Protocol Proposal for Practical Implementation V3

Internet of Things (2IMN15) 2016-2017, Eindhoven University of Technology
By Leila Fatmasari Rahman (L.F.Rahman@tue.nl)

Table of Content

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of Content</td>
<td>1</td>
</tr>
<tr>
<td>Document Change History in V3</td>
<td>2</td>
</tr>
<tr>
<td>Introduction</td>
<td>2</td>
</tr>
<tr>
<td>System Deployment</td>
<td>3</td>
</tr>
<tr>
<td>Use Cases</td>
<td>4</td>
</tr>
<tr>
<td>Light Device: LWM2M Client</td>
<td>5</td>
</tr>
<tr>
<td>Light Profile (Object Id: 10250)</td>
<td>6</td>
</tr>
<tr>
<td>Configuration File: OwnershipPriority.json</td>
<td>8</td>
</tr>
<tr>
<td>Sensor Device: LWM2M Client</td>
<td>9</td>
</tr>
<tr>
<td>Sensor Profile (Object Id: 10350)</td>
<td>10</td>
</tr>
<tr>
<td>Protocols for the Use Cases</td>
<td>11</td>
</tr>
<tr>
<td>Execute the System</td>
<td>11</td>
</tr>
<tr>
<td>Set User Account</td>
<td>13</td>
</tr>
<tr>
<td>Set Identity and Binding</td>
<td>14</td>
</tr>
<tr>
<td>Set Ownership Priority</td>
<td>16</td>
</tr>
<tr>
<td>Observe State</td>
<td>16</td>
</tr>
<tr>
<td>Request Lighting Usage and Desk Occupancy Report</td>
<td>19</td>
</tr>
<tr>
<td>Update Priority Ownership</td>
<td>20</td>
</tr>
<tr>
<td>Set Behavior Deployment</td>
<td>20</td>
</tr>
<tr>
<td>Update Light Behavior</td>
<td>21</td>
</tr>
<tr>
<td>Adjust Lighting</td>
<td>22</td>
</tr>
<tr>
<td>Adaptive Lighting Based on Presence</td>
<td>23</td>
</tr>
<tr>
<td>Distributed Behavior Deployment</td>
<td>23</td>
</tr>
<tr>
<td>Centralized Behavior Deployment</td>
<td>25</td>
</tr>
</tbody>
</table>
Document Change History in V3

<table>
<thead>
<tr>
<th>No</th>
<th>Changes</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fix Typos</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Add Table of Content</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Add Introduction</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>Update the use case diagram for Building Manager. One new use case is added called “Set Behavior Deployment”</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Add Dim Color information in Light Color resource in the Light Profile object</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>Add two new resources to the Light Profile object: Ownership Priority and Light Behavior for updating ownership priority configuration and the light behavior software.</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>Remove Firmware Update object</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Add description and avahi command templates in the “Execute the System” use case</td>
<td>11</td>
</tr>
<tr>
<td>9</td>
<td>Add one more field in the User Account Data: “RoomID”</td>
<td>13</td>
</tr>
<tr>
<td>10</td>
<td>Add the description for “Set Behavior Deployment” use case</td>
<td>20</td>
</tr>
</tbody>
</table>

Introduction

This document serves as the latest reference for implementing the IoT Practical assignment. This document provides updates and more detailed information to the IoT Practical Description found at [http://www.win.tue.nl/~johanl/educ/2IMN15/PracticalDescription-2IMN15/](http://www.win.tue.nl/~johanl/educ/2IMN15/PracticalDescription-2IMN15/). Protocols and details involved in the implementation of each use cases are discussed in this document.
System Deployment

![Diagram of System Deployment]

Figure 1: Deployment View of the Office Lighting System
Use Cases

Office Lighting System

Figure 2: Use Cases that involve the Building Manager
Office Lighting System

Office Worker

Experience adaptive lighting based on presence

Adjust lighting

Lights and Sensors

Figure 3: Use Cases that involve the Office Worker

Light Device: LWM2M Client

This section defines a Light device (which is an LWM2M client residing in the Raspberry Pi) with UDP interface which includes the instantiated Objects and their value.

The Light device has the Endpoint Name “Light-Device-GroupNo-LightDeviceNo”, for example, Light Device 1 of Group 1 will have endpoint name “Light-Device-1-1”.

