IoT Practical Instruction
Office Lighting System

Leila F. Rahman

Eindhoven, 15 December 2016

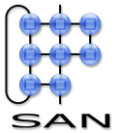
SAN

TU/e Technische Universiteit Eindhoven University of Technology

Where innovation starts

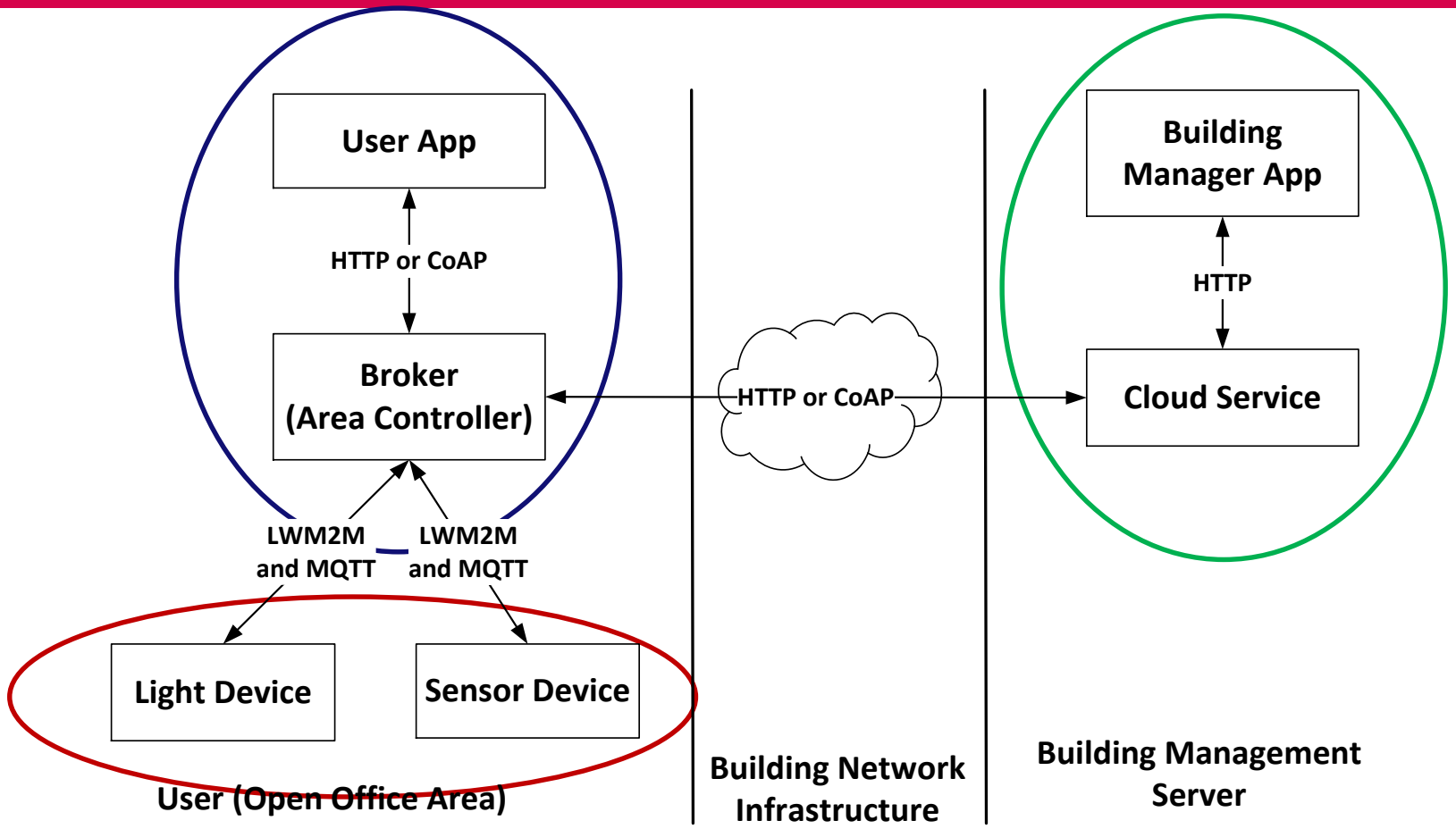
Agenda

- **Protocol description**
- **Tutorials run-through**

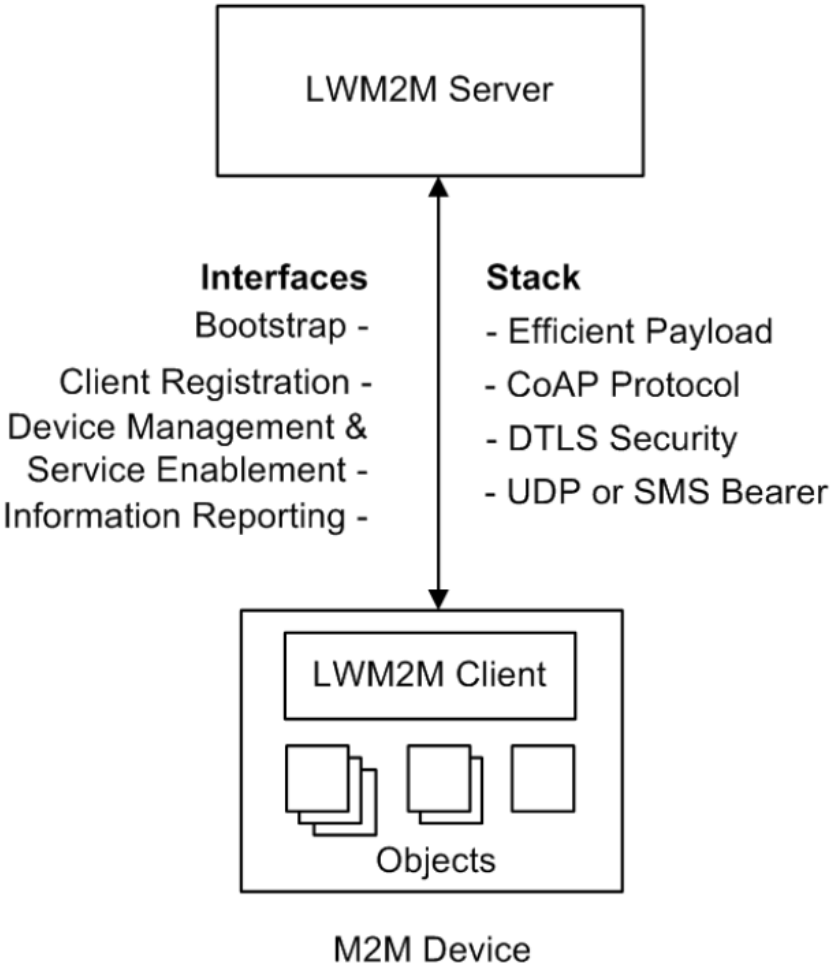


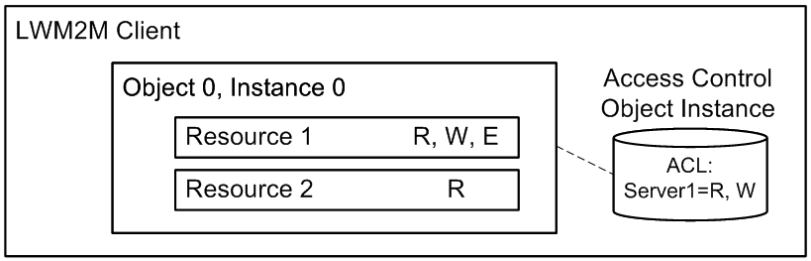
TU/e Technische Universiteit Eindhoven University of Technology

