# 2IMN15: Internet of Things Practical Assignment

## Introduction

The goal of the practical is to obtain hands-on experience with protocols and techniques for Internet of Things (IoT) systems. We target the following aspects:

- 1. Connected IP infra structure
- 2. Service discovery: advertisement, description, discovery
- 3. Designing a REST protocol, working with multiple clients, SOA, publish/subscribe, "Cloud" storage
- 4. Managing a device (different from using it, applying management protocols)
- 5. Interoperability
- 6. Framework(s) for programming

## Problem Outline and Scope

A set of desks in an open office room is equipped with a sensor to observe occupation and lights to illuminate its area. The sensors and lights are also capable to do local processing and they belong to a certain group which corresponds to the desk. Each desk is dedicated to a person. The sensors and lights are IP connected and part of a larger system for lighting control and building management. An office worker in the room can adjust the color of the lights through an app when she or he has ownership of the lights. This ownership is regulated by an authentication and ownership priority mechanism defined by the building manager during commissioning time of the lighting system. Building manager can observe the state of the lights and the sensors, and can view the history of the desks occupancy as well as lighting and energy usage. Building manager can also change system configuration and perform lighting behavior updates during maintenance.



Picture from <a href="http://www.openais.eu/en/home/">http://www.openais.eu/en/home/</a>

## Deployment Setup and System Entities

We will simulate this system, but remain close to a possible reality. Entities that need to be present in the system (e.g. lights, sensors, users, building manager and managerial entities) should be implemented as separate processes. We provide Raspberry Pi boards with the SenseHat extension shields and camera shields to represent lights and sensor. User and other entities are to be represented inside phones, tablets or regular computers.

Lights. They have

- Identity (Light ID, Device Type, Room ID, Location (X,Y))
- Binding information (It's Group ID, and Ownership Priority)
- Color settings for User 1, User 2 and User 3
- Two states: FREE and USED (as illustrated in the following picture)



Each light has three kinds of ownerships: User 1, User 2 and User 3. User 1 is the office worker who sits underneath the light and has the highest priority of ownership. When User 1 sits, ownership goes directly to User 1. User 2 is the office worker who sits closest to User 1 and has the second priority of ownership. When User 1 is not present, ownership is automatically transferred to User 2. User 3 can be all of the other office workers in the room. User 3 can take ownership of the light when User 1 and User 2 are not present. If more than one person wants to be user 3, the person with the highest priority (the closer s/he sits to User 1, the higher the priority) wins and becomes User 3. When an office worker has ownership of the light, s/he can adjust the color and intensity of the light. When the light is in the *FREE* state, it goes off if no one sits in the room.

Lights expose a service that admits

- to change their states
- to inspect their states
- directly or through eventing (after subscription)

The access to the services is done through LWM2M and MQTT. The lights implement a LWM2M client and registers themselves to a LWM2M server which resides in a broker (area controller) in the same area. The light first needs to discover the broker through mDNS and DNS-SD protocol. Once it finds the broker's IP address, the light can register to the LWM2M server in the broker.

Lights subscribe to sensor readings through an MQTT broker (lights need to know whether User 1, User 2, User 3 or other persons are sitting in the room). Therefore, all lights will implement MQTT clients as well.

Sensors. They have

- Identity (Sensor ID, Device Type, Room ID, Location (X,Y))
- Binding information (Group ID which corresponds to a desk it belongs to)
- Face recognition information (User 1's face pattern).
- Two states: FREE and OCCUPIED (as illustrated in the following picture)



- When no face is detected, the sensor is in the *FREE* state.
- When User 1's face is detected, the sensor turns to the OCCUPIED state. The sensor triggers "User 1 sits" event and the light goes to USED state where User is set to User 1 and Light Setting is set to User 1's setting. When the face of User 1 is no longer detected, the sensor triggers "User 1 leaves" event.
- When someone other than User 1 is sitting on the desk, the sensor will stay in the FREE state. The lights will stay in its current state.

Sensors expose a service that admits:

To inspect their occupancy state

The access to the services is done through LWM2M and MQTT. The sensors implement a LWM2M client and registers themselves to a LWM2M server which resides in a broker (area controller) in the same area. The sensor first needs to discover the broker through mDNS and DNS-SD protocol. Once it finds the broker's IP address, the sensor can register to the LWM2M server in the broker. Sensors publish its sensor readings through an MQTT broker, therefore, all sensors will implement MQTT clients as well.

