Middleware for the internet of things

TP report – 5 ISS INSA Toulouse

This report covers all the TPs for Internet of Things, it is individual/by pair and will be integrated into your portfolio. The purpose of this report is to show the skills you acquired and the concepts you understood as a result of the courses and the various practical sessions.

The structure can be modified, as long as all the different concepts are addressed. If additional elements seem relevant for you to detail, you are free to integrate them. The size of this report should be limited to 10 pages maximum excluding annexes. (Do not hesitate to attach diagrams in appendix if it helps you to clarify your speech)

**Lastname 1 : *KRISHNA SUJAN***

**Firstname 1 : *INUGANTI***

Group : B2

TP teacher : Garzone Guillaume

# 1. KNOW HOW TO POSITION THE MAIN STANDARDS OF THE INTERNET OF things

***Q) Briefly explain the principle of the oneM2M standard and how it is positioned with respect to other existing standards and technologies.***

***ANSWER :***

OneM2M is an open-source application/platform, providing inter-operatability based on global M2M standards for machine to machine communication. OM2M is developed to run on OSGi layer that makes OM2M more flexible to use it by implementing various plugins like om2m-ttn-ipe for communication with TTN. Further, OM2M supports various network protocols ranging from bluetooth to HTTP to CoAP etc, providing various options to be used for IoT network. Due to RESTful approach along with open interfaces, it further help in developing applications and services without any dependency on the underlying network.

At present, there are many M2M platforms for IoT. However, OM2M differs from them in various aspects.

1. OM2M was developed to be OS independent. Unlike most of the open source platform which are linux based, OM2M can be deployed in Windows, Linux and Mac OS as well.
2. Most of the platform are created for a specific aspect. The platforms are designed and deployed for a specific IoT use case and cannot be re-used. On the other hand, as OM2M is deigned for inter-operatability, helps in resolving the non reusability of the platform.

As OM2M is supported and developed by well renowned institutions, it is positioned at a higher status and used across the globe as it deals with resolving the issue of fragmentation of machine to machine in IoT industry. Further, it provides us the flexibility of inter-operatabilty.

Unlike other platforms, Various interworking proxies are provided to enable seamless communication with vendor-specific technologies such as Zigbee and Phidgets devices.

Several other initiatives like that exist in the word, like ITU-T from the International Communication Union, the MQTT from OASIS which is an ISO standard for TCP/IP protocol

OM2M also helps in reusing the platform for various solution and accomodation of feature development by reusing the deployment.

Unlike other platforms which are developed individualy, OM2M have more than 200 companies involved in developing the platform, for example, Orange company is involved in working on docker solution for OM2M.

# 2. Deploy a standard-compliant architecture and implement a network of sensors

## 2.1. DEPLOY AND CONFIGURE AN IOT ARCHITECTURE USING OM2M

***Q) Explain how you deployed an IoT architecture with OM2M: the types of nodes used, the entities interacting with the middleware, the level at which the objects / sensors interact with the system as well as the different applications interacting with it, and so on. (TP 1 + 2 +3)***