System Deployment

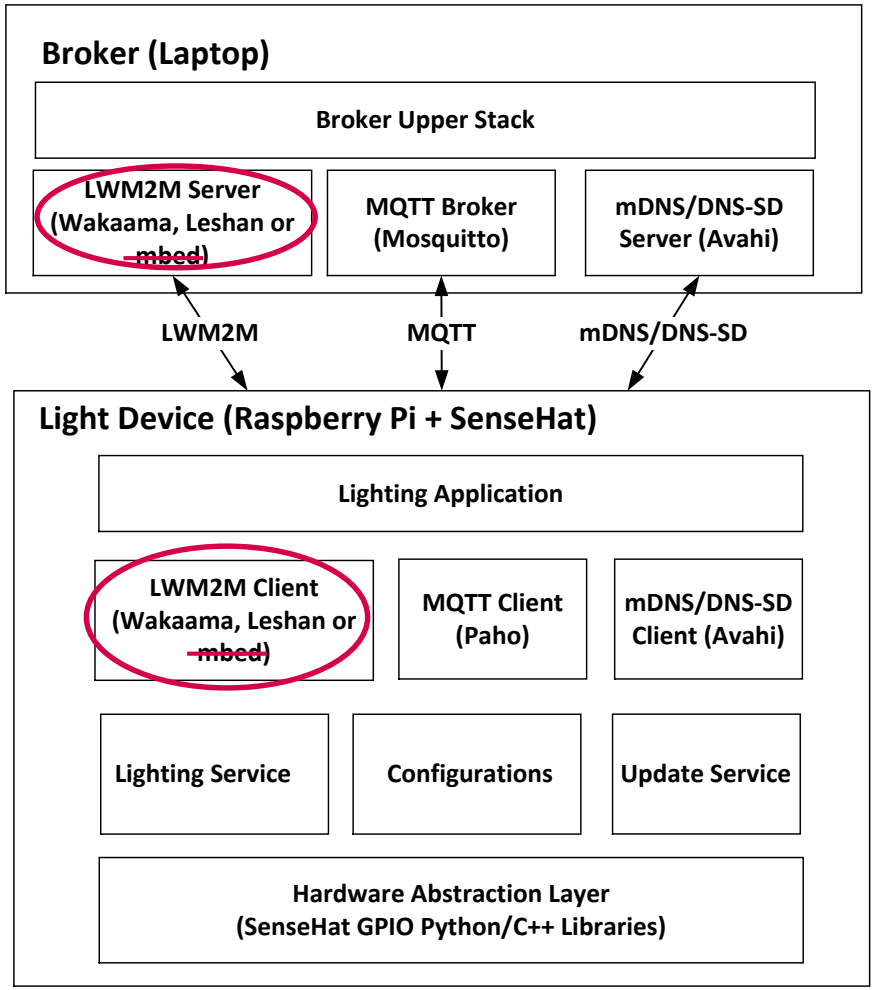


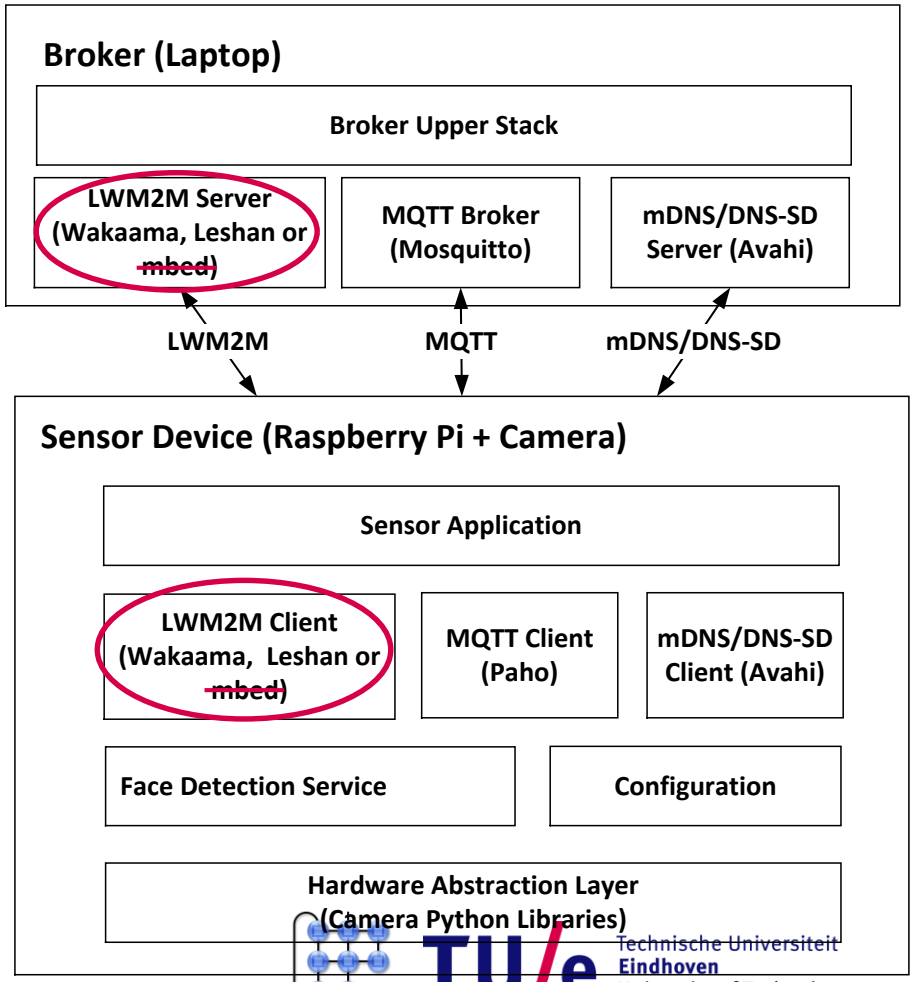
LWM2M Overview





LWM2M in Broker and End Devices





LWM2M Objects

- LWM2M Objects can be acquired from:
  - LWM2M objects defined by OMA (Appendix E of LWM2M Technical Specification)
  - IPSO smart object starter pack (http://www.win.tue.nl/~johanl/educ/2IMN15/local/IPSO-Smart-Objects-Starter-Pack.pdf)
  - IPSO smart object expansion pack (http://www.ipso-alliance.org/ipso-community/resources/smart-objects-interoperability/)
  - Private objects by using free range id 10241 – 32768 (http://technical.openmobilealliance.org/Technical/technical-information/omna/lightweight-m2m-lwm2m-object-registry)

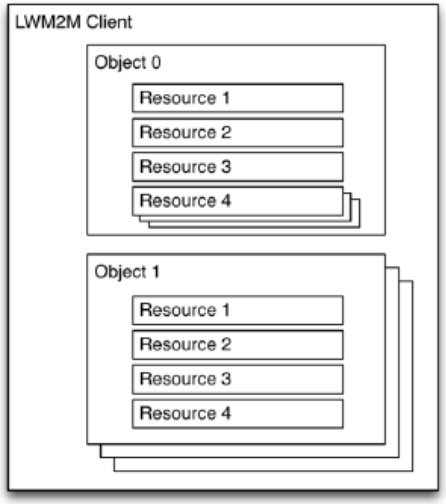


Figure 13: Relationship between LWM2M Client, Object, and Resources

OMA LWM2M Objects

Registered with Open Mobile Naming Authority

Object	Object ID
LWM2M Security	0
LWM2M Server	1
Access Control	2
Device	3
Connectivity Monitoring	4
Firmware	5
Location	6
Connectivity Statistics	7

E.6 LWM2M Object: Firmware Update

Description

This LWM2M Object enables management of firmware which is to be updated. This Object includes installing firmware package, updating firmware, and performing actions after updating firmware. A reboot of the device must occur for taking into account the new successfully installed firmware.

After reboot of the device:

- the "State" Resource must be at Downloaded state (2) if the "Package" Resource contains a valid Package which has not been successfully installed yet, or at Idle state (0) otherwise.
- the Update Result must maintain the relevant value it has before Device reboot.

Object definition

Name	Object ID	Instances	Mandatory	Object URN
Firmware Update	5	Single	Optional	urn:oma:lwm2m:oma:5

Resource definitions

ID	Name	Operations	Instances	Mandatory	Type	Range or Enumeration	Units	Description
0	Package	W	Single	Mandatory	Opaque			Firmware package
1	Package URI	W	Single	Mandatory	String	0-255 bytes		URI from where the device can download the firmware package by an alternative mechanism. As soon the device has received the Package URI it performs the download at the next practical opportunity.
2	Update	E	Single	Mandatory	none	no argument		Updates firmware by using the firmware package stored in Package, or, by using the firmware downloaded from the Package URI. This Resource is only executable when the value of the State Resource is Downloaded.

Object and Resource Examples

- /3/0/4Device, Reboot
- /3/0/5Device, Factory Reset
- /3/0/9Device, Battery Level
- /5/0/1Firmware Update, Package
- /5/0/2Firmware Update, Update
- /6/0/0Location, Latitude
- /7/0/2Connectivity Statistics, Tx Data

- IPSO examples (see references)

Object and Resource IDs with the IPSO alliance

- IPSO alliance: IP for Smart Objects
- IPSO Smart Objects Starter Pack
  - Set of 18 smart objects for use with CoAP
  - Based on LWM2M object model
  - Not dependent on full LWM2M framework
    - Use CoAP directly
- IPSO Smart Objects Expansion Pack
  - 16 Common Template sensors
    - Pressure, Power, Distance, …
  - 6 Special Template sensors
    - Energy, Color, GPS Location, …
  - 5 Actuators
    - Buzzer, Display, …
  - 6 Control switch types
    - Up/Down, Push button, Multiple axis joystick, …

IPSO Smart Objects Starter Pack

Table 1SmartObjectsdefined by this Technical Guideline

Object	Object ID	Multiple Instances?
IPSO Digital Input	3200	Yes
IPSO Digital Output	3201	Yes
IPSO Analogue Input	3202	Yes
IPSO Analogue Output	3203	Yes
IPSO Generic Sensor	3300	Yes
IPSO Illuminance Sensor	3301	Yes
IPSO Presence Sensor	3302	Yes
IPSO Temperature Sensor	3303	Yes
IPSO Humidity Sensor	3304	Yes
IPSO Power Measurement	3305	Yes
IPSO Actuation	3306	Yes
IPSO Set Point	3308	Yes
IPSO Load Control	3310	Yes
IPSO Light Control	3311	Yes
IPSO Power Control	3312	Yes
IPSO Accelerometer	3313	Yes
IPSO Magnetometer	3314	Yes
IPSO Barometer	3315	Yes

10. IPSO Object: Temperature

Description: This IPSO objectshould be used with a temperature sensor to report a temperature measurement. It also provides resources for minimum/maximum measured values and the minimum/maximum range that can be measured by the temperature sensor. An example measurement unit is degrees Celsius (ucum:Cel).

Object Info:

Object	Object ID	Object URN	Multiple Instances?	Description
IPSO Temperature	3303	urn:oma:lwm2m:ext:3303	Yes	Temperature sensor, example units = Cel

Resources:

Resource Name	Resource ID	Access Type	Multiple Instances?	Mandatory	Type	Range or Enumeration	Units	Descriptions
Sensor Value	5700	R	No	Mandatory	Float			Last or Current Measured Value from the Sensor
Units	5701	R	No	Optional	String			Measurement Units Definition e.g. "Cel" for Temperature in Celsius.
Min	5601	R	No	Optional	Float	Same as	Same as	The minimum

http://www.win.tue.nl/~johanl/educ/2IMN15/local/IPSO-Smart-Objects-Starter-Pack.pdf

SAN

TU/e Technische Universiteit Eindhoven University of Technology

IPSO Smart Objects Expansion Pack

Type	Object	Object ID
Common Template Sensors	Voltage	3316
	Current	3317
	Frequency	3318
	Depth	3319
	Percentage	3320
	Altitude	3321
	Load	3322
	Pressure	3323
	Loudness	3324
	Concentration	3325
	Acidity	3326
	Conductivity	3327
	Power	3328
	Power Factor	3329
	Rate	3346
	Distance	3330
Special Template Sensors	Energy	3331
	Direction	3332
	Time	3333
	Gyrometer	3334
	Color	3335
	GPS Location	3336
Actuators	Positioner	3337
	Buzzer	3338
	Audio Clip	3339
	Timer	3340
	Addressable Text Display	3341
Controls	On/Off Switch	3342
	Push Button	3347
	Level Control	3343
	Up/Down Control	3344
	Multistate Selector	3348
	Multiple Axis Joystick	3345

sources.

Table 1 Summarizes the Objects defined by this Technical Guideline.

Table 1. Smart Objects defined by this Technical Guideline

3

- http://www.ipso-alliance.org/ipso-community/resources/smart-objects-interoperability/

6.6 IPSO Object: Multiple Axis Joystick

Description: This IPSO object can be used to report the position of a shuttle or joystick control. A digital input is provided to report the state of an associated push button.