The objects inside the Light device are chosen from one type of specification:

1. Private objects by using free range id 10241–32768
   ([http://technical.openmobilealliance.org/Technical/technical-information/omna/lightweight-m2m-lwm2m-object-registry](http://technical.openmobilealliance.org/Technical/technical-information/omna/lightweight-m2m-lwm2m-object-registry))

<table>
<thead>
<tr>
<th>Object</th>
<th>Object ID</th>
<th>Object Instance ID</th>
<th>Notes</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light Profile</td>
<td>10250</td>
<td>0</td>
<td>Represent the Light Device’s static and dynamic information</td>
<td>Private object</td>
</tr>
</tbody>
</table>

Table 1: Object Instance of the Light Device
On the following we define the value for each resources in the objects. In the notes column we describe whether the resource is used or not. When it is not used, handlers for the operations Read, Write, Execute and Observe for that resource do not need to be implemented.

**Light Profile (Object Id: 10250)**

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>Resource ID</th>
<th>Operation</th>
<th>Type</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Type</td>
<td>1</td>
<td>R</td>
<td>String</td>
<td>“Light Device”</td>
<td>The device type, in this case a “Light Device”</td>
</tr>
<tr>
<td>Light State</td>
<td>2</td>
<td>R, W</td>
<td>String</td>
<td>“USED” or “FREE”</td>
<td>Represents the state of the Light Device, whether it is in the “USED” or the “FREE” state</td>
</tr>
<tr>
<td>User Type</td>
<td>3</td>
<td>R, W</td>
<td>String</td>
<td>“USER1”, “USER2”, “USER3”</td>
<td>Represents the type of user that is using the Light Device, whether it is “USER1”, “USER2” or “USER3”</td>
</tr>
<tr>
<td>Light Color</td>
<td>5</td>
<td>R, W</td>
<td>String</td>
<td>“(r, g, b)”</td>
<td>Represents the current color on the Sense Hat’s LED matrix. “(r, g, b)” is a string which represents a tuple containing the RGB (red, green, blue) values of the color. Each element must be an integer between 0 and 255. Examples of LED color represented in (r, g, b): Off = “(0, 0, 0)” Red = “(255, 0, 0)” White = “(255, 255, 255)” Dim = “(250, 200, 100)”</td>
</tr>
<tr>
<td>Low Light</td>
<td>6</td>
<td>R, W</td>
<td>Boolean</td>
<td>True, False</td>
<td>Represents the current color intensity on the Sense Hat’s LED matrix. True = Low light mode on, False = Low light mode off</td>
</tr>
<tr>
<td>Group No</td>
<td>7</td>
<td>R, W</td>
<td>Integer</td>
<td>Group Number that the Light Device belongs to. Each desk is assigned to a group of light(s) and a sensor. It also represents the group number who is implementing the end devices.</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>---</td>
<td>------</td>
<td>---------</td>
<td>----------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Location X</td>
<td>8</td>
<td>R, W</td>
<td>Float</td>
<td>X location of the Light Device (approximation value in meter unit) relative to a reference point (0, 0) in the room.</td>
<td></td>
</tr>
<tr>
<td>Location Y</td>
<td>9</td>
<td>R, W</td>
<td>Float</td>
<td>Y location of the Light Device (approximation value in meter unit) relative to a reference point (0, 0) in the room.</td>
<td></td>
</tr>
<tr>
<td>Room ID</td>
<td>10</td>
<td>R, W</td>
<td>String</td>
<td>“Room-No” The ID of the room where the Light Device is located. For example “Room-1”, “Room-2”, etc</td>
<td></td>
</tr>
<tr>
<td>Behavior Deployment</td>
<td>11</td>
<td>R, W</td>
<td>String</td>
<td>“Broker” or “Distributed” Stating which lighting behavior deployment is used, broker deployment or distributed deployment. The default value is “Distributed”. When the system decides to use behavior deployment in the broker, the value is changed to “Broker”, and the behavior deployment in the Light Device is de-activated.</td>
<td></td>
</tr>
<tr>
<td>Ownership Priority</td>
<td>12</td>
<td>W</td>
<td>String</td>
<td>0-255 bytes An HTTP(S) URL to a JSON-formatted configuration file (see OwnershipPriority.json below). Start downloading when the value is written and apply the update on download completion.</td>
<td></td>
</tr>
<tr>
<td>Light Behavior</td>
<td>13</td>
<td>W</td>
<td>String</td>
<td>0-255 bytes An HTTP(S) URL to a lighting behavior update file. The contents are specific to the implementation of distributed behavior deployment by the Red team. Start downloading when the value is written and apply the update on download completion</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Resources in the Light Profile Object (Object Id: 10250)
Configuration File: OwnershipPriority.json

