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

Document Change History in V3

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

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/ . Protocols and details involved in the implementation of each use cases are discussed in this document.

System Deployment

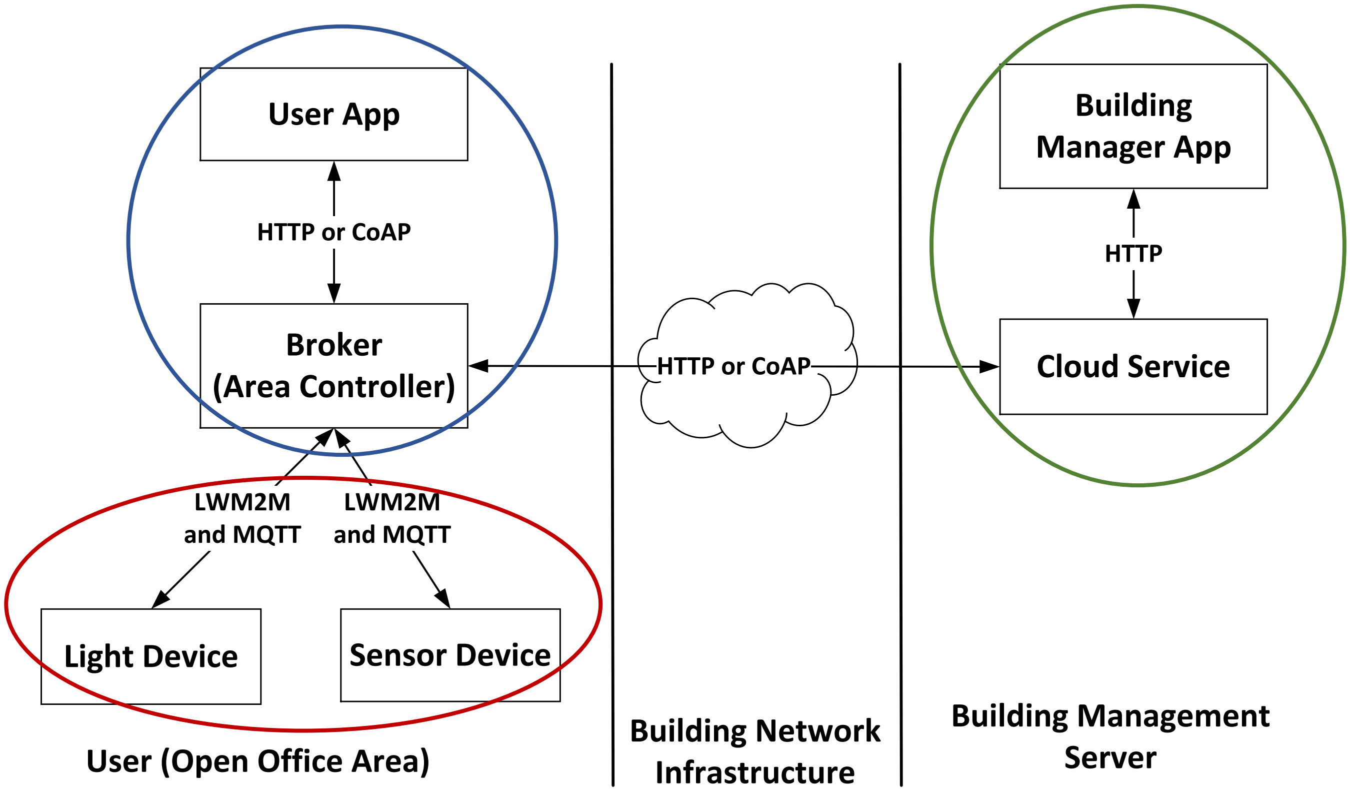


Figure 1 : Deployment View of the Office Lighting System

Use Cases

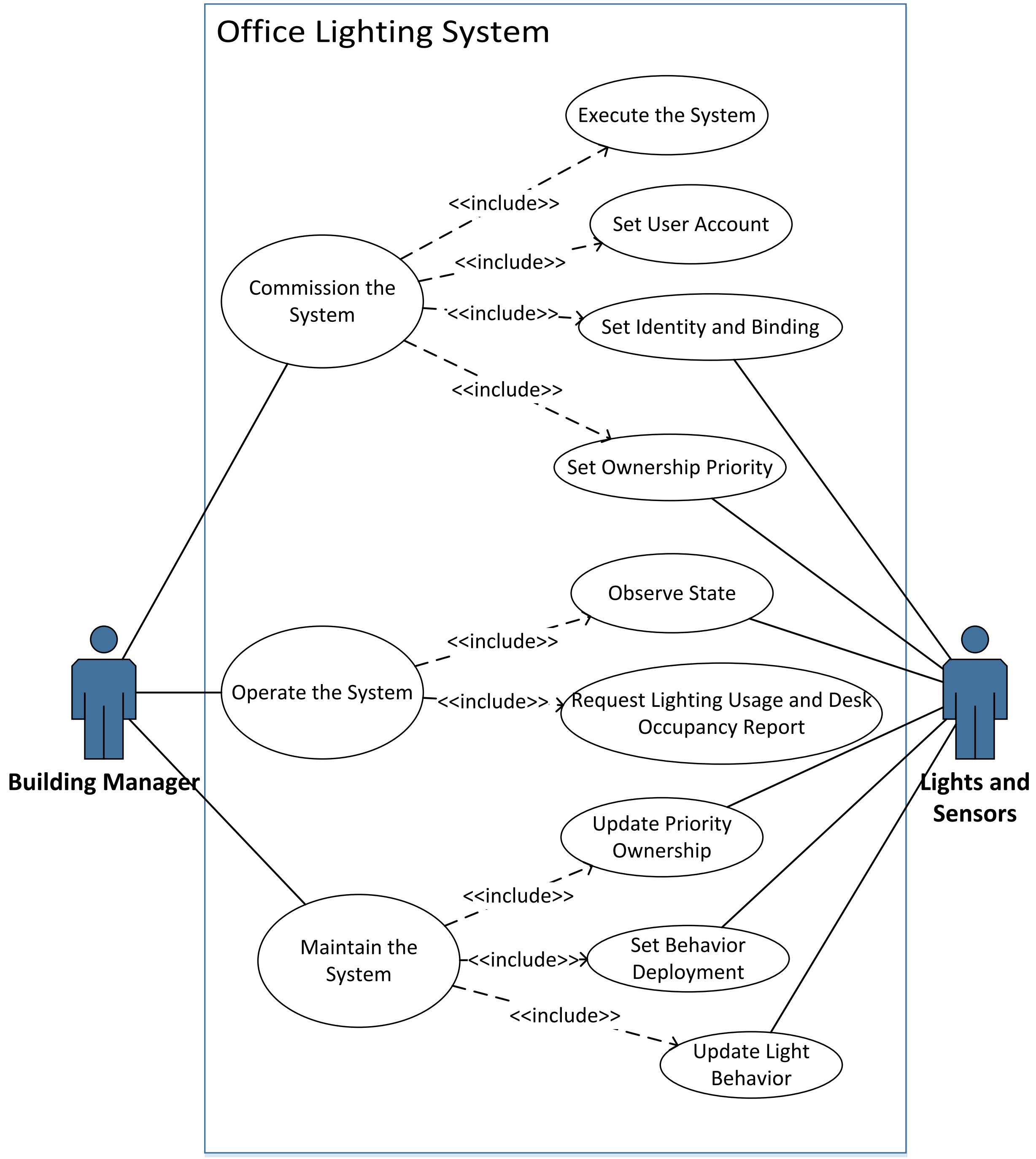


Figure 2 : Use Cases that involve the Building Manager

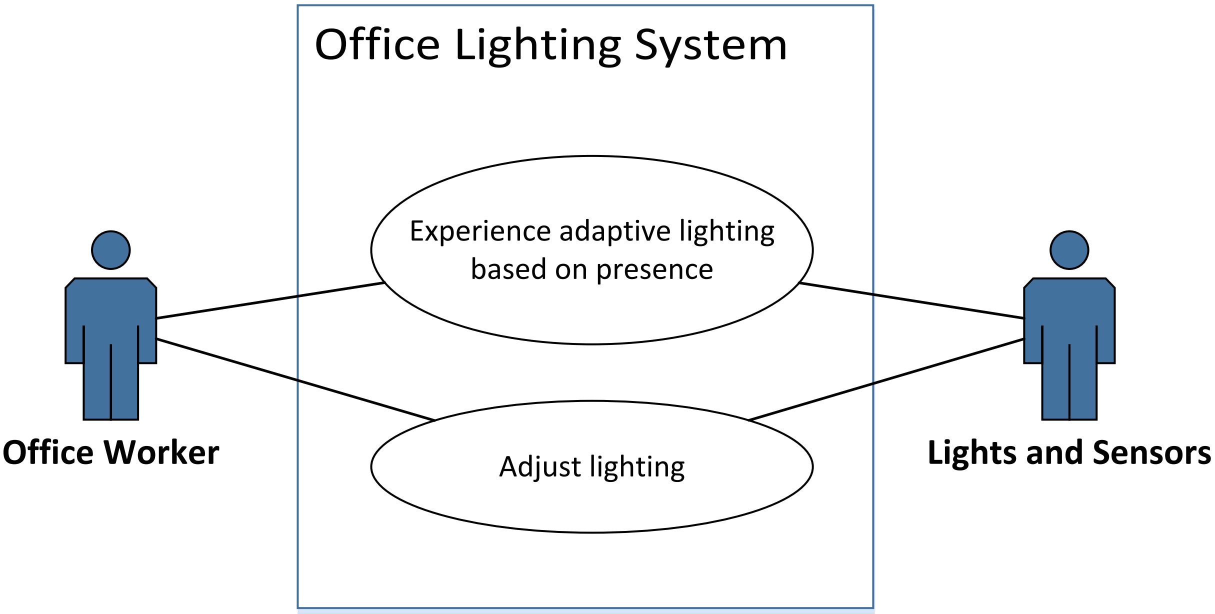


Figure 3 : Use Cases that involve the Office Worker

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)