Object Info:

Object	Object ID	Object URN	Multiple Instances?	Description
Multiple Axis Joystick	3345	urn:oma:lwm2m:ext:3345	Yes	Used for a 1 axis (shuttle) control, 2 axis control, or 3 axis control

Resources:

Resource Name	Resource ID	Access Type	Multiple Instances?	Mandatory	Type	Range or Enumeration	Units	Descriptions
Digital Input State	5500	R	No	Optional	Boolean			The current state of a digital input.
Digital Input Counter	5501	R	No	Optional	Integer			The number of times the input transitions from 0 to 1
X Value	5702	R	No	Optional	Float	-100 +100	%	The measured value along the X axis.
Y Value	5703	R	No	Optional	Float	-100 +100	%	The measured value along the Y axis.
Z Value	5704	R	No	Optional	Float	-100 +100	%	The measured value along the Z axis.
Application Type	5750	R,W	No	Optional	String			The application type of the sensor or actuator as a string depending on the use case.

SAN

TU/e Technische Universiteit Eindhoven University of Technology

Object Instances in Light Device

Object	Object ID	Object Instance ID	Notes	Specification
Light Profile	10250	0	Represent the Light Device's static and dynamic information	Private object
Firmware Update	5	0	Represents the Update for the Lighting Behavior Component of the Light Device	OMA LWM2M Objects
Firmware Update	5	1	Represents the Update for Configuration File about Ownership Priority of the Light Device	OMA LWM2M Objects

Resources in the *Light Profile* Object (Object Id: 10250) (cont'd)

Resource Name	Resource ID	Opera-tion	Type	Value	Notes
Light ID	0	R, W	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, W	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. Examp: "Office-Worker-25".
Light Color	5	R, W	String	"(r, g, b)"	Represents the 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)

Resources in the *Light Profile* Object (Object Id: 10250)

Resource Name	Resource ID	Opera-tion	Type	Value	Notes
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 assign to a group of light(s) and a sensor.
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	Boolean	"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.

Resources in the *Firmware Update* object (Object Id: 5)

Resource Name	Resource ID	Opera-tion	Type	Value	Notes
Package	0	W	Opaque		Not used
PackageURI	1	W	String	0-255 bytes	The URL of the software update. Download and execute the update after value is changed.
Update	2	E	none		Not used
State	3	R	Integer	0-3	Indicates current state with respect to this firmware update. This value is set by the LWM2M client. 0: Idle 1: Downloading 3: Updating (Refer to LWM2M TS for more detail information)
Update Supported Objects	4	RW	Boolean		Not used
Update Result	5	R	Integer	0-6	Contains the result of downloading or updating the firmware. (Refer to LWM2M TS for more detail information)
PkgName	6	R	String		Not used
PkgVersion	7	R	String		Not used

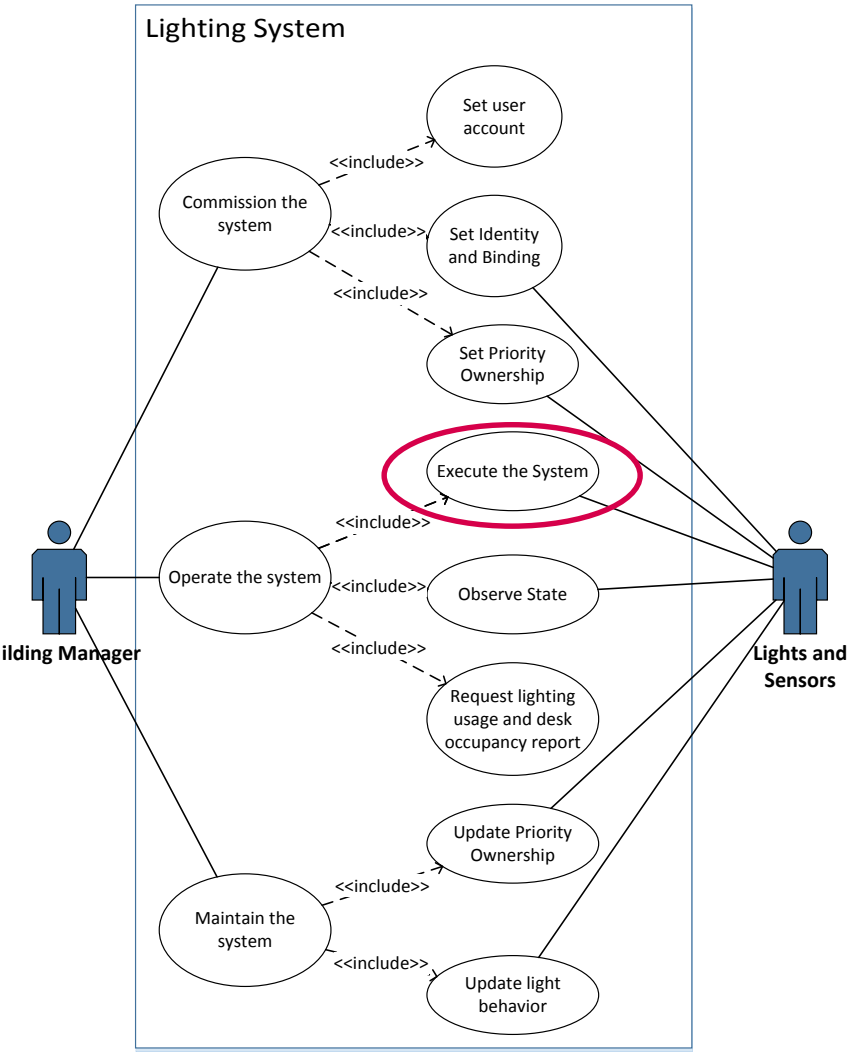
Object Instances in Light Device

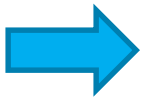
Object	Object ID	Object Instance ID	Notes	Specification
Sensor Profile	10350	0	Represent the Sensor Device's static and dynamic information	Private Object

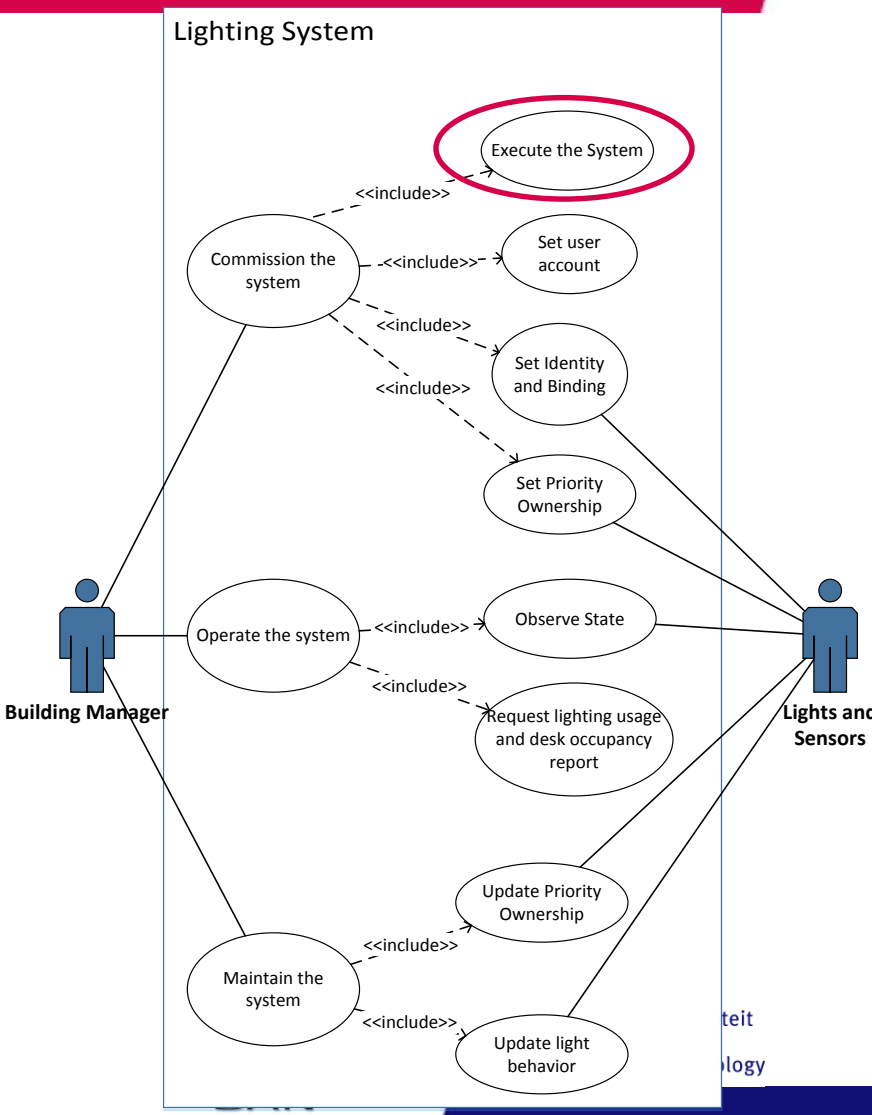
Resources in the *Sensor Profile* Object (Object Id: 10350)

Resource Name	Resource ID	Opera-tion	Type	Value	Notes
Sensor ID	0	R, W	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, W	String	"Sensor Device"	The device type, in this case a "Sensor Device"
Sensor State	2	R, W	String	"USED" or "FREE"	Represents the state of the Sensor Device, whether it is in the "USED" 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 assign to a group of light(s) and a sensor.
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

