ressource, dès qu'une nouvelle valeur est disponible. Ainsi les noeuds souscripteurs n'ont pas à consommer de l'énergie à écouter les canaux de communication.

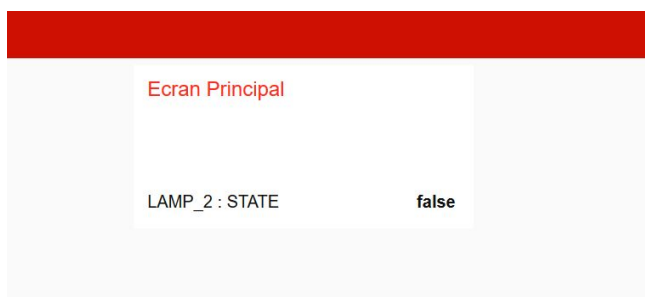
Maintenant que nous avons expliqué le rôle des principaux noeuds node-red nous allons détailler un cas d'utilisation utilisant la majorité des noeuds décrits ci-dessus.

Nous avons dans notre cas développé une application basée sur un capteur incorporé dans une Raspberry Pi (Fibaro) et dont le rôle était d'enregistrer les données dans une arborescence oneM2M.

L'application ainsi développée sur Node-Red a pour but de récupérer la dernière valeur de l'état de la lampe et, en fonction de son état, la modifier. Cette application effectue les opérations codées tout au long des TP 1,2 et 3.



Il affiche l'état sur un dashboard et sur la console de debug comme illustré ci-dessous et permet également de modifier l'état de la lampe à l'aide du noeud 'NameActuator'.



```

msg.payload : string[505]
> "{ "m2m:cin" : { "rn" :
"cin_162540928", "ty" : 4, "ri" :
"/mn-cse/cin-162540928", "pi" :
"/mn-cse/cnt-290843475", "ct" :
"20191122T152401", "lt" :
"20191122T152401", "st" : 0, "cnf"
: "application/obix:0", "cs" : 179,
"con" : "<obj>\n <str
name=\"location\" val=\"Home\"/>\n
<str name=\"state\" val=\"false\"/>\n
<str name=\"color\" val=\"RED\"/>\n
<str name=\"hue\" val=\"0\"/>\n <str
name=\"type\" val=\"AMB_LAMP
\"/>\n</obj>\n" } }"
11/23/2019, 10:27:39 AM node: 658ec6ec.63e75
topicb3e9924c.b8931 : msg.payload : boolean
false
  
```

## CONCLUSION :

Au cours de ces TP nous avons découvert un des standards majeurs dans le développement d'application IoT qui est oneM2M.

Ces TP nous ont amené à établir un état de l'art de tous les standards et protocoles qui peuvent exister et qui constituent le cercle de l'Internet des Objets.

Nous avons également eu l'occasion de développer une application qui permettait au travers d'une API REST de commander des lampes. Ensuite, nous avons étendu l'application en ajoutant un "Middle node" qui nous a amenés à nous interroger sur les problèmes de routage entre les noeuds. Enfin nous avons pu développer des applications Node-red afin de commander, via des tableaux de bords, les lampes en se basant sur résultats de TP précédents.

Le but de cet enseignement a été atteint car nous sommes maintenant capables d'appliquer ces connaissances dans le projet de management des salles INSA en SOA. Ainsi que de développer des tableaux Node-red dans nos projets intégrateurs.



Nous remercions Monsieur Thierry Monteil ainsi que tous les encadrants de TP pour l'aide qu'il nous ont apportée dans la réussite de ces TPs.