Object	Object ID	Object Instance ID	Notes	Specification
Light Profile	10250	0	Represent the Light Device's static and dynamic information	Private object

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)

Resource Name	Resource ID	Operation	Type	Value	Notes
Light ID	0	R	String	"Light-Device-GroupNo-LightDeviceNo"	Represents the ID of the light device. For example "Light-Device-1-1" for Light Device 1 of group 1. "Light-Device-25-2" for Light Device 2 of Group 25.
Device Type	1	R	String	"Light Device"	The device type, in this case a "Light Device"
Light State	2	R, W	String	"USED" or "FREE"	Represents the state of the Light Device, whether it is in the "USED" or the "FREE" state
User Type	3	R, W	String	"USER1", "USER2", "USER3"	Represents the type of user that is using the Light Device, whether it is "USER1", "USER2" or "USER3"
User ID	4	R, W	String	"Office-Worker-GroupNo"	The office worker who is using (taking ownership) of the Light Device. Example: "Office-Worker-25".
Light Color	5	R, W	String	"(r, g, b)"	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)"
Low Light	6	R, W	Boolean	True, False	Represents the current color intensity on the Sense Hat's LED matrix. True = Low light mode on, False = Low light mode off

Group No	7	R, W	Integer		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.
Location X	8	R, W	Float		X location of the Light Device (approximation value in meter unit) relative to a reference point (0, 0) in the room.
Location Y	9	R, W	Float		Y location of the Light Device (approximation value in meter unit) relative to a reference point (0, 0) in the room.
Room ID	10	R, W	String	"Room-No"	The ID of the room where the Light Device is located. For example "Room-1", "Room-2", etc
Behavior Deployment	11	R, W	String	"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.
Ownership Priority	12	W	String	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.
Light Behavior	13	W	String	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