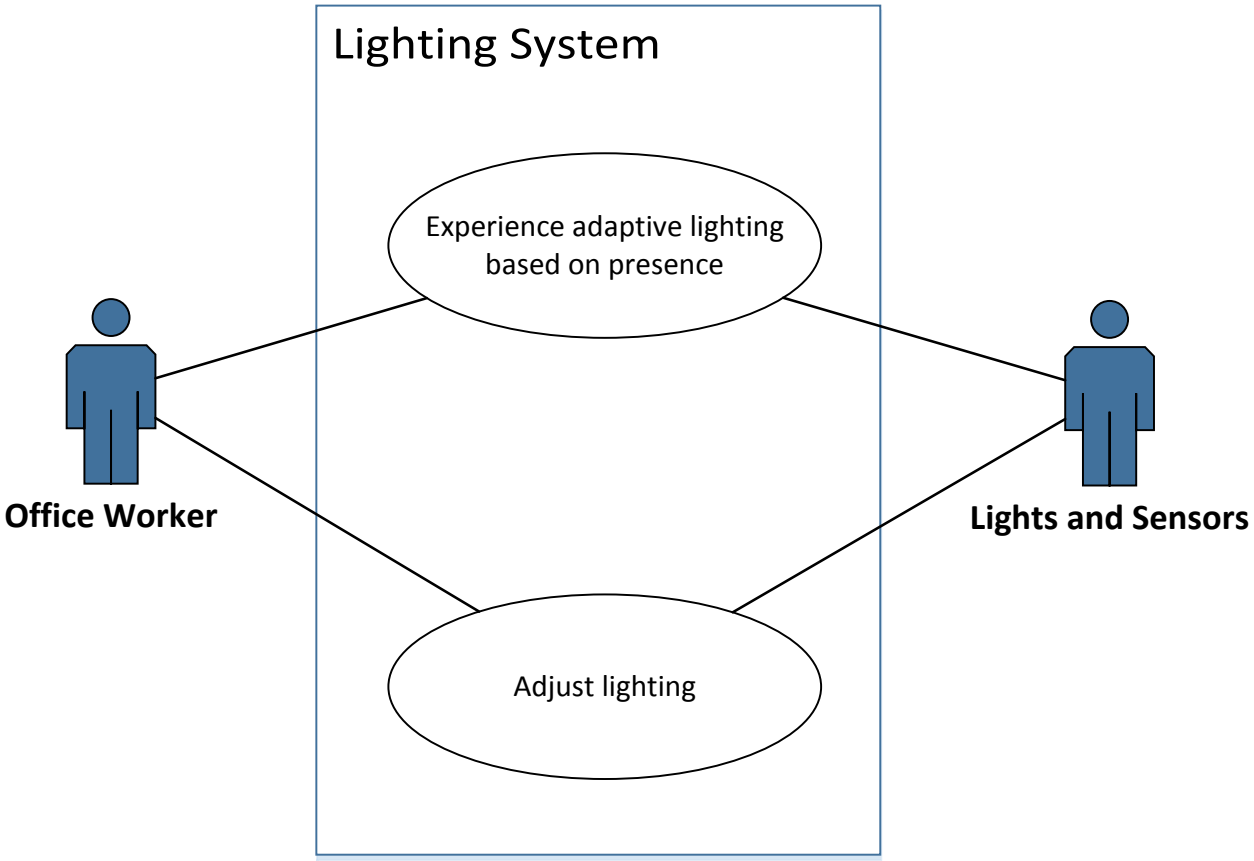
Use Case Diagram (Building Manager)



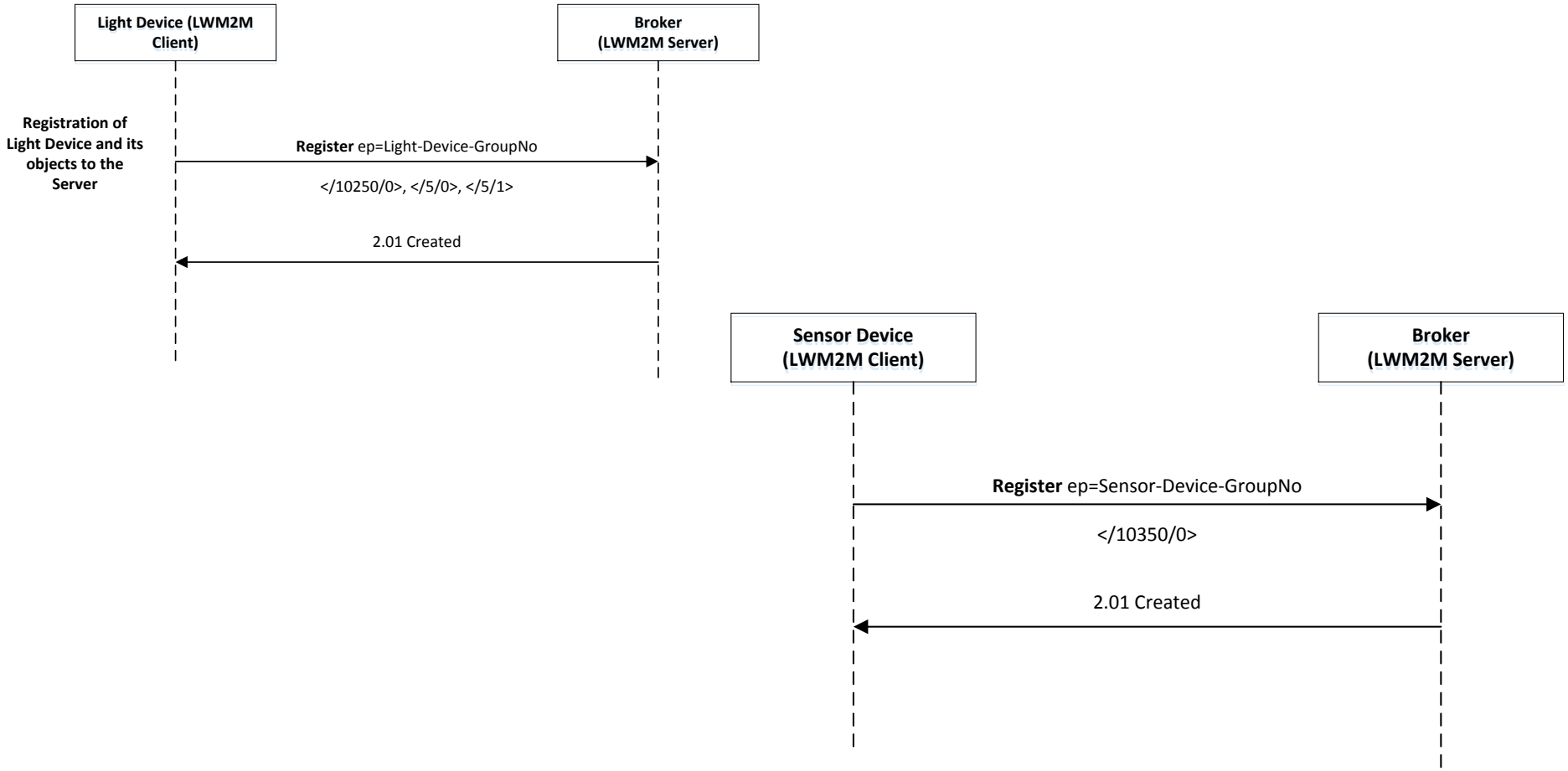




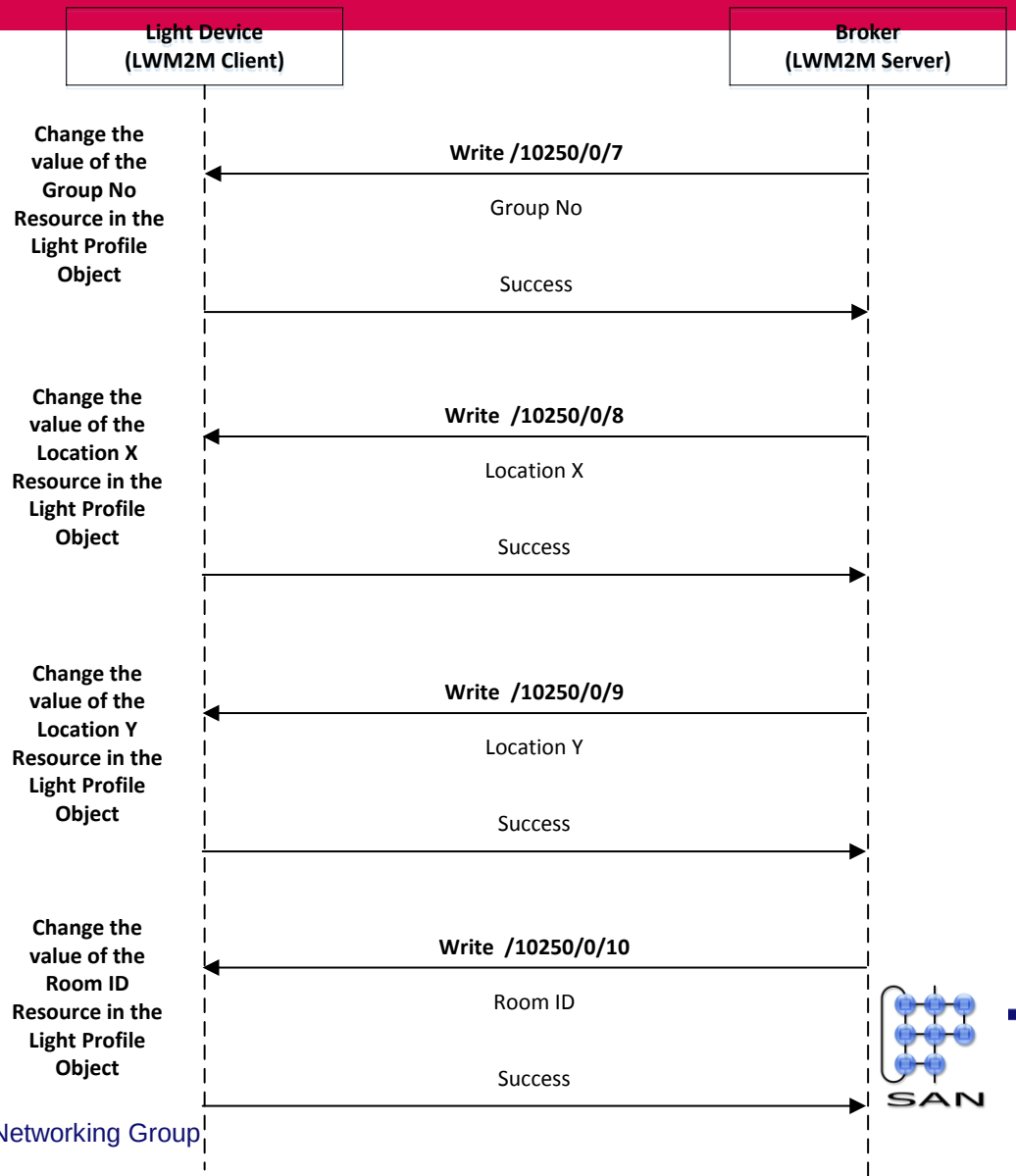
Use Case Diagram (Office Worker)