#### Managing entities

- Data Store
  - Create a data store that records what happens over time. Desk occupancy and lighting states over time. These data can be used to generate reports on lighting (or energy) usage and desk occupancy (or user's behavior patterns).
- Visualizer: visualizes the current state of the lighting system: overview of the *light* and sensor ID, their location and their *states*. Visualizes the reports on lighting (or energy) usage and desk occupancy (or user's behavior patterns).
- Broker / area controller:
  - used as contact point by the users
  - each office room has its own broker/area controller
  - maintain the current states of the lighting system
  - implement ownership priority policy for each group of lights (who is User 1, who is User 2 and a set of persons eligible for User 3 and their priority)
  - Implement authentication of the user app.
  - Discover lights that can be controlled by an authenticated user.
  - Implement configuration service for the building manager.
  - Implement LWM2M server for managing the lights and sensors.
  - Implement MQTT broker that has sensors as the publishers and lights as the subscribers
- Cloud service:
  - Maintain an API server for building manager app or other third party apps
  - The cloud service resides in the Internet or in the local network, and connects to the brokers of various office rooms.
  - The cloud service first needs to discover the broker through mDNS and DNS-SD protocol. Once it finds the broker's IP address, the cloud service can connect to the broker.
  - The cloud services uses APIs of the LWM2M server in the broker to get access to the lights and sensors.

**Users.** Users represent the office workers that work in the open office room. They have an identity and their behavior is sketched as the following:

- move through 2D space in the open office room
- decide that they want to sit on their desk. While sitting, they want to adjust the color of the lights above them and in their surroundings through an app. The app connects to the broker and can do the following:
  - $\circ~$  discovers the lights of which he can take ownership. The discovery is done by the broker through its LWM2M server's client registry.
  - adjust the color and intensity of the lights directly above his desk

- $\circ\;$  adjust the color and intensity of the lights in his surrounding of which he can take ownership
- leave their desk for breaks or for going home
  - this releases ownership of the lights

**Building Manager**. Building manager is responsible for configuring and maintaining the lighting system. They have an identity and their behavior is sketched as follow:

- During commissioning time, building manager uses his app to configure the identity and binding information of the lights and sensors in various office rooms. The app is connected to the cloud service.
- The building manager creates and sets user account (name, email, username, password, face pattern)
- The building manager binds user accounts to desks (desks are bound to rooms)
- The building manager sets the lights and sensors' location.
- The building manager binds lights and sensors to groups, and groups to desks.
- The building manager sets the user priority policy for each light.
- During operation, building manager uses his app to monitor the states of the lights and the desk or room occupancy. He can also request reports on lighting usage and desk occupancy history.
- During maintenance, building manager uses his app to change necessary configuration (lights and sensors identity and binding information or authentication and ownership priority policy configuration). This is done through the APIs to the LWM2M server in the broker.
- The building manager can also update the lights behavior through his app. This is done through the APIs to the LWM2M server in the broker. Example of a new behavior would be, each light is showing different random color every 1 second.

## Use Cases



#### Use Case LS3: "Commission the System"

User-goal level

Principle Actor: Building Manager (BM)

Pre-condition: The lighting system has been installed but not been commissioned.

Description:

- BM: opens a blue print of the lighting installment in a room where lights and sensors are mapped to Location (X,Y). Location is relative to the room, where one corner of the room is considered as (0,0). X and Y are approximation values represented in meter. This blue print also shows the mapping of lights and sensors to Groups and the mapping of Groups to Desks. In the open office room, one Group consists of two lights and one sensor. One Group is associated to one Desk.
- 2. BM opens a list that maps office workers to desks. An office worker sits in exactly one desk.
- 3. BM: opens the Building Manager Web Application on his computer and opens "Commission the System" menu
- 4. LS: the app displays the sub menus for commissioning the system.
- 5. BM: chooses the "User Account" menu.
- 6. BM: sets user account for office workers and sets their identity and data (name, email, username, password, face pattern)
- 7. LS: saves the user account information in the relevant entities
- 8. BM: goes back to the "Commission the System" menu and opens the "Identity and Binding" menu
- 9. BM: sets the identity of the lights and sensors (their device type and location in the room)
- 10. BM: binds lights and sensors to groups, binds groups to desks, binds desks to rooms and binds an office worker to a desk.
- 11. LS: saves the binding information in the relevant entities
- 12. BM: goes back to the "Commission the System" and opens the "Ownership Priority" menu
- 13. BM: sets the ownership priority information for each light. BM sets User 1, User 2 and User 3 of each light. The light setting of User 1, User 2, and User 3 are set to default value of white color with high intensity, rgb (255,255,255), lowlight = false
- 14. LS: saves the ownership priority information in the relevant entities
- 15. BM may repeat steps 3 -14 until the lighting system is ready for operation.

#### Use Case LS3: "Operate the System"

User-goal level

Principle Actor: Building Manager (BM)

Pre-condition: The lighting system has been commissioned.