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`

```
[{
  "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": 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)

Object	Object ID	Object Instance ID	Notes	Specification
Sensor Profile	10350	0	Represent the Sensor Device’s static and dynamic information	Private Object

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)

Resource Name	Resource ID	Opera-tion	Type	Value	Notes
Sensor ID	0	R	String	“Sensor-Device-GroupNo-SensorDeviceNo”	Represents the ID of the Sensor device. For example “Sensor-Device-1-1” for Sensor Device 1 of group 1. “Sensor-Device-25-2” for Sensor Device 2 of Group 25.
Device Type	1	R	String	“Sensor Device”	The device type, in this case a “Sensor Device”
Sensor State	2	R, W	String	“OCCUPIED” or “FREE”	Represents the state of the Sensor Device, whether it is in the “OCCUPIED” or the “FREE” state
User ID	3	R, W	String	“Office-Worker-GroupNo”	The office worker who is sitting in front of the Sensor Device. Example: “Office-Worker-25”.
Group No	4	R, W	Integer		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.
Location X	5	R, W	Float		X location of the Sensor Device (approximation value in meter unit) relative to a reference point (0, 0) in the room.
Location Y	6	R, W	Float		Y location of the Sensor Device (approximation value in meter unit) relative to a reference point (0, 0) in the room.