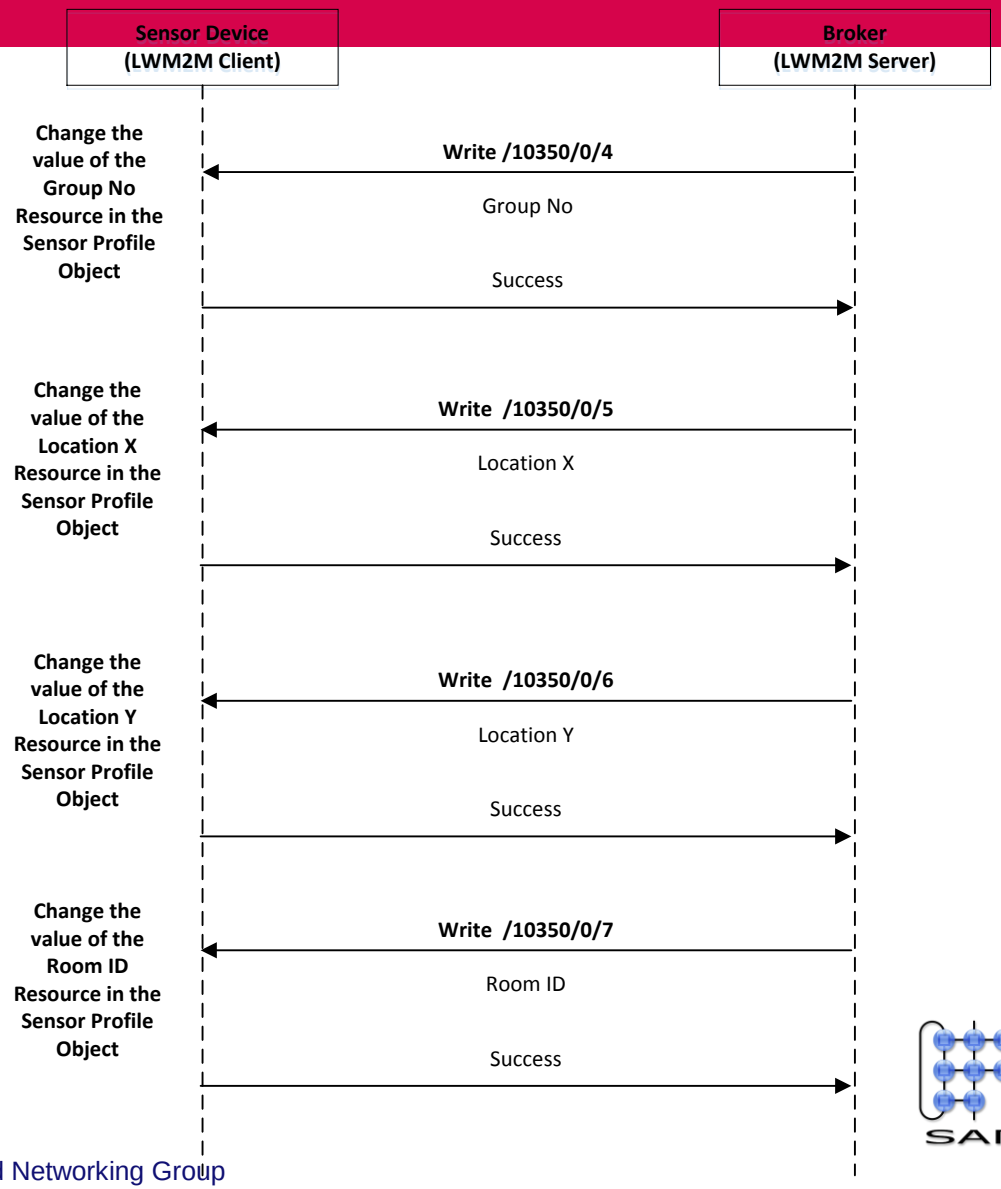
“Execute the System” use case



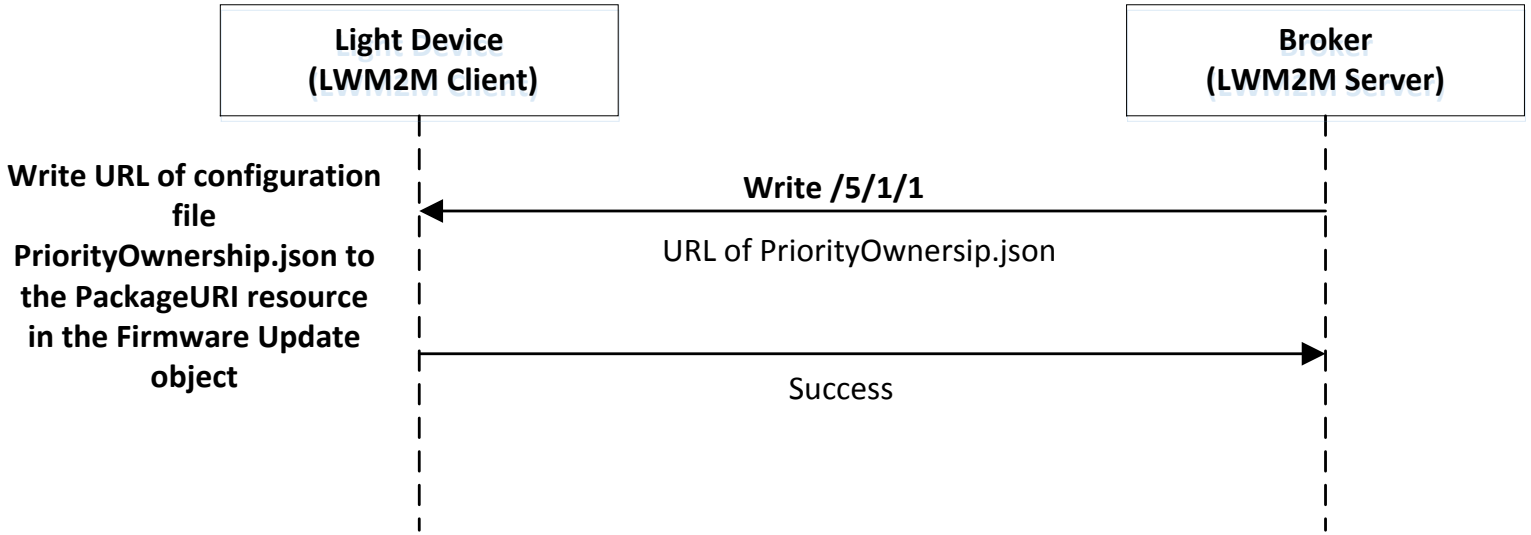
“Set Identity and Binding” use case



“Set Identity and Binding” use case

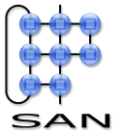


“Set Priority Ownership” use case



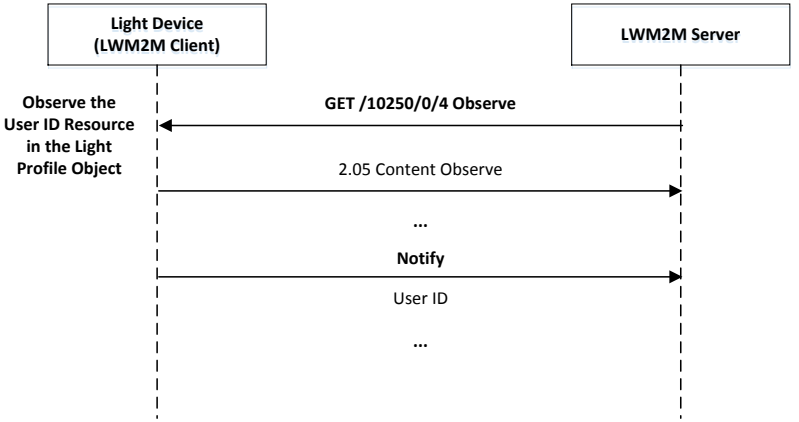
OwnershipPriority.json

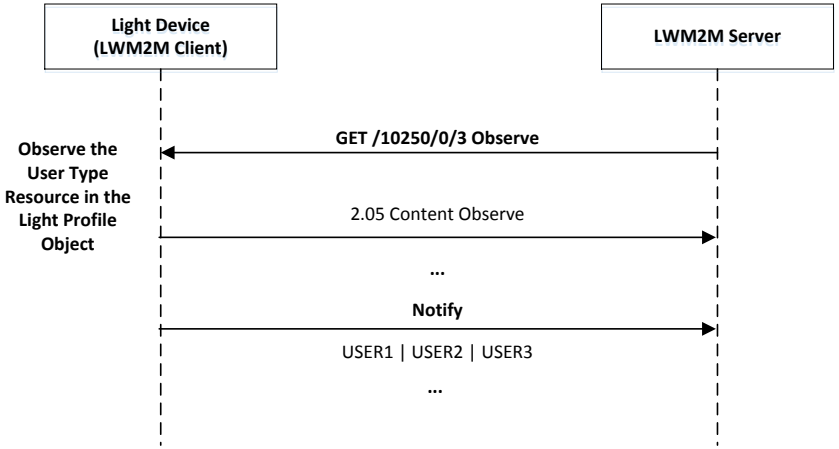

```
[{
  "user_type": "USER1",
  "user_id": "Office-Worker-1",
  "light_color": "(255, 255, 255)",
  "low_light": false,
  "user_location_x": 2,
  "user_location_y": 1
 },
 {
  "user_type": "USER2",
  "user_id": "Office-Worker-3",
  "light_color": "(255, 0, 0)",
  "low_light": true,
  "user_location_x": 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
}]
```