Below is example of the JSON format of configuration file OwnershipPriority.json

```json
[
  {
    "user_type": "USER1",
    "user_id": "Office-Worker-1",
    "light_color": "(255, 255, 255)",
    "low_light": false,
    "user_location_x": 2,
    "user_location_y": 1
  },
  {
    "user_type": "USER2",
    "user_id": "Office-Worker-3",
    "light_color": "(255, 0, 0)",
    "low_light": true,
    "user_location_x": 4,
    "user_location_y": 1
  },
  {
    "user_type": "USER3",
    "user_id": "Office-Worker-20",
    "light_color": "(0, 0, 255)",
    "low_light": false,
    "user_location_x": 2,
    "user_location_y": 3
  },
  {
    "user_type": "USER3",
    "user_id": "Office-Worker-25",
    "light_color": "(0, 0, 255)",
    "low_light": false,
    "user_location_x": 4,
    "user_location_y": 3
  },
  {
    "user_type": "USER3",
    "user_id": "Office-Worker-40",
    "light_color": "(0, 0, 0)",
    "low_light": false,
    "user_location_x": 2,
    "user_location_y": 3
  }
]
```
"user_location_y": 5
},
{
  "user_type": "USER3",
  "user_id": "Office-worker-30",
  "light_color": "(0, 255, 255)",
  "low_light": true,
  "user_location_x": 4,
  "user_location_y": 5
}
]

Sensor Device: LWM2M Client

This section defines a Sensor device (which is an LWM2M client residing in the Raspberry Pi) with UDP interface which includes the instantiated Objects and their value.

The Sensor device has the Endpoint Name “Sensor-Device-GroupNo-SensorDeviceNo”, which means that (for example) Group 1 will have endpoint name “Sensor-Device-1-1” for their sensor device 1.

The objects inside the Light device are chosen from one type of specification:

1. Private objects by using free range id 10241–32768
   (http://technical.openmobilealliance.org/Technical/technical-information/omna/lightweight-m2m-lwm2m-object-registry)

<table>
<thead>
<tr>
<th>Object</th>
<th>Object ID</th>
<th>Object Instance ID</th>
<th>Notes</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Profile</td>
<td>10350</td>
<td>0</td>
<td>Represent the Sensor Device’s static and dynamic information</td>
<td>Private Object</td>
</tr>
</tbody>
</table>

Table 3: Object Instance of the Sensor Device

On the following we define the value for each resources in the objects. In the notes column we describe whether the resource is used or not. When it is not used, handlers for the operations Read, Write, Execute and Observe for that resource do not need to be implemented.
## Sensor Profile (Object Id: 10350)

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>Resource ID</th>
<th>Operation</th>
<th>Type</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Type</td>
<td>1</td>
<td>R</td>
<td>String</td>
<td>“Sensor Device”</td>
<td>The device type, in this case a “Sensor Device”</td>
</tr>
<tr>
<td>Sensor State</td>
<td>2</td>
<td>R, W</td>
<td>String</td>
<td>“OCCUPIED” or “FREE”</td>
<td>Represents the state of the Sensor Device, whether it is in the “OCCUPIED” or the “FREE” state</td>
</tr>
<tr>
<td>Group No</td>
<td>4</td>
<td>R, W</td>
<td>Integer</td>
<td></td>
<td>Group Number that the Sensor Device belongs to. Each desk is assigned to a group of light(s) and a sensor. It also represents the group no who is implementing the end devices.</td>
</tr>
<tr>
<td>Location X</td>
<td>5</td>
<td>R, W</td>
<td>Float</td>
<td></td>
<td>X location of the Sensor Device (approximation value in meter unit) relative to a reference point (0, 0) in the room.</td>
</tr>
<tr>
<td>Location Y</td>
<td>6</td>
<td>R, W</td>
<td>Float</td>
<td></td>
<td>Y location of the Sensor Device (approximation value in meter unit) relative to a reference point (0, 0) in the room.</td>
</tr>
<tr>
<td>Room ID</td>
<td>7</td>
<td>R, W</td>
<td>String</td>
<td>“Room-No”</td>
<td>The ID of the room where the Sensor Device is located. For example “Room-1”, “Room-2”, etc</td>
</tr>
</tbody>
</table>

Table 4: Resources in the Sensor Profile Object (Object Id: 10350)
Protocols for the Use Cases

In this section we will describe about data exchange between the different parts and the interactions between the LWM2M server and the Light Device and Sensor Device for each use case.