Room ID	7	R, W	String	“Room-No”	The ID of the room where the Sensor Device is located. For example “Room-1”, “Room-2”, etc

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.

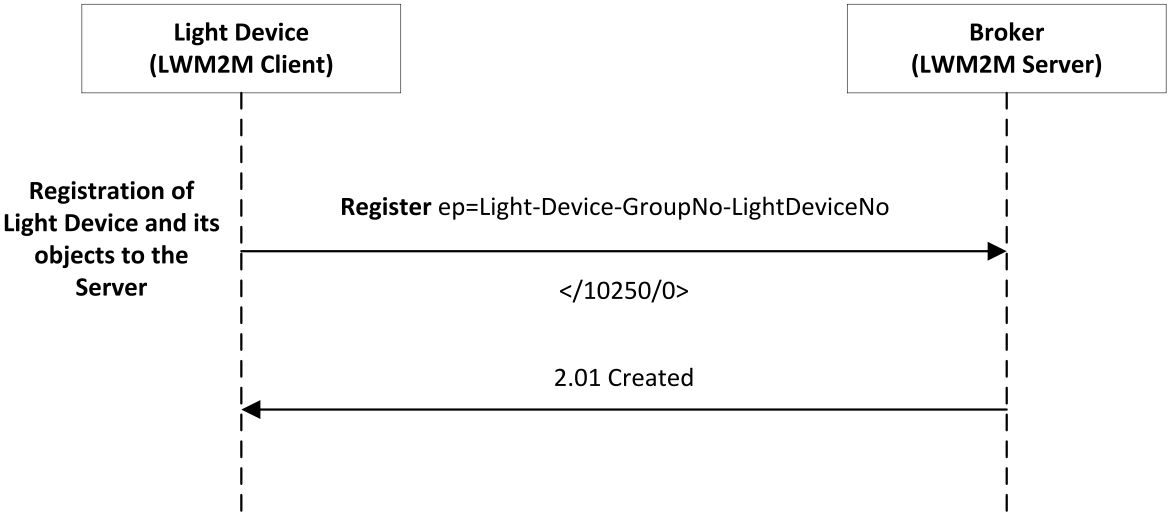

Light Device
(LWM2M Client)
Broker
(LWM2M Server)
Registration of Light Device and its objects to the Server
Register ep=Light-Device-GroupNo-LightDeviceNo
</10250/0>
2.01 Created

Figure 4 : Interaction between the Light Device and LWM2M server in the broker during "Execute the System" use case

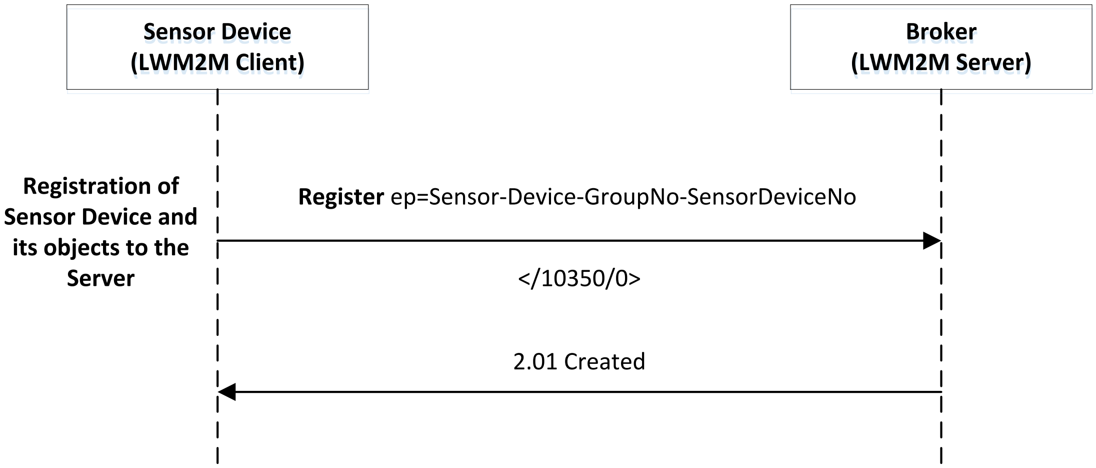

Sensor Device
(LWM2M Client)
Broker
(LWM2M Server)
Registration of Sensor Device and its objects to the Server
Register ep=Sensor-Device-GroupNo-SensorDeviceNo
</10350/0>
2.01 Created

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.

No	Data	Format	Example
1	UserID	"Office-Worker-GroupNo"	"Office-Worker-25"
2	GroupNo	Integer	25
3	RoomID	"Room-No"	"Room-1"
4	Name	"Last Name, First Name"	"Rahman, L.F."
5	Email	Email format	"L.F.Rahman@tue.nl"
6	Password	"pwd-GroupNo"	"pwd-25"

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

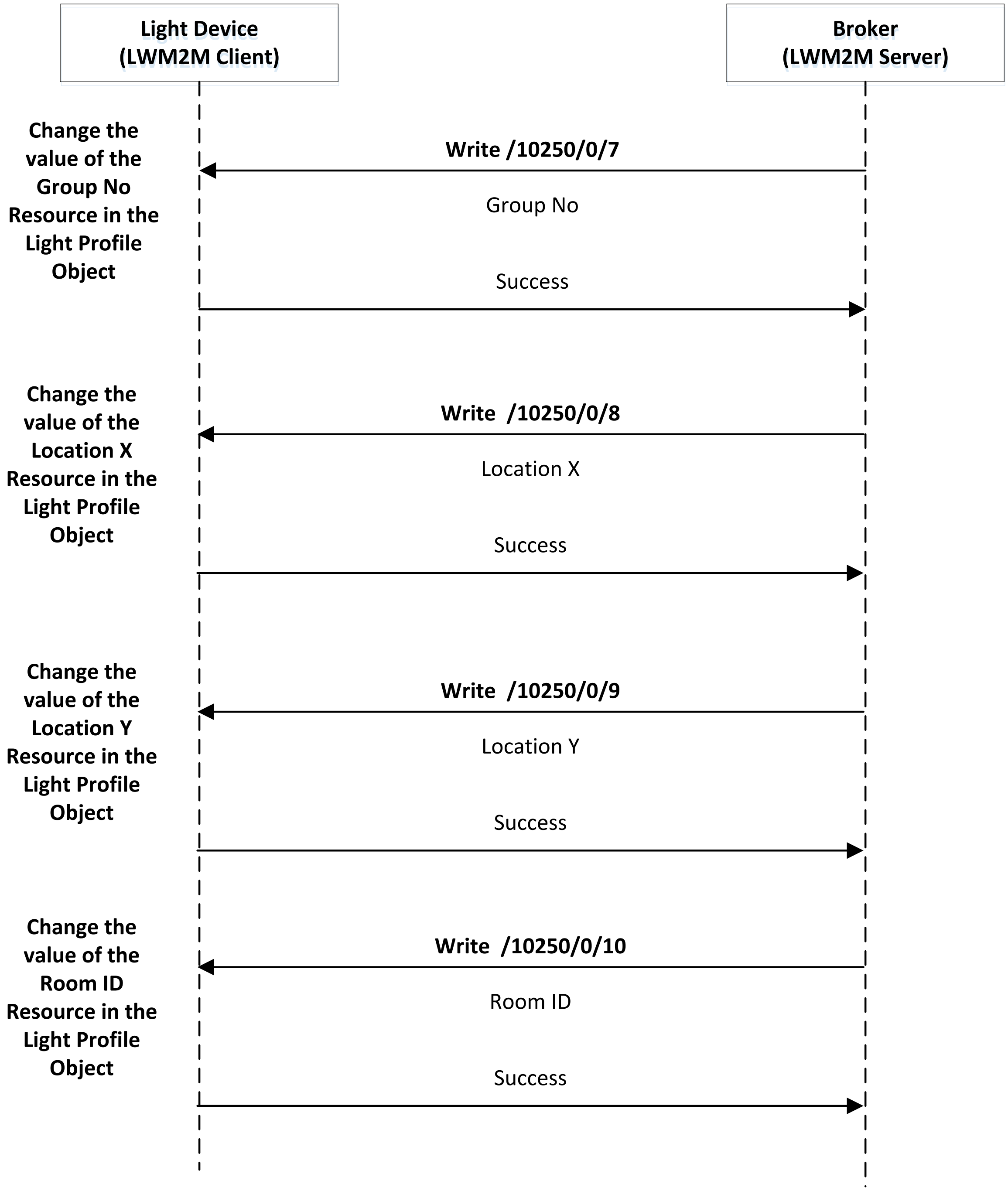

Light Device
(LWM2M Client)
Broker
(LWM2M Server)
Change the value of the Group No Resource in the Light Profile Object
Write /10250/0/7
Group No
Success
Change the value of the Location X Resource in the Light Profile Object
Write /10250/0/8
Location X