***ANSWER :***

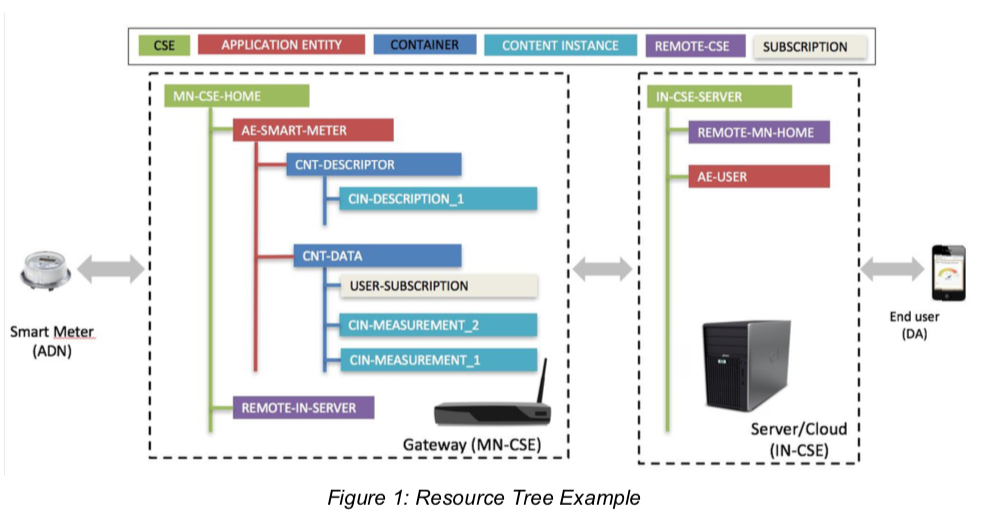
OM2M consists of two major components, IN-CSE(Server) and MN-CSE(Gateway). IN-CSE, the server part is used to send data to the user or different platforms for data analysis and websites. On the other hand MN-CSE is the gateway component and it is used to register new devices(IoT sensors and devices.)

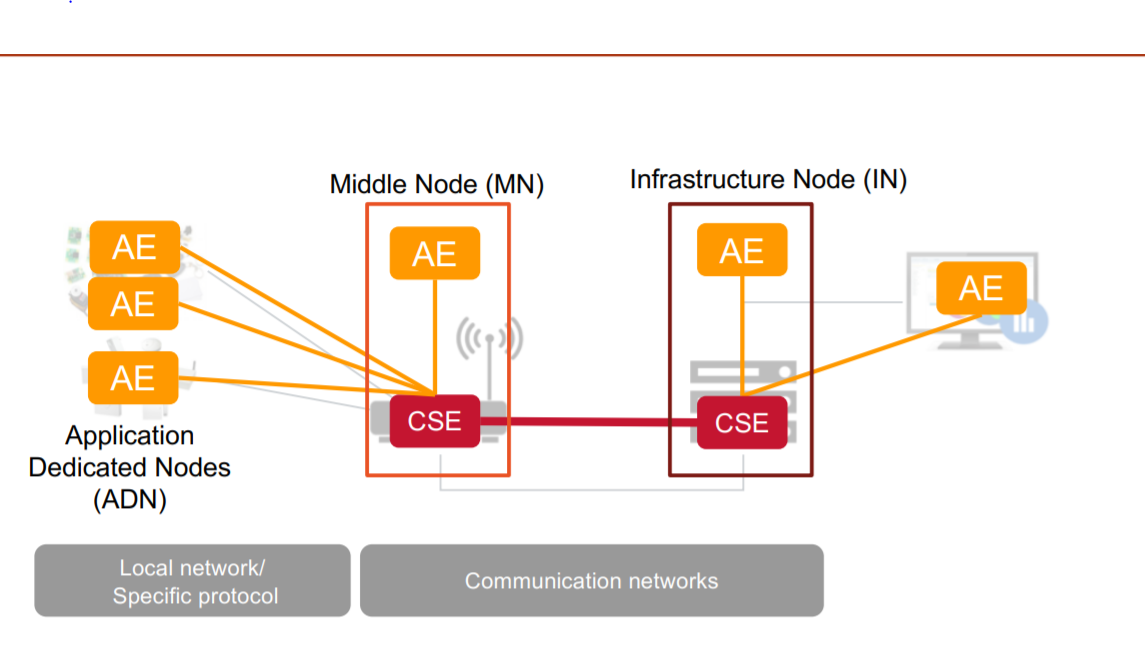
In order to deploy an IoT architecture with OM2M, we used a Server IN-CSE and a Gateway MN-CSE to be able to communicate with end devices. First, we configure the server and the gateway thanks to their config.ini file then we launch the server to register the gateway. The gateway registers its end devices (Application Entity). Each AE contains containers (CNT) which contains measurements (CIN) about the AE.