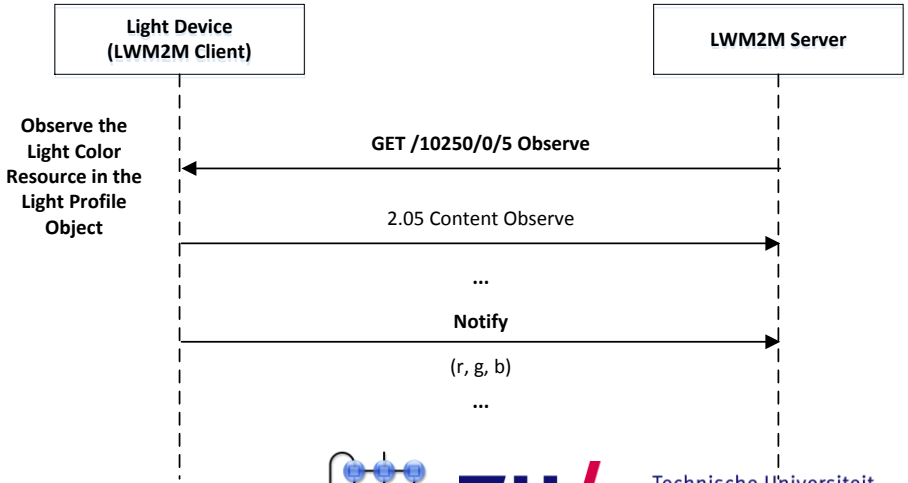



TU/e Technische Universiteit Eindhoven University of Technology

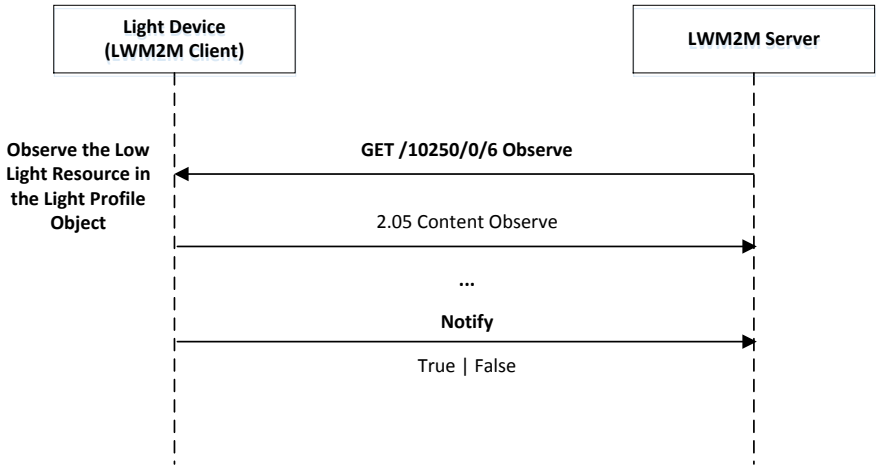
“Observe State” use case (cont’d)