Success
Change the value of the Location Y Resource in the Light Profile Object
Write /10250/0/9
Location Y
Success
Change the value of the Room ID Resource in the Light Profile Object
Write /10250/0/10
Room ID
Success

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

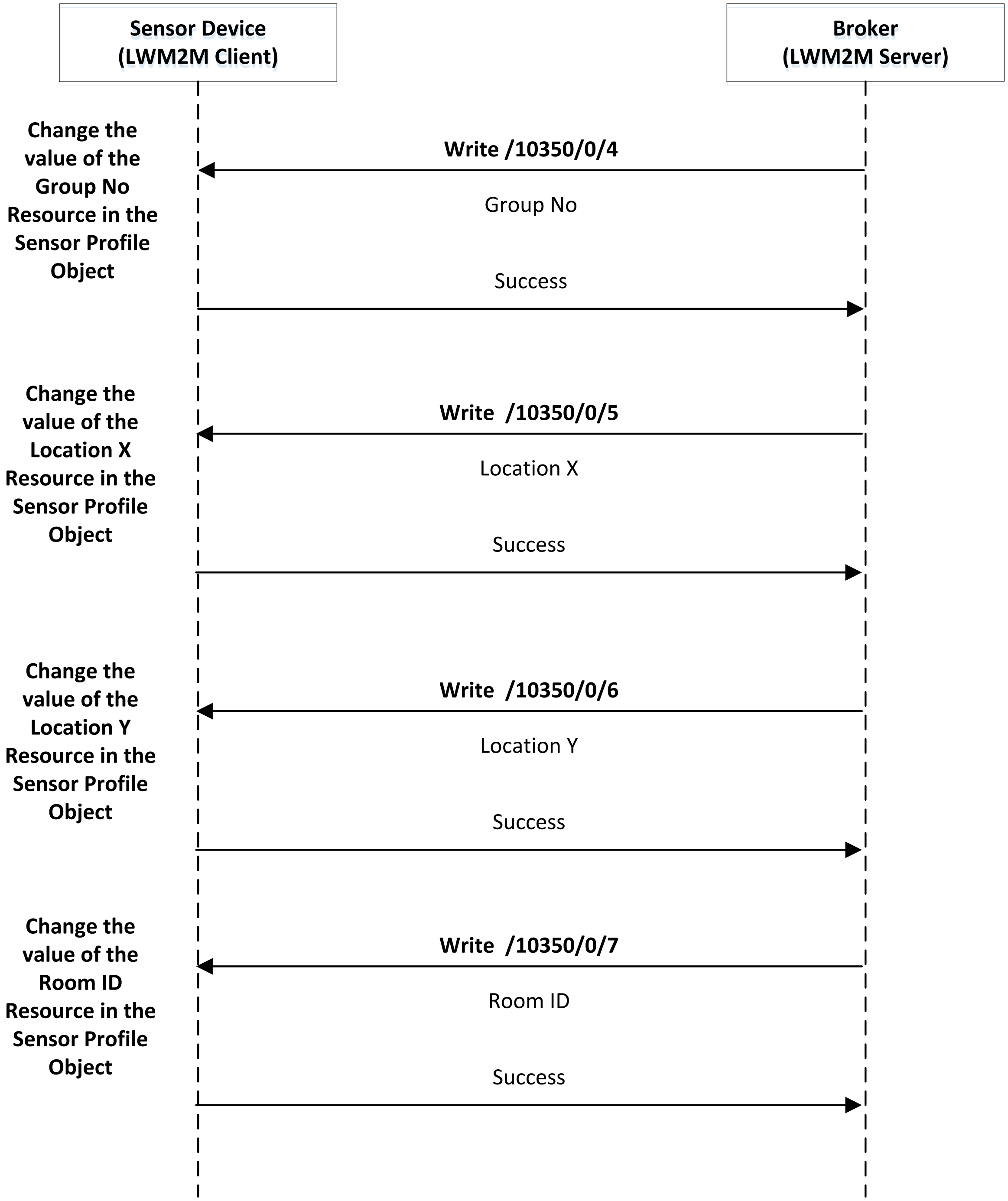

Sensor Device
(LWM2M Client)
Broker
(LWM2M Server)
Change the value of the Group No Resource in the Sensor Profile Object
Write /10350/0/4
Group No
Success
Change the value of the Location X Resource in the Sensor Profile Object
Write /10350/0/5
Location X
Success
Change the value of the Location Y Resource in the Sensor Profile Object
Write /10350/0/6
Location Y
Success
Change the value of the Room ID Resource in the Sensor Profile Object
Write /10350/0/7
Room ID
Success

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

Set Ownership Priority

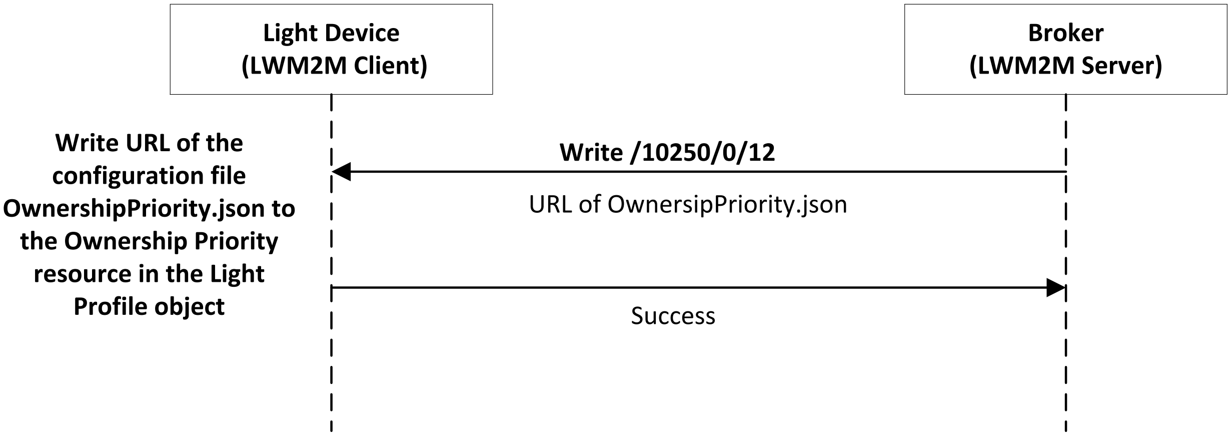
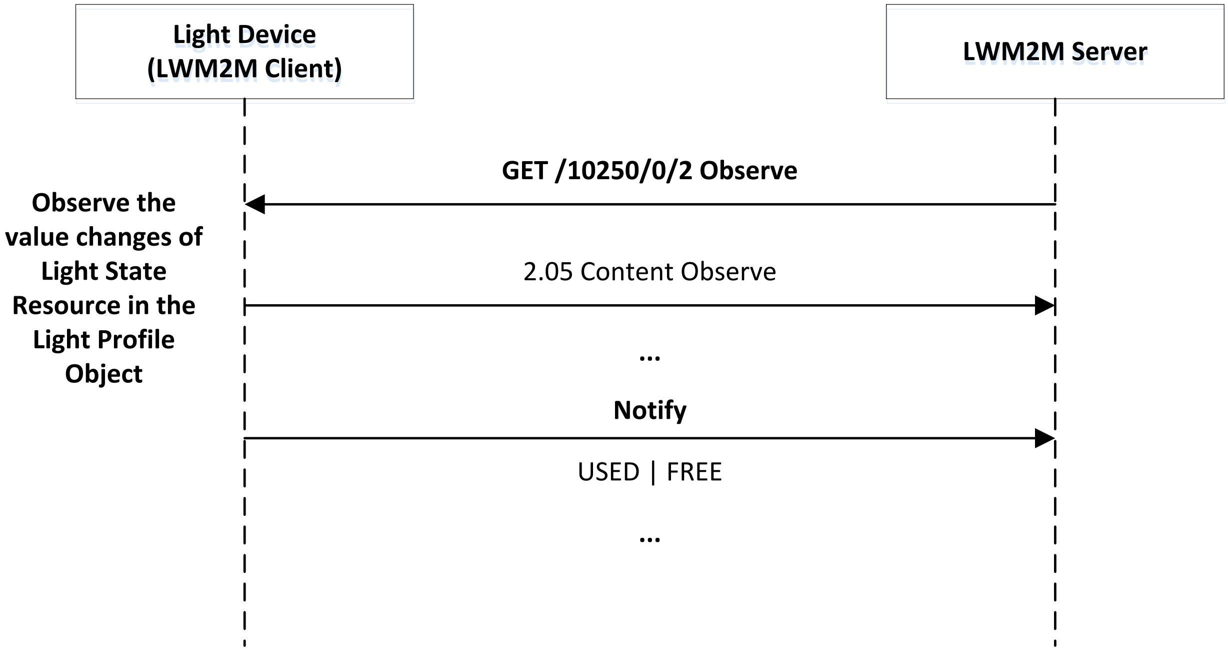
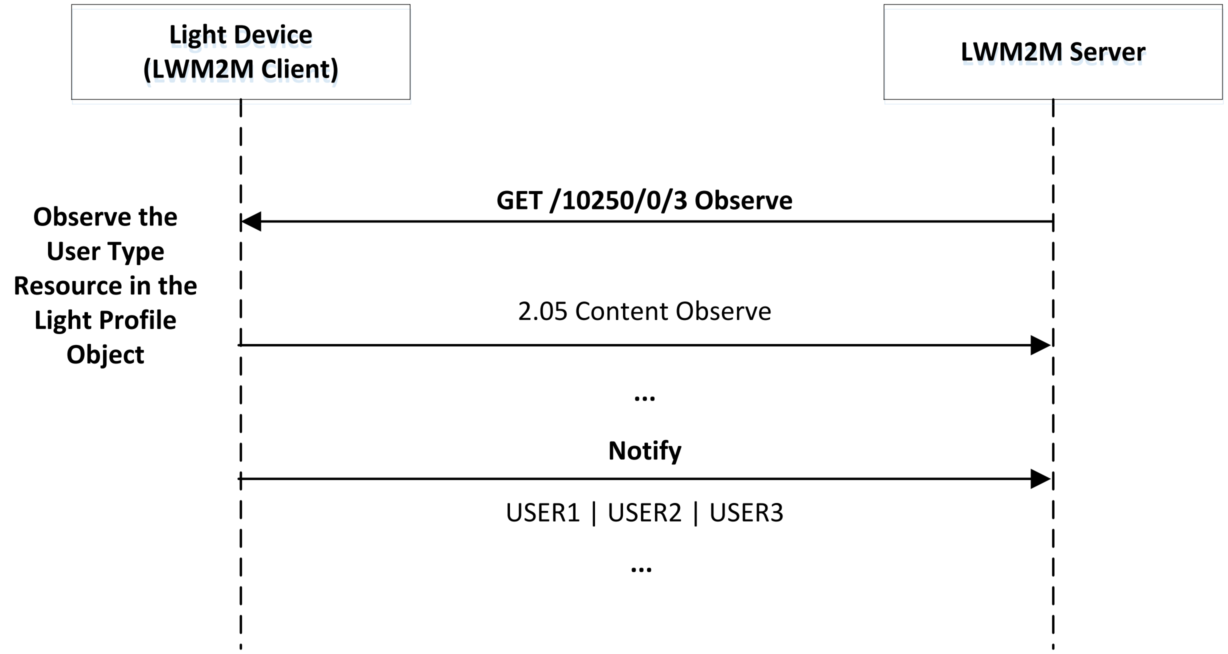
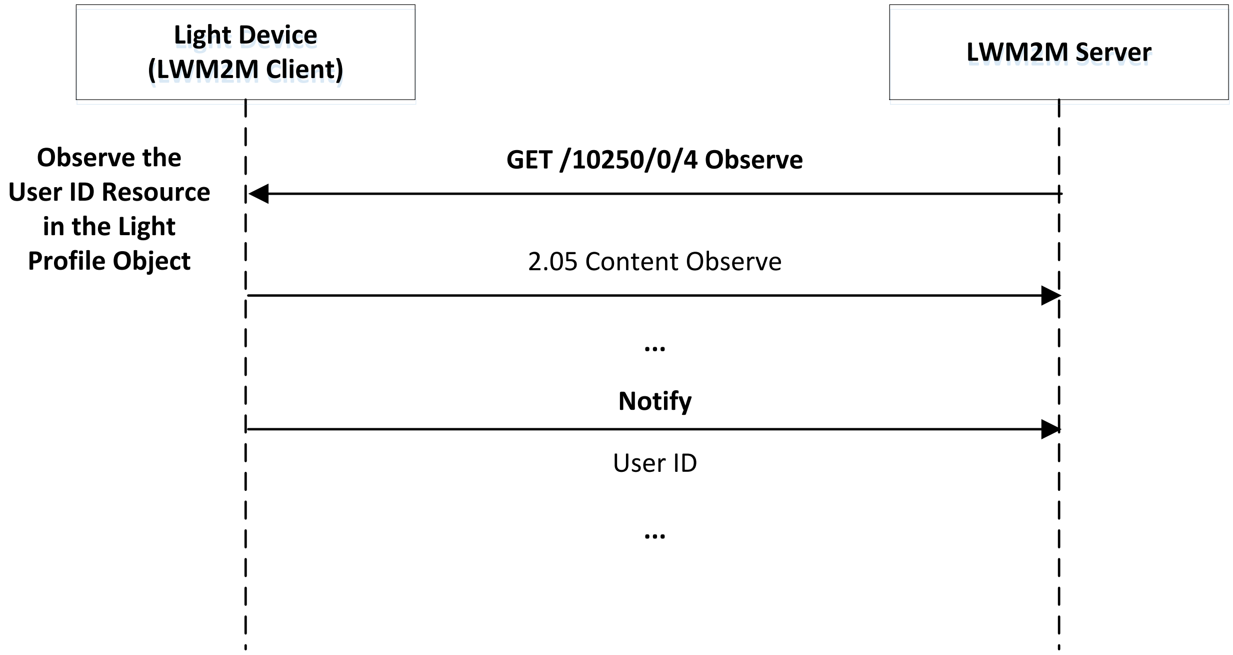

Light Device
(LWM2M Client)
Broker
(LWM2M Server)
Write URL of the configuration file OwnershipPriority.json to the Ownership Priority resource in the Light Profile object
Write /10250/0/12
URL of OwnersipPriority.json
Success

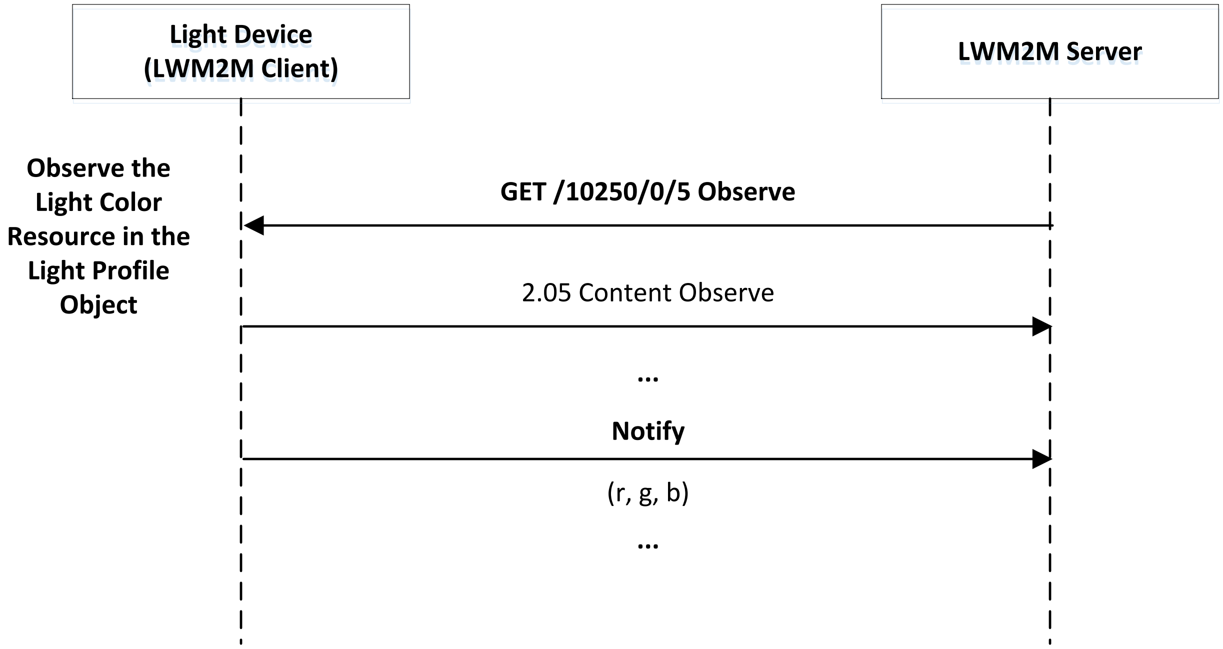
Figure 8 : Interaction between the Light Device and LWM2M server in the broker during "Set Priority Ownership" use case