OM2M consists of two major components, IN-CSE(Server) and MN-CSE(Gateway). IN-CSE, the server part is used to send data to the user or different platforms for data analysis and websites. On the other hand MN-CSE is the gateway component and it is used to register new devices(IoT sensors and devices.)

After building the OM2M locally, from the code available from <https://git.eclipse.org/c/om2m/org.eclipse.om2m.git/snapshot/org.eclipse.om2m-1.3.0.zip>, IN-CSE and MN-CSE components are built using parent pom.xml file. Later, after making the required changes in config.ini file of IN-CSE and MN-CSE are started thereby registering gateway with the server.

OM2M was deployed by us in two different cases. In one case, for the OM2M practicals, both IN-CSE and MN-CSE were deployed.





In the process of deployment, various activities have been observed. The smart devices termed as ADN are registered with the CSE or Common Service Entity server. Multiple AE representing devices were created using RESTful API. Inside each AE, containers are created which basically contains information like the device ID, description etc. Further, data in each container was organised as a distinguished entity which contained various data called content instances.

The user from a device or a platform like data-analytics send a request to IN-CSE which is received by AE-USER, which there on sends information to MN-CSE which is called the gateway which have the information in the form of CIN which is created through API and communicated over http and other network protocols.

However, beyond this we have also implemented OM2M in the academic project social networking for pollution, where we have deployed a single node OM2M containing only the IN-CSE or server which directlycommunicated to the TTN over MQTT an UDP protocol from the IoT devices regietered with the TTN gateway.

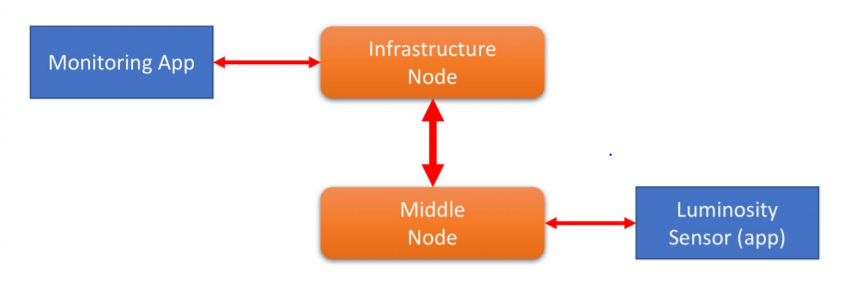
## 2.2. Interact with objects using a rest architecture

***Q) Explain how you could interact with objects and view the data sent by them (including resources generated by IPE Sample, simulated lamps). Explain the different oneM2M resources that you have manipulated (CSE-BASE, AE, CNT, CIN, SUB, REMOTE CSE) and how to interact with them (HTTP client + server). (TP 1 + 2)***

***ANSWER:***

Once the node were configured, we have used similliar deployment as it is presented in the architecture above. The simulating applications, here it is a bulb were basically java applications that were running in the work station.

So, using the bulb simullators, which could receive java commands being sent remotly and so that they could ruturn the state of the similated application.



Apart from this, we could also manipulate various resources using REST api sent from the tools like postman.

Application entity was created and OM2M have generated the unique resource and application ID. Further, using the resource ID generated above was used to create the containers.

Later, using the RI of container many CINs were created which had oBIX contents.

The below sample scripts were used to manipulate the resources.