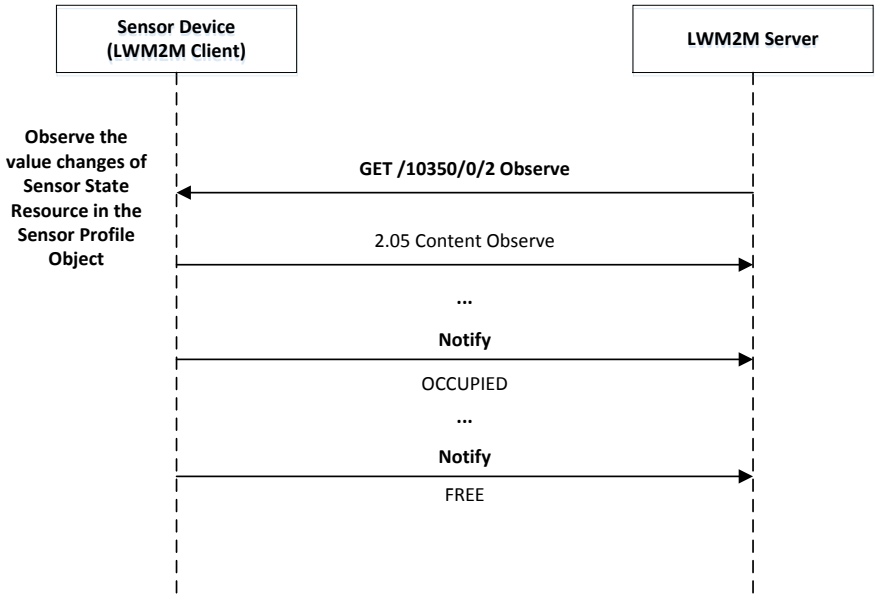




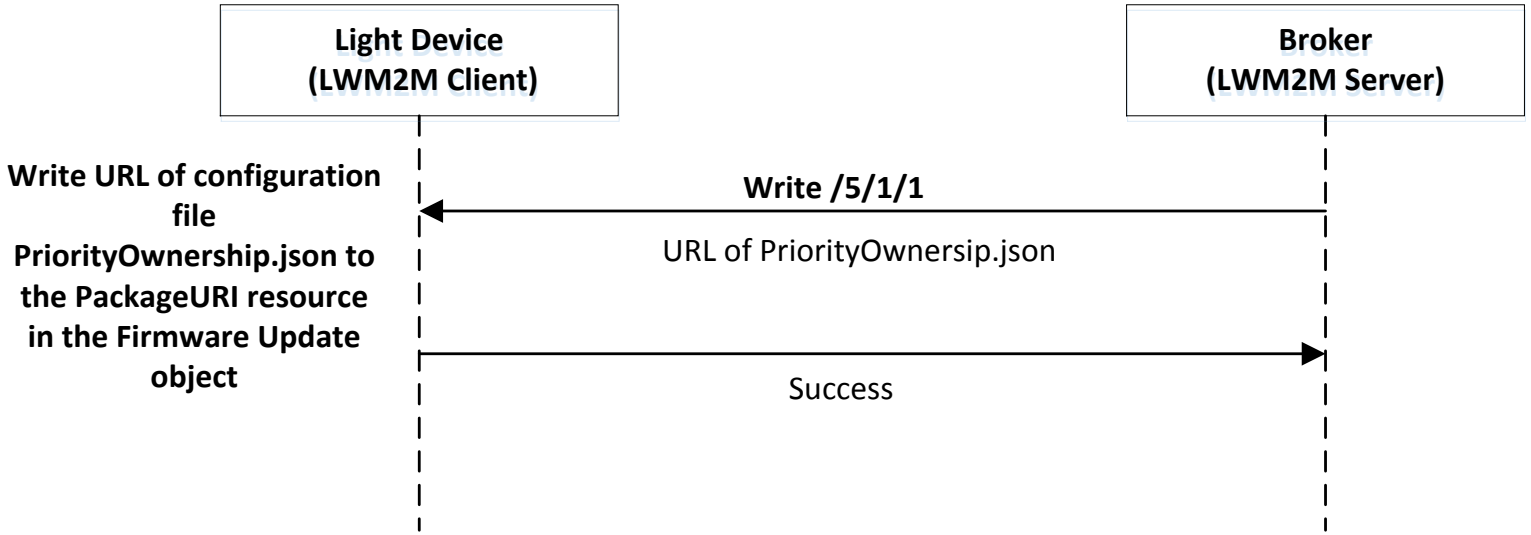


“Observe State” use case

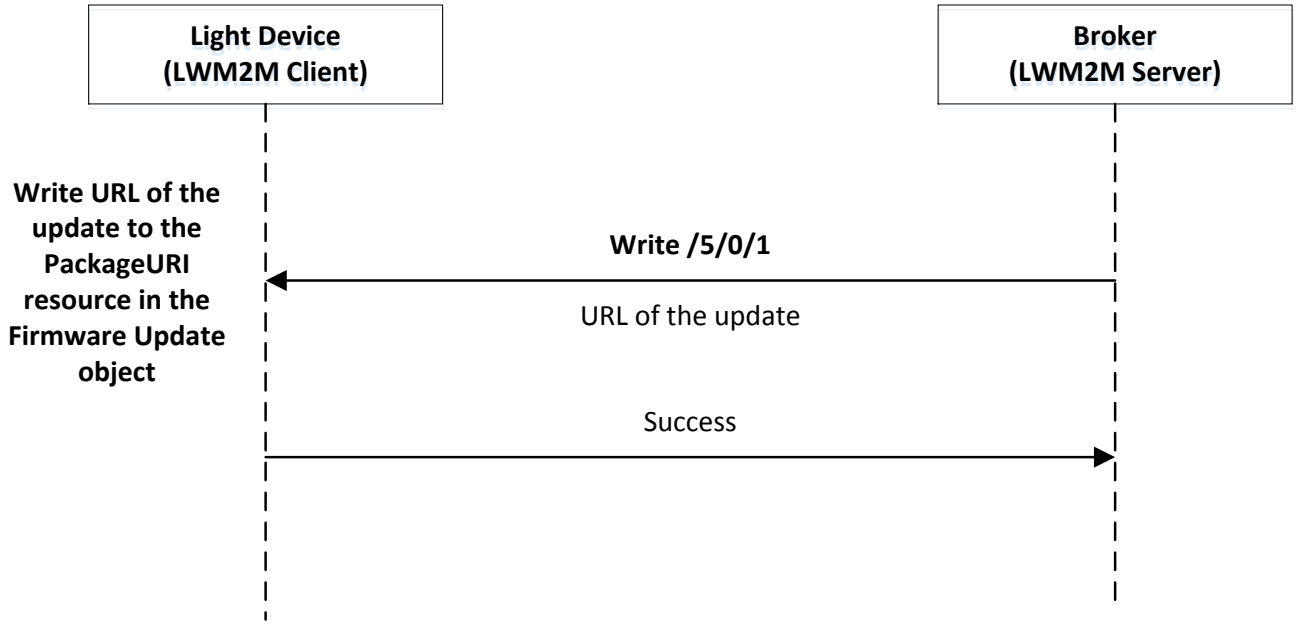




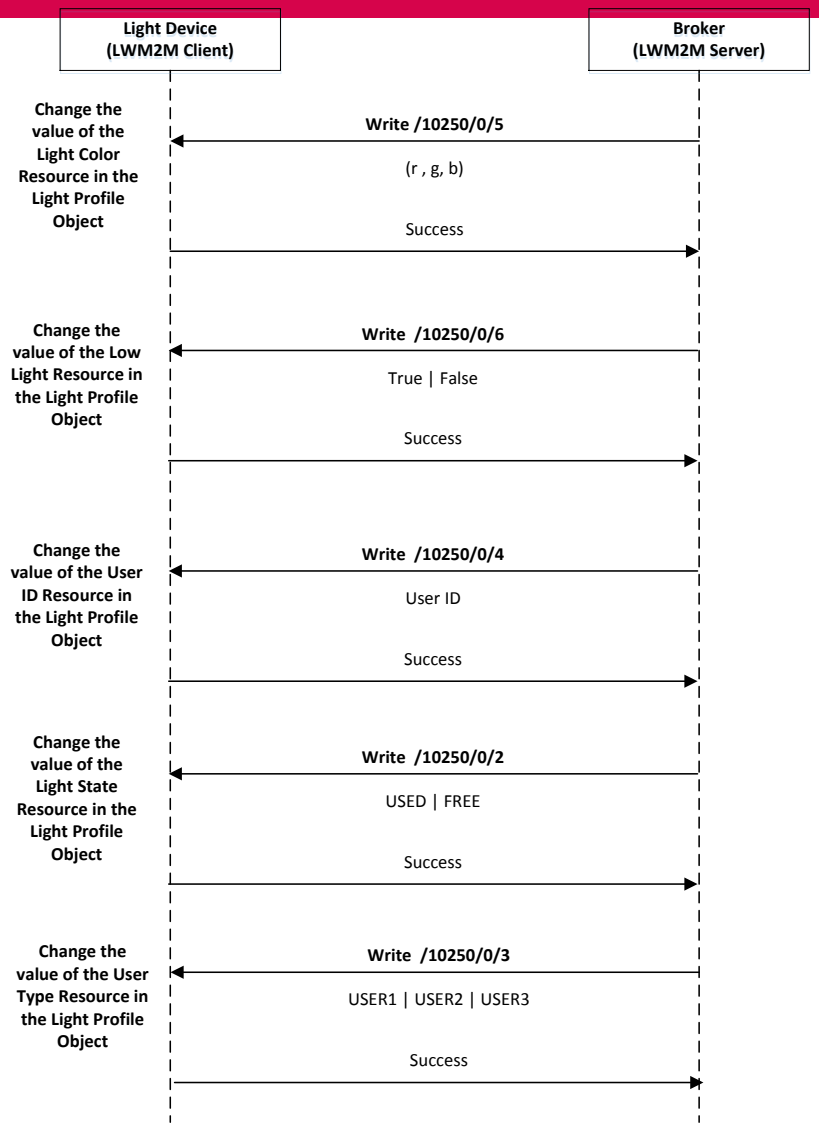
“Update Priority Ownership” use case



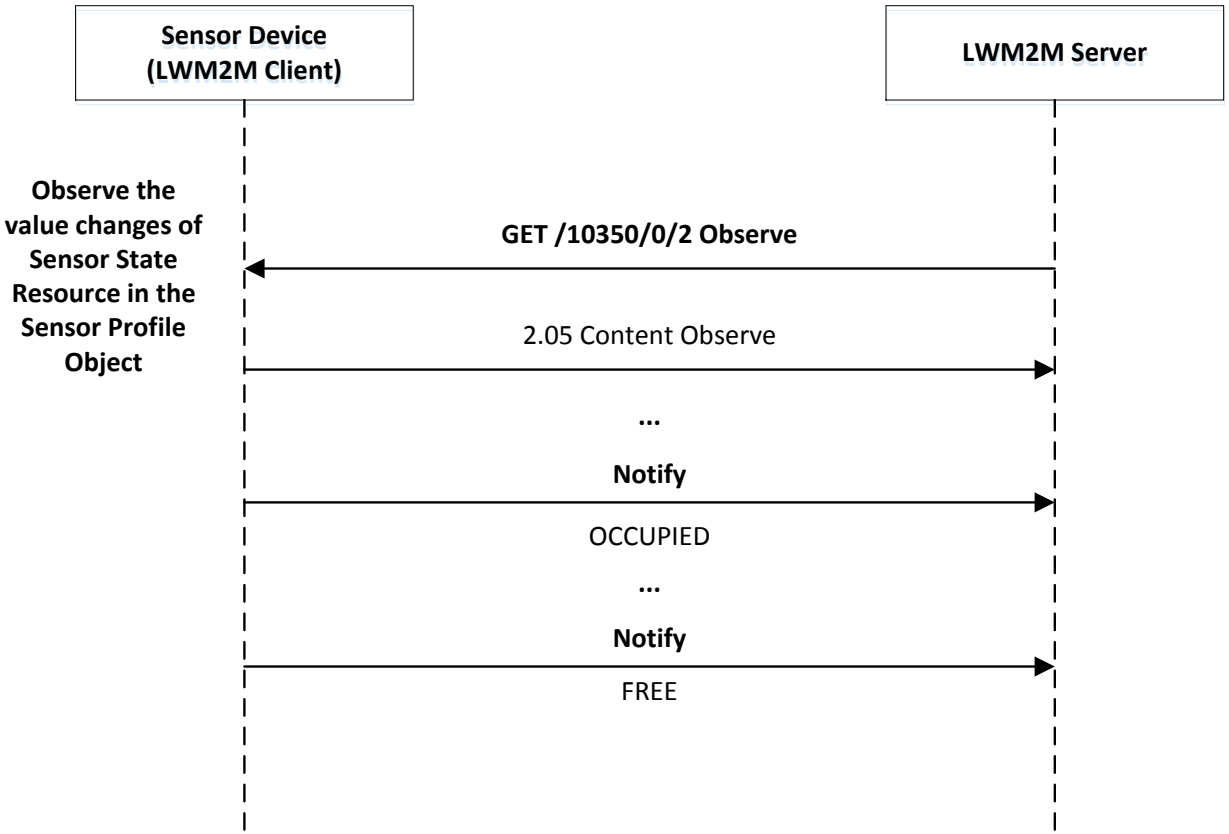
“Update Light Behavior” use case



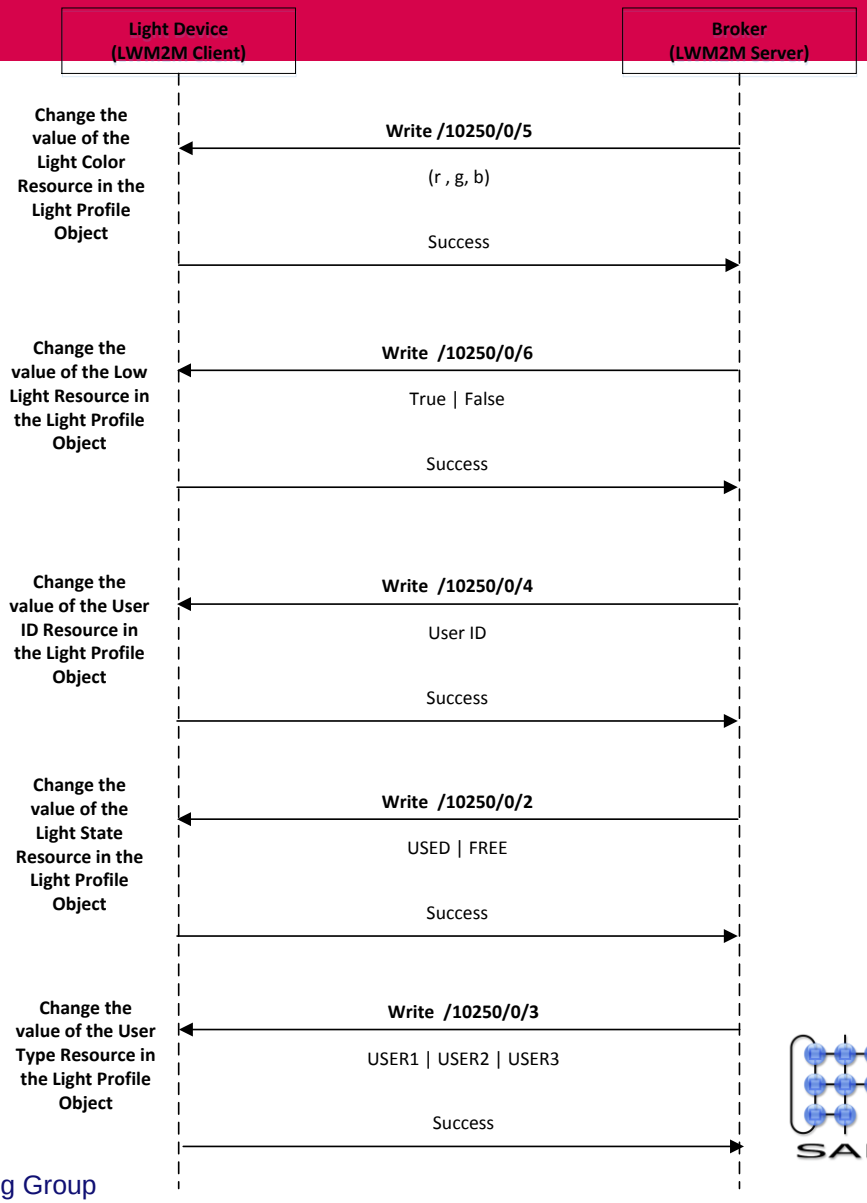
“Adaptive Lighting Based on Presence” for centralized behavior deployment