Observe State




Light Device
(LWM2M Client)
LWM2M Server
GET /10250/0/3 Observe
Observe the User Type Resource in the Light Profile Object
2.05 Content Observe
...
Notify
USER1 | USER2 | USER3
...


Light Device
(LWM2M Client)
LWM2M Server
GET /10250/0/4 Observe
Observe the User ID Resource in the Light Profile Object
2.05 Content Observe
...
Notify
User ID
...



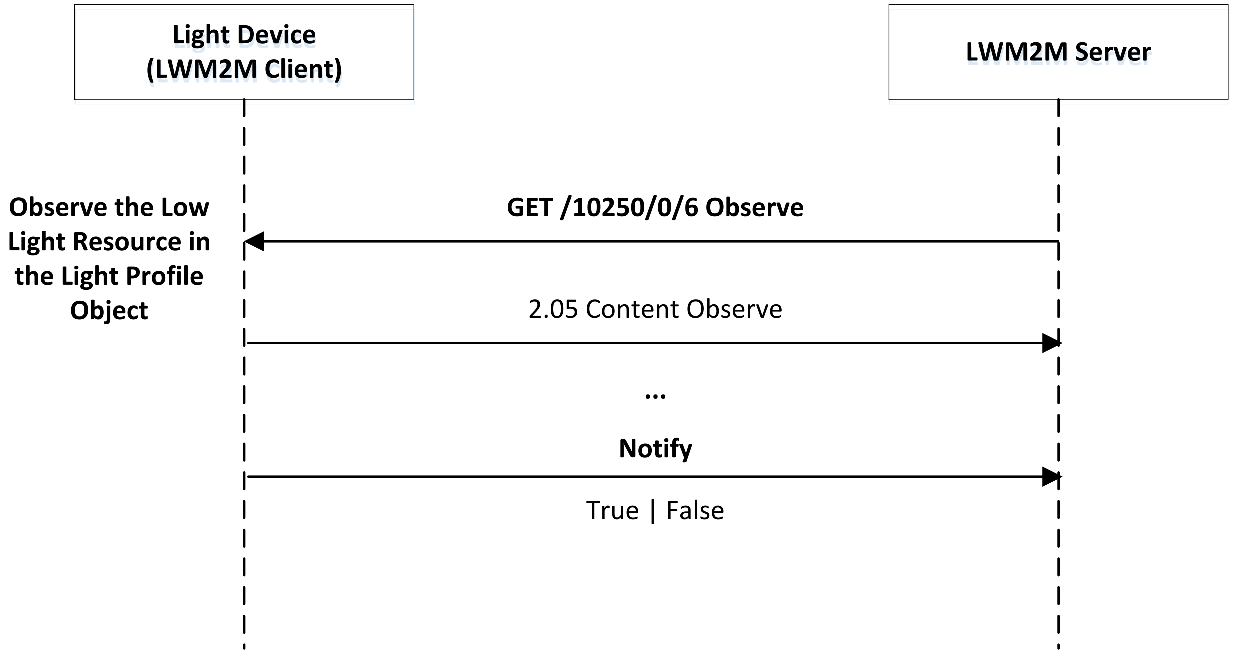


Figure 9 : Interaction between the Ligthing Device and LWM2M server in the broker during “Observe State” use case

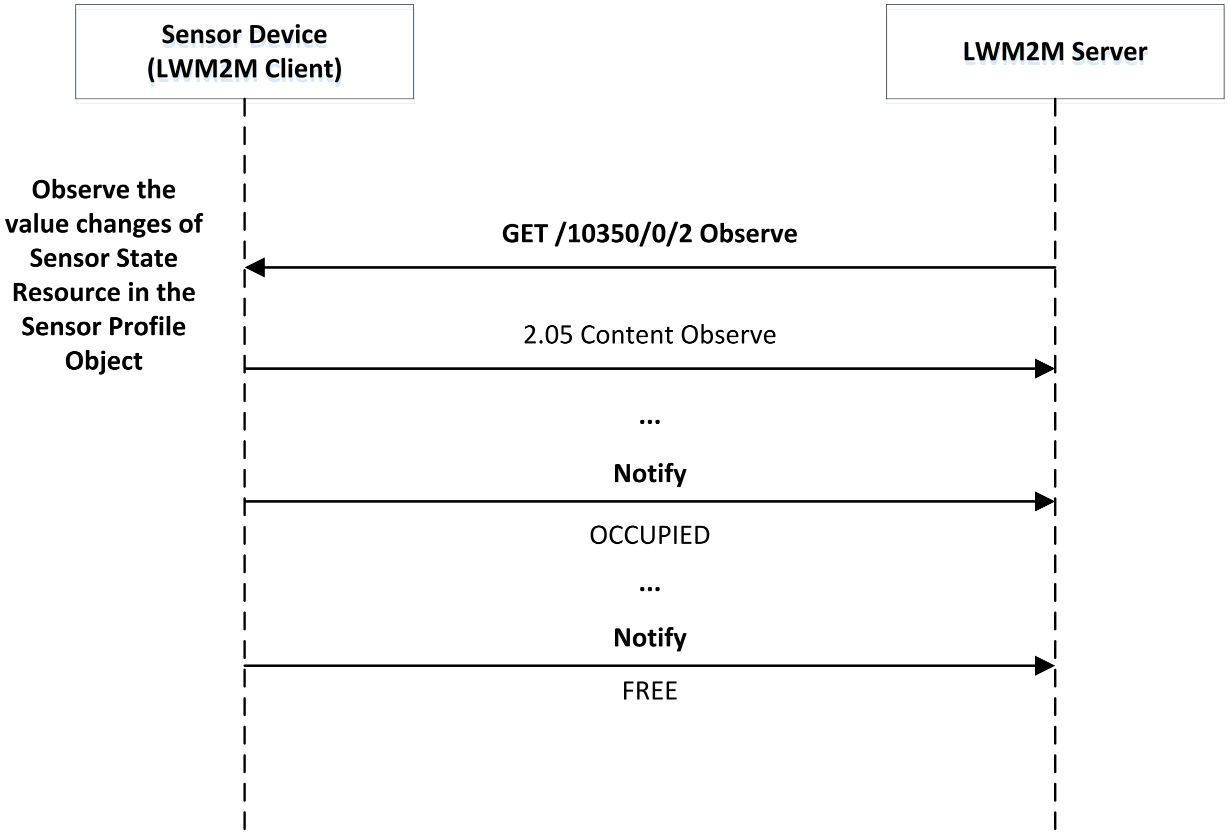

Sensor Device (LWM2M Client)
LWM2M Server
Observe the value changes of Sensor State Resource in the Sensor Profile Object
GET /10350/0/2 Observe
2.05 Content Observe
...
Notify
OCCUPIED
...
Notify
FREE

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

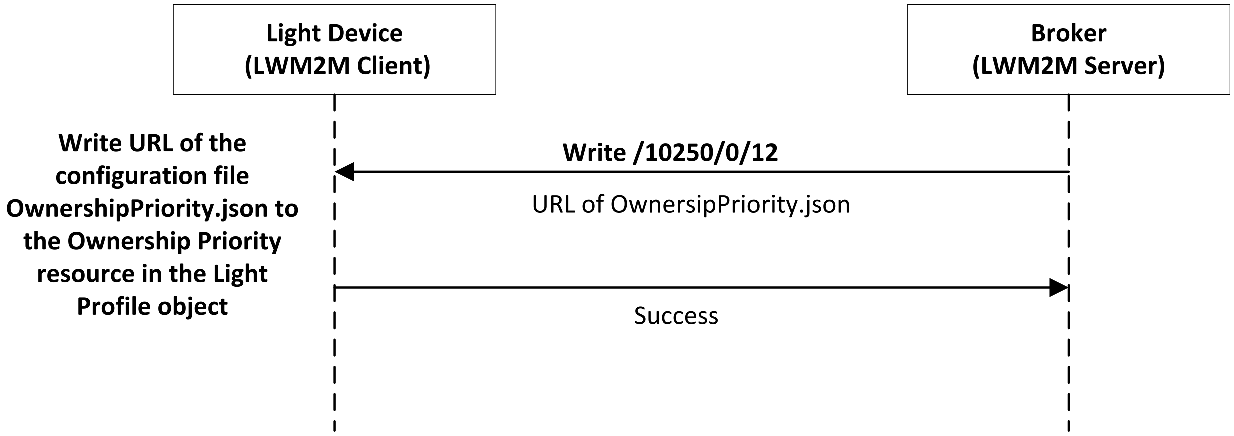

Light Device
(LWM2M Client)
Broker
(LWM2M Server)
Write URL of the configuration file OwnershipPriority.json to the Ownership Priority resource in the Light Profile object
Write /10250/0/12
URL of OwnersipPriority.json
Success

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.

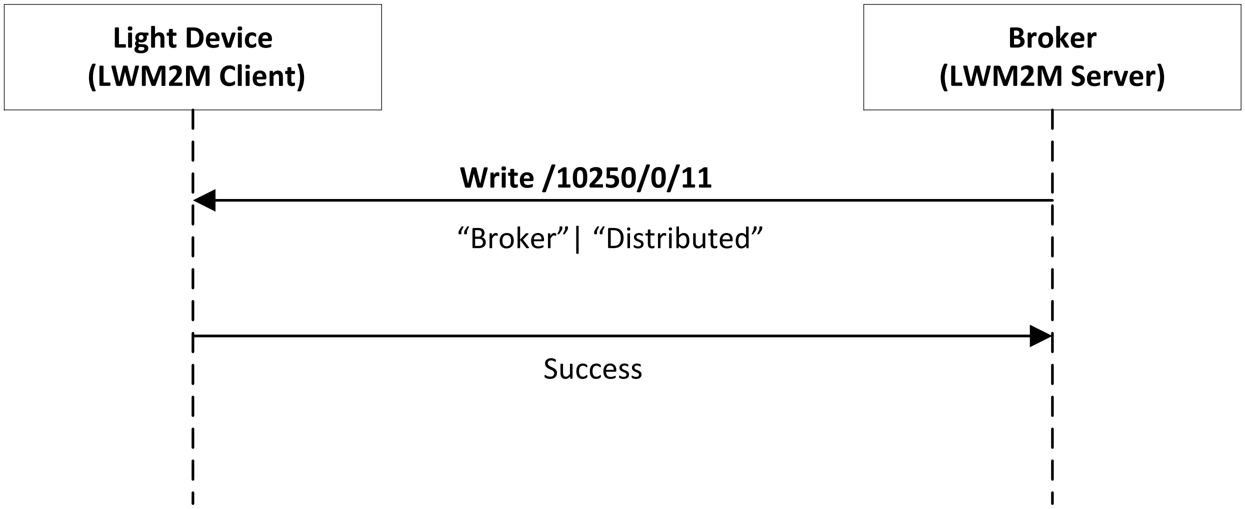


Figure 12 : Interaction between the Light Device and LWM2M server in the broker during “Set Behavior Deployment” use case

Update Light Behavior

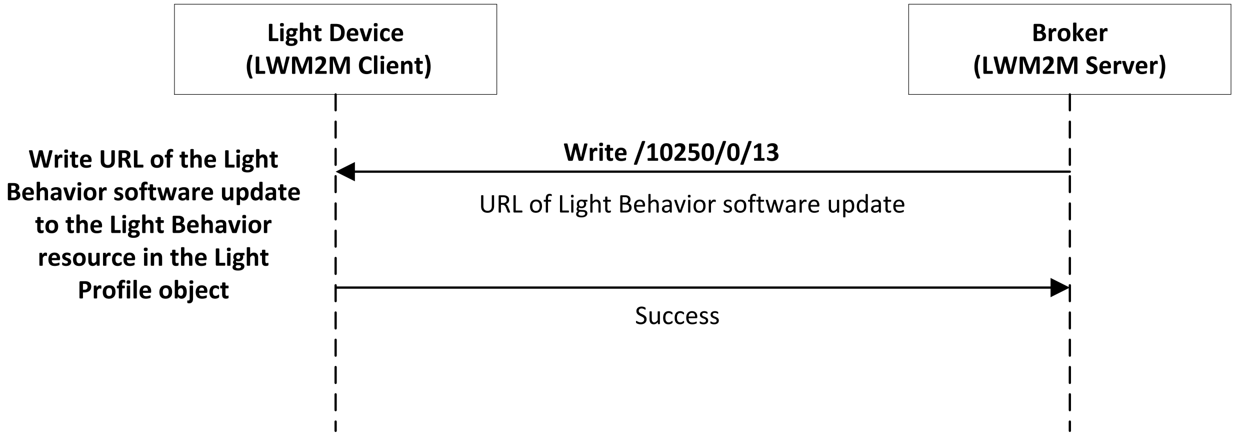