*<obj>*

*<str name="Type" val="TEST-SENSORS" />*

*<str name="Category" val="LIGHT" />*

*<str name="Unit" val="Lux" />*

*<str name="Model" val="1142\_0" />*

*<str name="Location" val="Home" />*

*<str name="Manufacturer" val="T-PHIDGETS" />*

*<str name="Consumption Max" val="27 mA" />*

*<str name="Voltage Min" val="4.8 V DC" />*

*<str name="Voltage Max" val="5.3 V DC" />*

*<str name="Operating Temperature Min" val="0 C" />*

*<str name="Operating Temperature Max" val="70 C" />*

*</obj>*

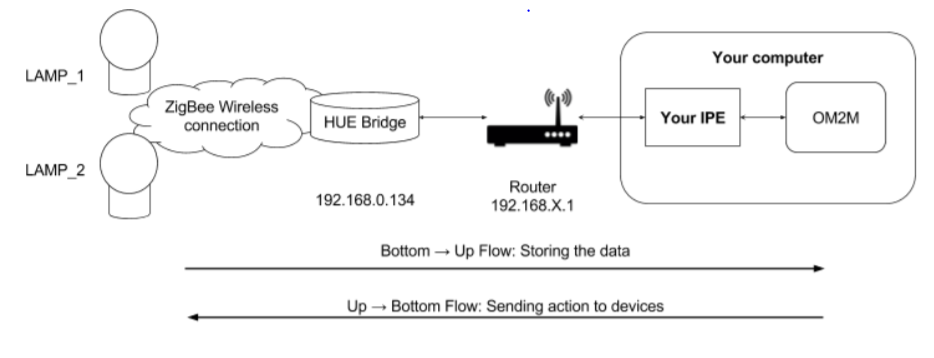
## 2.3. INTEGRATING A NEW / OTHER TECHNOLOGY IN AN existing IOT ARCHITECTURE

**Q) *Explain the principle of interworking Proxy Entity of oneM2M as well as the architecture used to integrate in the middleware a new technology not natively compatible with the standard. (Software architecture, oneM2M nodes used, oneM2M resources used to interface.) (TP 3)***

*ANSWER :*

OM2M, an open source platform for IoT, can be integrated with various technologies and resources using plugins. Interworking Proxy Entity of OM2M is a feature, where the devices are registered and the values are mapped with the OM2M thereby enabling the operations directly by sending commands directly.

While working on the Interworking Proxy Entity, the below architecture was used as a reference.



For the practical, the sample code of Philips JAVA SDK provided in the moodle was used to create an IPE.

Prior starting, in the MN-CSE, various resources like AE, CONTAINER, CIN with data were created. In CIN, two CINs were created where one had bottom to up approach data and the other up to bottom approach data.

Once the resources were developed the JAVA SDK was used to integrate the HUE lamp and the server so that they can be modified by the server through AE-USER. Durning this, the server was connected to HUE bridge along with the reources in OM2M.

Now after configuring the setup with required info, we could now manpulate the luminosity of the lights apart from switching on and off the lights using the POST and GET API calls.

Therefore, it can be concluded that, the basic http request from the web-interface or devices and apps, are converted into compatible requests that could be understood by lamp or other non-default OM2M entities through the data and commands stored in the CIN of CONTAINERS.

# 3. Deploy a composite application using various technologies thanks to NODE-RED,

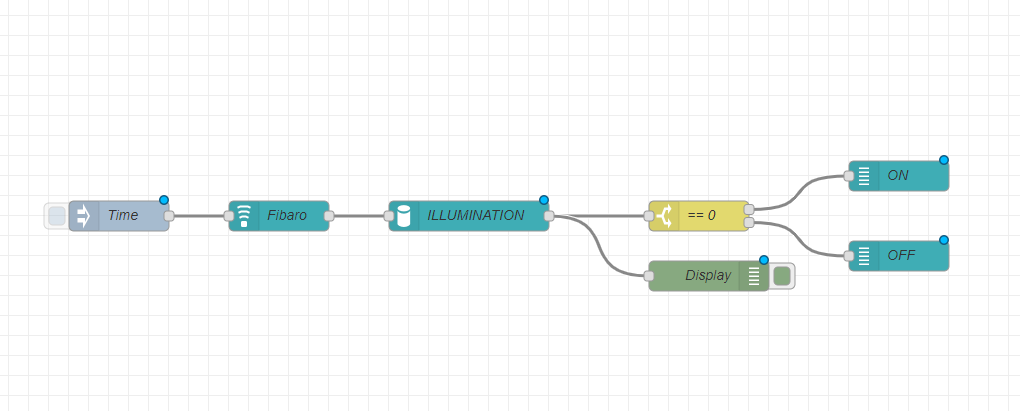
***Briefly explain the operating principle of NODE-RED.***

***Give examples of application(s) you have deployed with NODE-RED (screenshot flow and explanations for instance). Also export your application (export flow > json) and attach file(s) to your report. (TP 4)***

This exercise helps us to understand the interoperability of OM2M. During this exercise, the OM2M platform was made to start and the information was collected from the config.ini file and then we connected to the IPE ADN.