## BIBLIOGRAPHIE/WÉBOGRAPHIE

- Site openclassroom oneM2M INSA de Toulouse.
- <https://openclassrooms.com/fr/courses/5079046-mettez-en-place-une-architecture-pour-objets-connectes-avec-le-standard-onem2m/5079156-retrouvez-vous-dans-la-jungle-des-standards>
- [https://wiki.eclipse.org/OM2M/one/IPE\\_Sample](https://wiki.eclipse.org/OM2M/one/IPE_Sample)
- Site officiel de Node-Red: <https://nodered.org/>

## ANNEXES:

### CODES DE L'APPLICATION NODE-RED 1 "COMMANDER L'ÉTAT DE LA LAMP 2".

```
[{"id":"398b5a4b.7f7b4e","type":"tab","label":"TP4_TEST","disabled":false,"info":"","z":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":[["bff3e5ab.9b464"]]},{"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":[["276aa7ce.675de"]]},{"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":398b5a4b.7f7b4e,"name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","x":510,"y":1020,"wires":[["c105db42.d07fe8"]]},{"id":"c105db42.d07fe8","type":"inject","z":398b5a4b.7f7b4e,"name":"","topic":"","payload":"","payloadType":"date","repeat":"","crontab":"","once":false,"onceDelay":0.1,"x":220,"y":180,"wires":[["3438e48f.c6590c","5830ae9c.19e1"]]},{"id":"3438e48f.c6590c","type":"NamedSensor","z":398b5a4b.7f7b4e,"name":"LAMP_2","platform":"210e7af3.fcd396","sensor":"31a06491.49f3fc","container":"DATA","cntInstance":"/la","x":520,"y":180,"wires":[["54566e0b.73a15"]]},{"id":"54566e0b.73a15","type":"DataExtractor","z":398b5a4b.7f7b4e,"name":"ILLUMINANCE","viewtype":"data","viewunid1":"","viewunid2":"","x":780,"y":180,"wires":[["4495c611.d3f8b"]]},{"id":"5830ae9c.19e1","type":"NamedSensor","z":398b5a4b.7f7b4e,"name":"LAMP_2","platform":"210e7af3.fcd396","sensor":"31a06491.49f3fc","container":"DATA","cntInstance":"/la","x":520,"y":280,"wires":[["ce57d94.c39aea8"]]},{"id":"ce57d94.c39aea8","type":"debug","z":398b5a4b.7f7b4e,"name":"","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":false,"x":770,"y":280,"wires":[["4495c611.d3f8b"]]},{"id":"4495c611.d3f8b","type":"SimpleCondition","z":398b5a4b.7f7b4e,"operator":"=","value_c":false,"inputType":"msg","input":"","name":"if state == false","x":1040,"y":180,"wires":[["658ec6ec.63e75","285717f2.69aa08","90533ae3.1105f"]]},{"id":"658ec6ec.63e75","type":"debug","z":398b5a4b.7f7b4e,"name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":false,"x":1170,"y":100,"wires":[["285717f2.69aa08"]]},{"id":"285717f2.69aa08","type":"ui_text","z":398b5a4b.7f7b4e,"group":"ea8f9f23.603cb","order":5,"width":0,"height":0,"name":"","label":"LAMP_2 : STATE","format":"{{msg.payload}}","layout":"row-spread","x":1250,"y":180,"wires":[["90533ae3.1105f"]]},{"id":"90533ae3.1105f","type":"NamedAc
```

```
tuator","z":"398b5a4b.7f7b4e","platform":"210e7af3.fcd396","name":"Switch
ON/OFF","actuator":"31a06491.49f3fc","command":"","x":1200,"y":280,"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":"2e9f6e8b.8dbf4a"
,"type":"xN_CSE","z":"","platform":"Hue_MN","URLBase":"http://127.0.0.1:8080/~mn-cse/mn-name","user":"admin","passwor
d":"admin"}, {"id":"31a06491.49f3fc","type":"AE","z":"","appId":"LAMP_2"}, {"id":"210e7af3.fcd396","type":"xN_CSE","z":"","plat
form":"OM2M-Local","URLBase":"http://127.0.0.1:8080/~mn-cse/mn-name","user":"admin","password":"admin"}, {"id":"ea8f9
f23.603cb","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}}
```