Light Device
(LWM2M Client)
Broker
(LWM2M Server)
Write URL of the Light Behavior software update to the Light Behavior resource in the Light Profile object
Write /10250/0/13
URL of Light Behavior software update
Success

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

Adjust Lighting

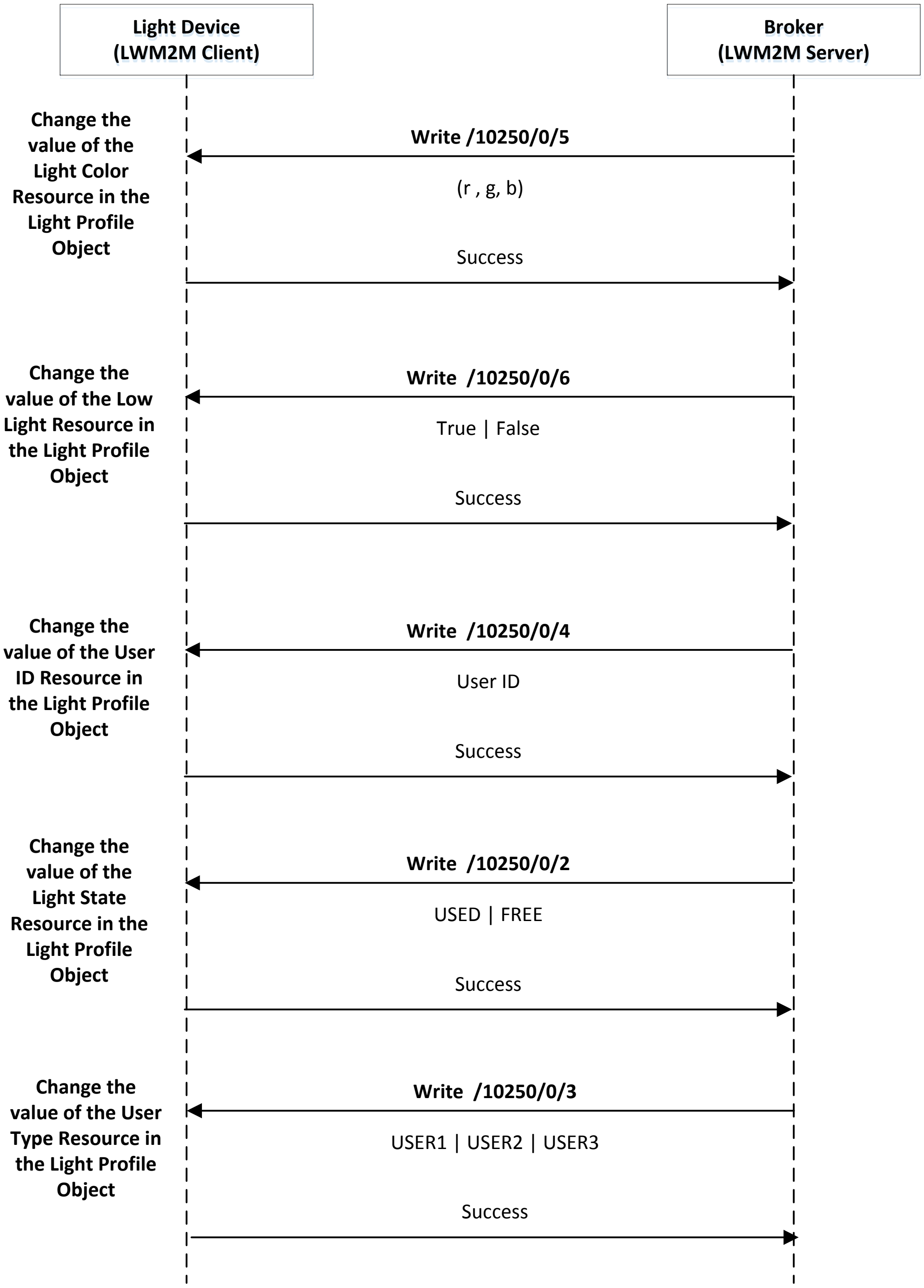

Light Device
(LWM2M Client)
Broker
(LWM2M Server)
Change the value of the Light Color Resource in the Light Profile Object
Write /10250/0/5
(r , g, b)
Success
Change the value of the Low Light Resource in the Light Profile Object
Write /10250/0/6
True | False
Success
Change the value of the User ID Resource in the Light Profile Object
Write /10250/0/4
User ID
Success
Change the value of the Light State Resource in the Light Profile Object
Write /10250/0/2
USED | FREE
Success
Change the value of the User Type Resource in the Light Profile Object
Write /10250/0/3
USER1 | USER2 | USER3
Success

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 deploymentent. Figure 15 shows interaction example between the Sensor Device and the MQTT broker.

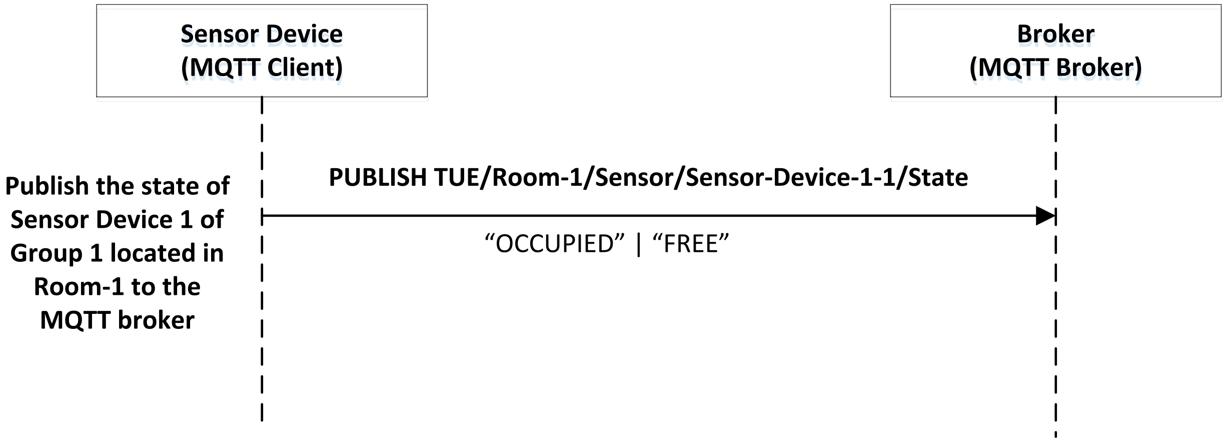

Sensor Device
(MQTT Client)
Broker
(MQTT Broker)
Publish the state of Sensor Device 1 of Group 1 located in Room-1 to the MQTT broker
PUBLISH TUE/Room-1/Sensor/Sensor-Device-1-1/State
“OCCUPIED” | “FREE”

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.