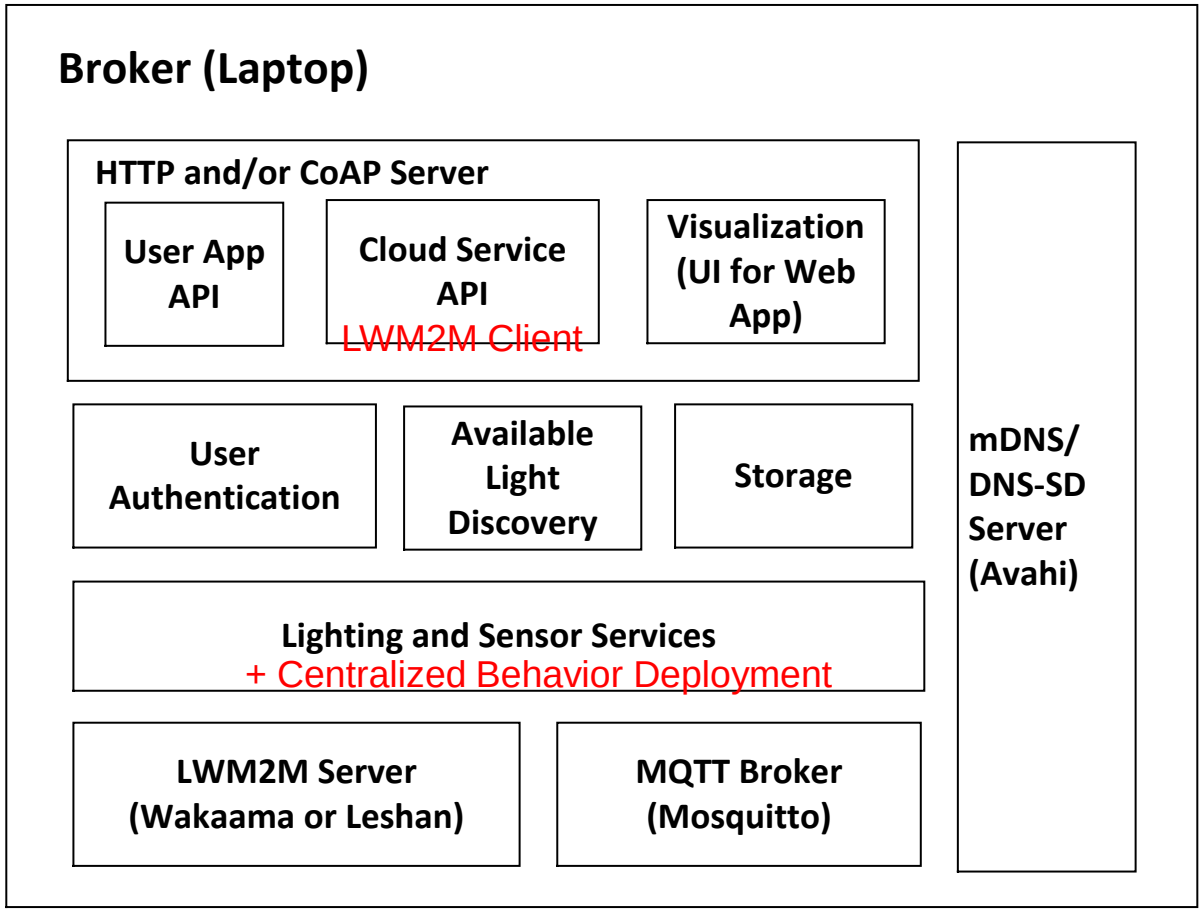
“Adaptive Lighting Based on Presence” for centralized behavior deployment



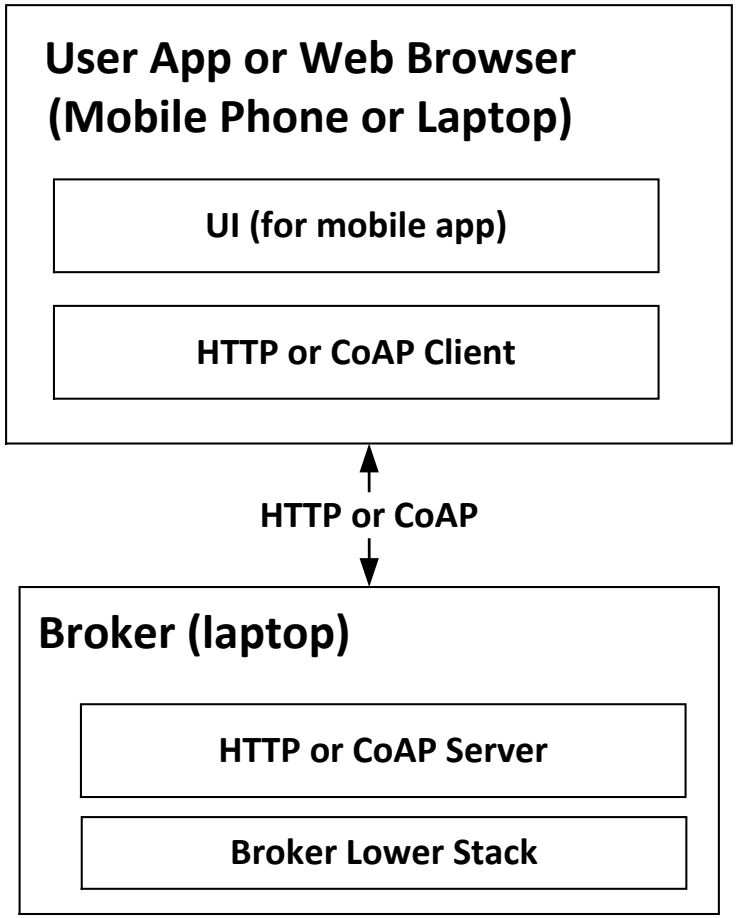
“Adjust Lighting” use case



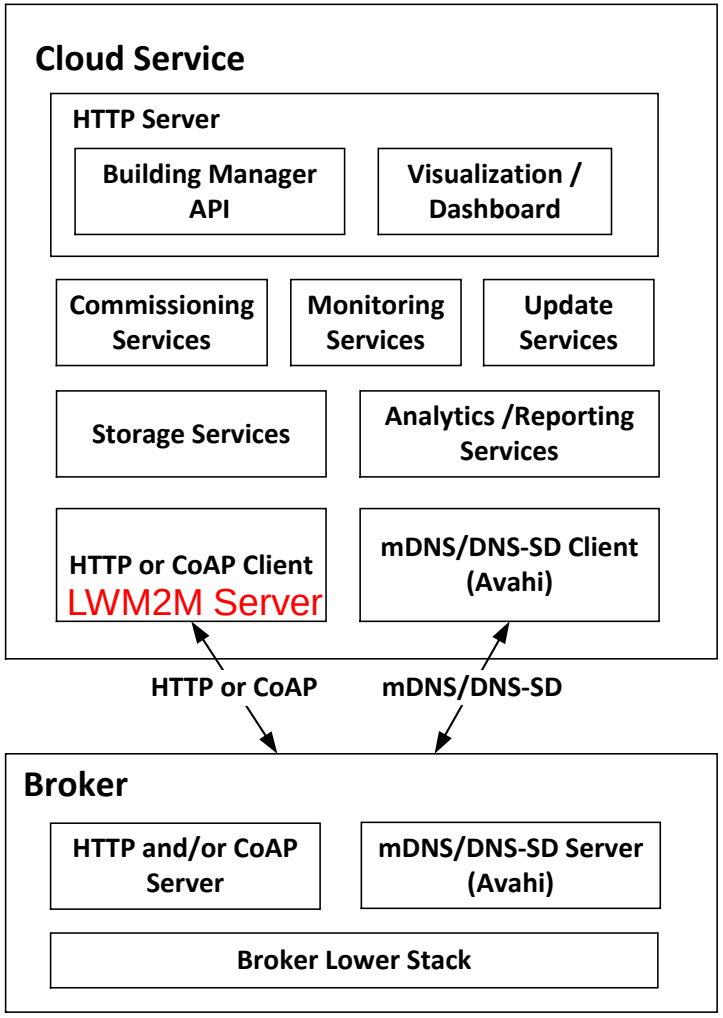
Broker Architecture



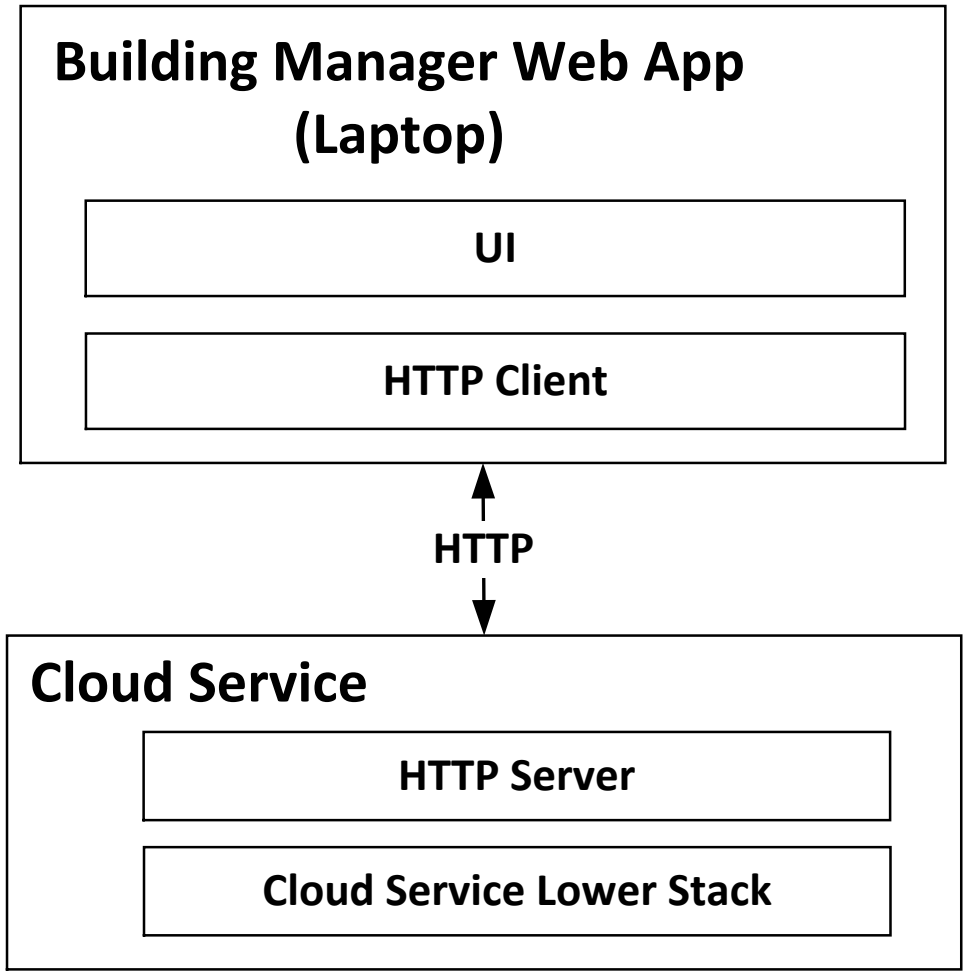
User App Architecture



Cloud Service Architecture



Building Manager App Architecture



Tutorials

http://www.win.tue.nl/~lrahman/iot_2016/tutorial/

- Tutorial for Raspberry Pi with Sense Hat
- Tutorial for Raspberry Pi with Camera module
- Tutorial for txThings, an open source implementation of CoAP in Python.
- Tutorial on Avahi, an implementation of mDNS-SD, a service discovery protocol
- Tutorial for Eclipse Leshan, an open source programming framework for developing LWM2M server and client in Java.
- Tutorial for Wakaama, an open source programming framework for developing LWM2M server and client in C.
- ~~Tutorial for mbed, a semi open source programming framework for developing LWM2M server and client in C++~~
- Tutorial for Paho and Mosquitto, open source implementation of MQTT client and server. Paho is available in several programming language.
- Protocol description between the end devices and the broker, which will be based on the LWM2M and MQTT specification.