Execute the System

The use case “Execute the System” is done by first running the broker, second running the end devices, and last, running the Cloud Service. The execution of each part is done independently and is not done centrally by the building manager. When the broker runs, it publishes its service to the network using mDNS/DNS-SD protocol by calling the following command template:

```
avahi-publish-service Broker-Room-No _coap._udp 5683 “/broker” --sub _roomNo._sub._coap._udp
```

Example of publishing the service of Broker for Room 1 (there is exactly one broker in each room):

```
avahi-publish-service Broker-Room-1 _coap._udp 5683 “/broker” --sub _room1._sub._coap._udp
```

When the end devices are run, they first try to discover the broker’s IP address using mDNS/DNS-SD protocol by calling the following command template:

```
avahi-browse -rtp _roomNo._sub._coap._udp
```

Example of discovering the service of Broker for Room 1 (there is exactly one broker in each room):

```
avahi-browse -rtp _room1._sub._coap._udp
```

Once the IP address of the broker is found, the end devices register themselves to the LWM2M Server residing in the broker. Figure 4 and Figure 5 shows this registration interaction.

When the Cloud Service are run, they first try to discover the broker’s IP address using mDNS/DNS-SD protocol by calling the following command:

```
avahi-browse -rtp _coap._udp
```

In theory, the Cloud Service should be able to connect to all the brokers in different rooms. However, for the Plug Fest and the assignment, we pair one Cloud to one Broker only.
Figure 4: Interaction between the Light Device and LWM2M server in the broker during “Execute the System” use case

Figure 5: Interaction between the Sensor Device and LWM2M server in the broker during “Execute the System” use case
Set User Account

The building manager sends the following users’ data (Table 5) to the broker during the “Set User Account” use case.

<table>
<thead>
<tr>
<th>No</th>
<th>Data</th>
<th>Format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UserID</td>
<td>“Office-Worker-GroupNo”</td>
<td>“Office-Worker-25”</td>
</tr>
<tr>
<td>2</td>
<td>GroupNo</td>
<td>Integer</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>RoomID</td>
<td>“Room-No”</td>
<td>“Room-1”</td>
</tr>
<tr>
<td>4</td>
<td>Name</td>
<td>“Last Name, First Name”</td>
<td>“Rahman, L.F.”</td>
</tr>
<tr>
<td>5</td>
<td>Email</td>
<td>Email format</td>
<td>“<a href="mailto:L.F.Rahman@tue.nl">L.F.Rahman@tue.nl</a>”</td>
</tr>
<tr>
<td>6</td>
<td>Password</td>
<td>“pwd-GroupNo”</td>
<td>“pwd-25”</td>
</tr>
</tbody>
</table>

Table 5: User Account Data

Please note that the building manager has to assign exactly one user to a Group No. Group No identifies a group of end devices which are assigned to a desk. The Group No also represents the group number who implements the end-devices. These user data is then stored in a database in the broker and will be used for the User Authentication process. An office worker need to log in using her/his User ID and Password on the user app. The office worker can only adjust lighting through the app when she or he is occupying her/his desk.

As a revision to the practical description, face pattern data do not need to be sent to the broker as they are not needed. The sensor (camera) device only need to detect a face, which will trigger state change to “OCCUPIED”. The sensor (camera) device does not need to recognize the face.
Set Identity and Binding

Figure 6: Interaction between the Light Device and LWM2M server in the broker during “Set Identity and Binding” use case
Figure 7: Interaction between the Sensor Device and LWM2M server in the broker during “Set Identity and Binding” use case
Set Ownership Priority

![Diagram showing the interaction between Light Device (LWM2M Client) and Broker (LWM2M Server).]

Write URL of the configuration file OwnershipPriority.json to the Ownership Priority resource in the Light Profile object.