Than we have installed the node-red using the commands provided to us. This OM2M was connected to the node-red by providing the details of the MN-CSE.

Then the below UI was configured to create a flow in the project.



This deployment was created to manipulate the process of controlling the lamp simulators by controlling the luminosity or by turning on and off the bulbs from the node-red console.

The below code was exported for the above flow.

[{"id":"d8fax89j28gq","type":"tab","label":"Flow 1","disabled":false,"info":""},{"id":"p9mkl3w.jki86y","type":"tab","label":"Flow 2","disabled":false,"info":""},{"id":"9e1bb887.d1bdb8","type":"xN\_CSE","z":"","platform":"TP4","URLBase":" http://localhost:8080/~/mn-zwave/mn-name","user":"admin","password":"admin"},{"id":"35e2bcec.7e8264","type":"AE","z":"","appId":""},{"id":"e6f340ac.b8be","type":"AE","z":"","appId":"Zw\_FIBARO\_MOTION\_SENSOR\_90876498501"},{"id":"ux8h3i42.fx","type":"xN\_CSE","z":"","platform":"LAMPS","URLBase":"http://localhost:8080/~/mn-cse/mn-name","user":"admin","password":"admin"},{"id":"dd30e467.855ee8","type":"AE","z":"","appId":"LAMP\_1"},{"id":"ecx9091.41f102ns","type":"inject","z":"d10fax96b699637","name":"Time","topic":"","payload":"","payloadType":"date","repeat":"","crontab":"","once":false,"onceDelay":0.1,"x":270,"y":480,"wires":[["5a783kmw.27e110"]]},{"id":"5ab6ebb2.41e304","type":"NamedSensor","z":"d8fac45a.647158","name":"Fibaro","platform":"9e1bb887.d1bdb8","sensor":"e6f340ac.b8be","container":"ILLUMINANCES","cntInstance":"/la","x":430,"y":480,"wires":[["1c133b0.5704ec5"]]},{"id":"913ba39a.3c612","type":"debug","z":"d8fac45a.647158","name":"Display","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","x":860,"y":540,"wires":[]},{"id":"1c133b0.5704ec5","type":"DataExtractor","z":"msa.d,fnwa.647158","name":"ILLUMINATION","viewtype":"data","viewunid1":"","viewunid2":"","x":620,"y":480,"wires":[["a7a4155a.4c07a8","913ba39a.3c612"]]},{"id":"a7a4155a.4c07a8","type":"switch","z":"d8fac45a.647158","name":"== 0","property":"payload","propertyType":"msg","rules":[{"t":"eq","v":"0.0","vt":"str"},{"t":"neq","v":"0.0","vt":"str"}],"checkall":"true","repair":false,"outputs":2,"x":850,"y":480,"wires":[["12c042b7.796f1d"],["alkdsf2313654q.mk84"]]},{"id":"55dsfa4qw.565fwre","type":"NamedActuator","z":"09fsd0fmc23","platform":"asdnf98ifs56cq","name":"ON","actuator":"dd30e467.855ee8","command":"op=setOn&lampid=LAMP\_1","x":1050,"y":440,"wires":[]},{"id":"d547a32a.da3fa","type":"NamedActuator","z":"d8fac45a.647158","platform":"3b83e01c.38f68","name":"OFF","actuator":"dd30e467.855ee8","command":"op=setOff&lampid=LAMP\_1","x":1050,"y":520,"wires":[]}]