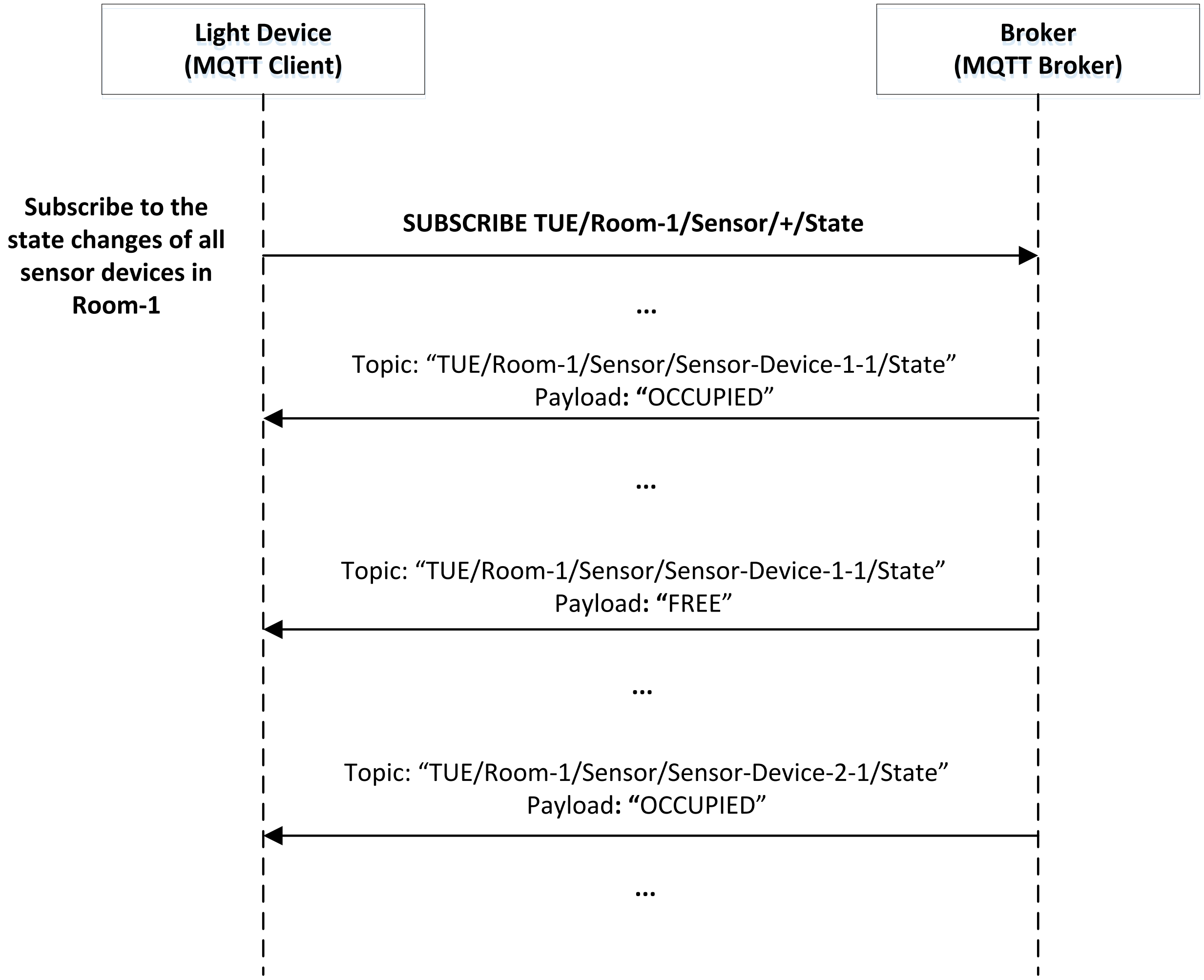

Light Device
(MQTT Client)
Broker
(MQTT Broker)
Subscribe to the state changes of all sensor devices in Room-1
SUBSCRIBE TUE/Room-1/Sensor/+/State
...
Topic: "TUE/Room-1/Sensor/Sensor-Device-1-1/State"
Payload: "OCCUPIED"
...
Topic: "TUE/Room-1/Sensor/Sensor-Device-1-1/State"
Payload: "FREE"
...
Topic: "TUE/Room-1/Sensor/Sensor-Device-2-1/State"
Payload: "OCCUPIED"
...

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.

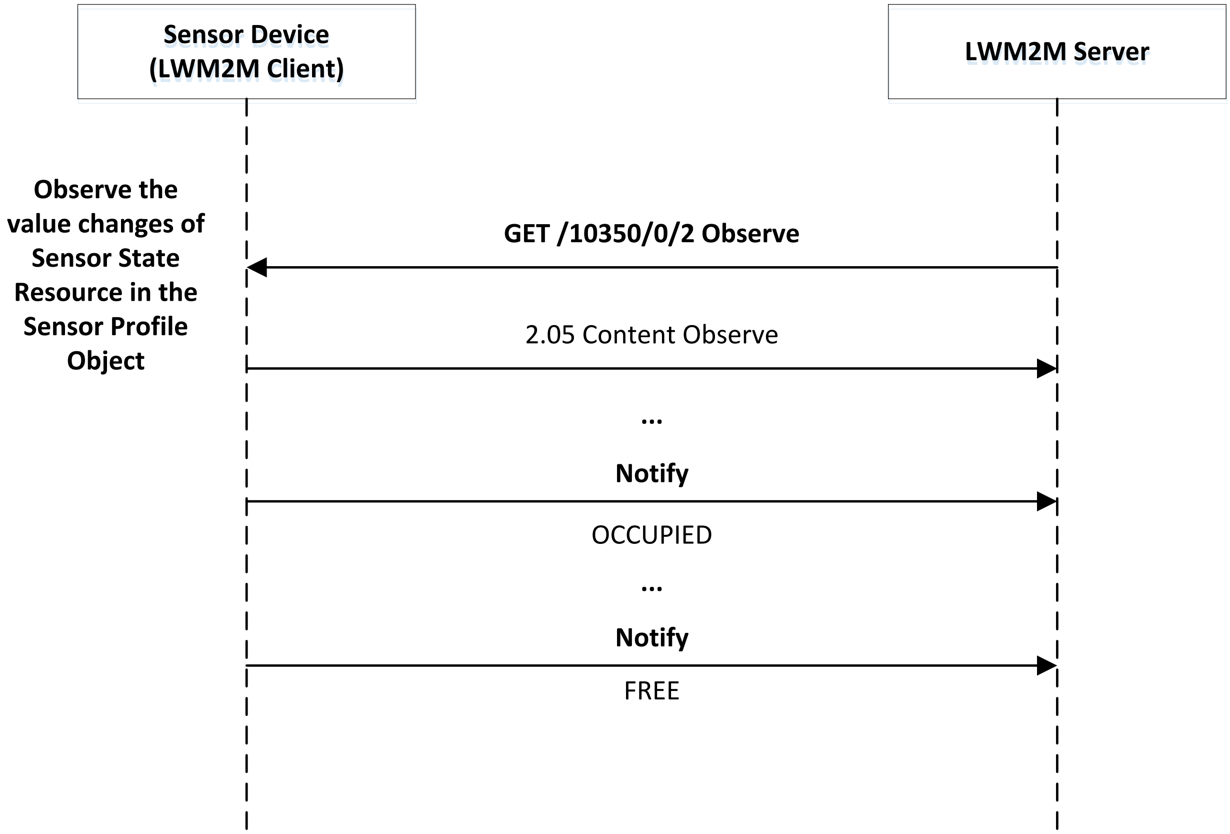


Figure 17 : Interaction between the Light Device and LWM2M server in the broker during "Adaptive Lighting Based on Presence" use case if using centralized behavior deployment

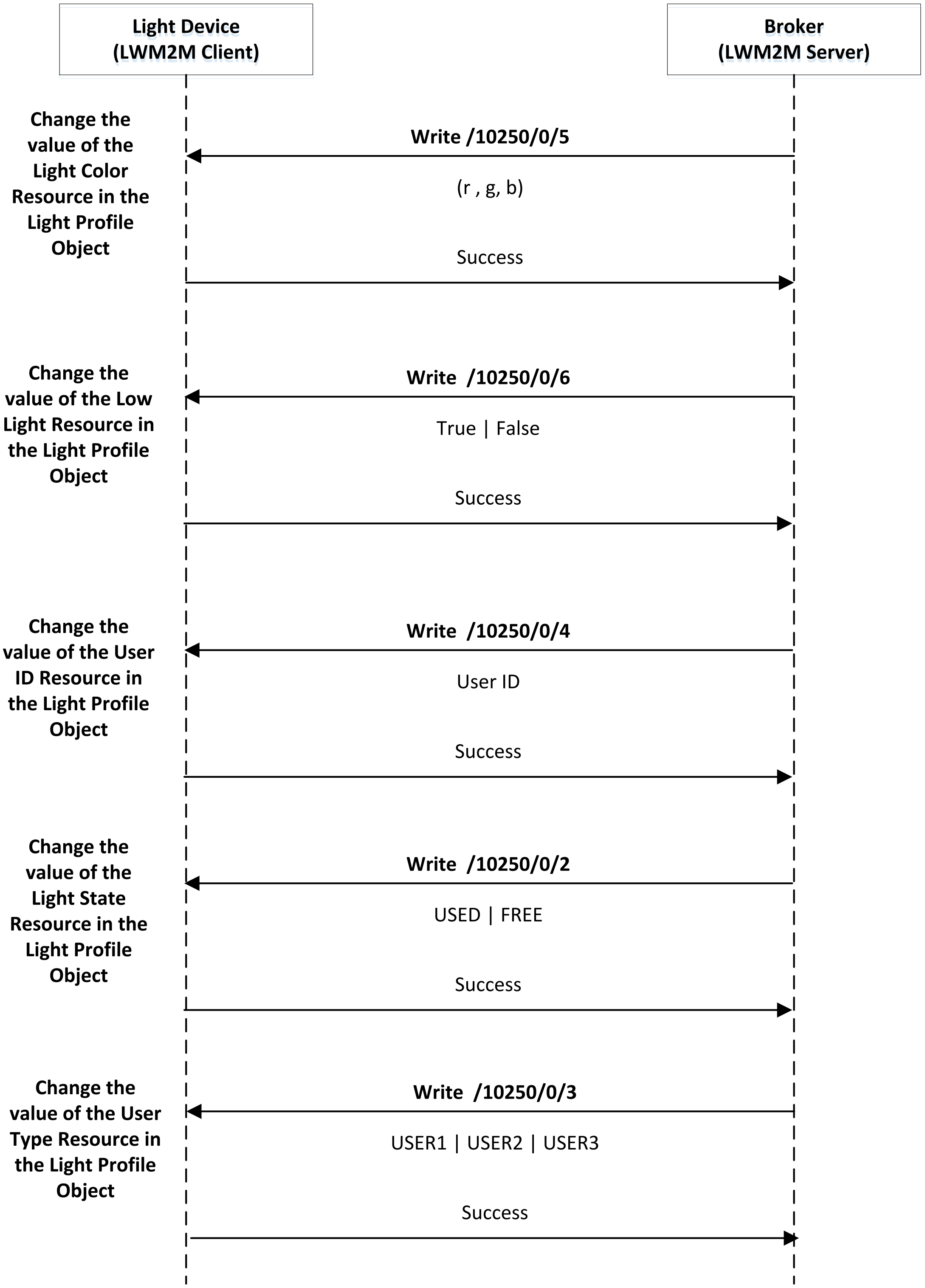

Light Device
(LWM2M Client)
Broker
(LWM2M Server)
Change the value of the Light Color Resource in the Light Profile Object
Write /10250/0/5
(r , g, b)
Success
Change the value of the Low Light Resource in the Light Profile Object
Write /10250/0/6
True | False
Success
Change the value of the User ID Resource in the Light Profile Object
Write /10250/0/4
User ID
Success
Change the value of the Light State Resource in the Light Profile Object
Write /10250/0/2
USED | FREE
Success
Change the value of the User Type Resource in the Light Profile Object
Write /10250/0/3
USER1 | USER2 | USER3
Success

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