Write /10250/0/12 URL of OwnershipPriority.json

Success URL of OwnershipPriority.json

Figure 8: Interaction between the Light Device and LWM2M server in the broker during “Set Priority Ownership” use case

Observe State

![Diagram showing the interaction between Light Device (LWM2M Client) and LWM2M Server.]

Observe the value changes of Light State Resource in the Light Profile Object.

GET /10250/0/2 Observe

2.05 Content Observe

... Notify

USED | FREE

...
Observe the User Type Resource in the Light Profile Object

GET /10250/0/3 Observe

2.05 Content Observe

... Notify

USER1 | USER2 | USER3

Observe the User ID Resource in the Light Profile Object

GET /10250/0/4 Observe

2.05 Content Observe

... Notify

User ID
Figure 9: Interaction between the Lighting Device and LWM2M server in the broker during “Observe State” use case
Figure 10: Interaction between the Sensor Device and LWM2M server in the broker during “Observe State” use case

Request Lighting Usage and Desk Occupancy Report

In this use case, the building manager, through the building manager app, can request the following information:

1. Total time each Light Device is used by USER1, USER2 and USER3 in a period of time. Each light is identified by its ID.
2. Total time each Light Device is in a Dim Light in a period of time. Each light is identified by its ID.
3. Total time each Light Device is off in a period of time. Each light is identified by its ID.
4. Total time each Sensor Device is in “OCCUPIED” state and in “FREE” State. Each sensor is identified by its ID.

The groups working on the cloud part are free to decide on how to display this information on the building manager app. Good representation of these information will be appreciated.
Update Priority Ownership

Figure 11: Interaction between the Light Device and LWM2M server in the broker during “Update Priority Ownership” use case

Set Behavior Deployment

The cloud service needs to provide a User Interface for the Building Manager to set the lighting behavior to “Broker” (centralized) or “Distributed”. The default active behavior is the distributed behavior.

Figure 12: Interaction between the Light Device and LWM2M server in the broker during “Set Behavior Deployment” use case
Update Light Behavior

Light Device (LWM2M Client)  Broker (LWM2M Server)

Write URL of the Light Behavior software update to the Light Behavior resource in the Light Profile object  Write /10250/0/13

URL of Light Behavior software update  Success

Figure 13: Interaction between the Light Device and LWM2M server in the broker during “Update Light Behavior” use case
Adjust Lighting

Figure 14: Interaction between the Light Device and LWM2M server in the broker during “Adjust Lighting” use case
Adaptive Lighting Based on Presence

Distributed Behavior Deployment

In the distributed behavior deployment, each sensor device publishes its sensor state changes to the MQTT broker. The topic format for the publish message is:

“TUE/Room-No/Sensor/Sensor-Device-GroupNo-SensorDeviceNo/State”

The payload of the publish message can either be “OCCUPIED” or “FREE”. Figure 15: Interaction between the Sensor Device and MQTT Broker during “Adaptive Lighting Based on Presence” use case if using distributed behavior deployment. Figure 15 shows interaction example between the Sensor Device and the MQTT broker.

Figure 15: Interaction between the Sensor Device and MQTT Broker during “Adaptive Lighting Based on Presence” use case if using distributed behavior deployment

The light devices on the other hand, subscribes to the state changes of all sensor devices in the room. The light devices then subscribe to the following topic format:

“TUE/Room-No/Sensor/+/-State”

Every time a sensor device publishes a state change to the MQTT Broker, the light devices will receive this update. Figure 16 shows interaction example between the Light Device and the MQTT broker.
Figure 16: Interaction between the Light Device and MQTT Broker during “Adaptive Lighting Based on Presence” use case if using distributed behavior deployment.
Centralized Behavior Deployment

In centralized behavior deployment, the broker determines the behavior of the light device based on the states of the sensors in the room. Therefore, the broker observes the state changes of the sensors through its LWM2M Server as shown in Figure 17, and it changes the light settings of the corresponding light devices, as well as updating their state, user type and user id as shown in Figure 18. The implementation of the centralized behavior deployment is optional for the broker part, however if implemented, bonus points will be given.

![Diagram of Centralized Behavior Deployment]

Figure 17: Interaction between the Light Device and LWM2M server in the broker during “Adaptive Lighting Based on Presence” use case if using centralized behavior deployment
Figure 18: Interaction between the Light Device and LWM2M server in the broker during “Adaptive Lighting Based on Presence” use case if using centralized behavior deployment