Description:

- 1. BM: opens the Building Manager Web Application on his computer and opens "Operate the System" menu
- 2. BM: Execute the broker (area controller) in all the rooms
- 3. LS: The broker is running
- 4. BM: Execute all the lights and sensors
- 5. LS: The lights and sensors are running, discover the broker and register themselves to the broker
- 6. BM: Execute the cloud service
- 7. LS: The cloud service is running, discovers the broker and connects to the broker
- 8. LS: The lighting system is ready for operation
- 9. BM: During operation, opens the menu for monitoring the states of the lights and sensors
- 10. LS: Visualizes the states of the lights and sensors
- 11. BM: Request lighting usage and desk occupancy report over certain time
- 12. LS: Visualizes the requested report
- 13. BM: may repeat step 9 or step 11

#### Use Case LS3: "Maintain the System"

User-goal level

Principle Actor: Building Manager (BM)

Pre-condition: The lighting system has been commissioned and are operating.

Description:

- 1. BM: opens the Building Manager Web Application on his computer and opens "Maintain the System" menu
- 2. BM: opens the "Update Ownership Priority" menu
- 3. BM changes User 2 of each light in a room to different office worker.
- 4. LS: saves the changes and the light system follows the new ownership priority setting.
- BM: goes back to "Maintain the System" menu and opens the "Update light behavior" menu

- 6. BM: uses a provided menu to update new executable of light behavior where each light is showing different random color every one second. BM uploads the new executable in a network location, and input the URL of the new executable on a provided URL field in the app.
- 7. LS: downloads the new executable from the relevant entities and replace the current executable with the new executable
- 8. LS: Each light in the system now changes to random color every one second. The user app cannot be used to adjust the light anymore.



### Use Case LS1: "Experience adaptive lighting based on presence"

User-goal level

Principle Actor: Office Worker (OW)

Pre-condition: The lighting system has been commissioned and executed by the building manager

Description:

- 1. OW: enters the room and sits on her desk
- 2. LS: the sensor detect OW's face and set the lights above her desk to her personal setting.
- 3. LS: turn the lights of the desk closest to her to her personal setting if it is not occupied.
- 4. LS: turn the lights of the rest unoccupied desks to dim (rgb(250, 200, 100); lowlight)
- 5. OW: leaves her desk

- 6. LS: the sensor doesn't detect OW's face, and after 3 seconds turn the lights above her desk to one of these settings:
  - a. To the personal setting of the desk closest to her if it is occupied.
  - b. To dim (rgb(250, 200, 100); lowlight) if the desk closest to her is unoccupied and the room is not empty (someone is sitting).
  - c. To off if no one else is present in the room

#### Use Case LS2: "Adjust lighting"

User-goal level

Principle Actor: Office Worker (OW)

Pre-condition: The lighting system has been commissioned and executed by the building

manager

Description:

- 1. OW: enters the room and sits on her desk
- 2. OW: open the User App on her mobile phone
- 3. LS: recognizes that OW is sitting and activates the menu for adjusting lights
- 4. OW: opens the menu for adjusting lights
- 5. LS: discovers the lights that she can adjust and displays this information on the app
- 6. OW: chooses the light that she wants to adjust on her app
- 7. OW: adjust the color of the chosen light to the color of her wish on her app
- 8. LS: set the light to the color of her wish
- 9. OW: repeat step 6-7 until she is satisfied

## Organization of the Practical

Each exercise team (consists of two students) will receive one Raspberry Pi board with one Sense Hat shield which will represent the light and sensor. Three exercise teams shall partner up to work on the practical assignment. These three teams (which form a practical team) will work together to design and implement a lighting system with three Raspberry Pi boards, three Sense Hat shields, and one camera shield. However, eventually the system should work with all lights and sensors developed by all practical teams. All entities and use cases mentioned in the practical description above should be realized by the practical team (consists of six students), with clear task divisions between the exercise teams (of two students). The whole system can, for example, be divided into three parts: (1) the light and sensor; (2) the broker and the user app; (3) the cloud service and the building manager app, where one exercise team (of two students) can work on one part of the system. We will provide interface description between the Things (the light and sensor) and the broker, which will be specified as LWM2M and MQTT based protocols. However, all member of the class can discuss this interface further in the Canvas forum, in case some interfaces are missing or needs improvement. All practical teams should then adhere to the agreed interface between the Things and the broker. The interface between other entities in the system (i.e. between the broker and the cloud service, between the user app and the broker, between the building manager and the cloud service) can be defined independently by members of the practical teams.

At the end of the practical assignment, the practical teams shall operate their system in a plug fest to test interoperability between different parts of the systems of different practical teams. The broker and cloud service of each practical team should be interoperable with the Things developed by different practical teams. During the plug fest, we will also test functionality of the realized systems by all practical teams.

At the end of the course, each exercise team (of two students) should deliver a detailed report on their contribution in the practical which consist of detailed design and implementation documentation on the parts that they were working on and the rationale behind the design and implementation decisions. They should describe any architectural or design pattern that they use. Description of system interfaces and test result with their partner teams should also be included in the report. Other than that, the report should include discussion about relevant nonfunctional requirements and a reflection on the practical assignment. Grading for the practical assignment will be based on the contribution that each exercise team (of two students) makes in the practical execution.

## Architecture Guidelines

The practical team shall design and implement two types of light behavior (control) deployment: (1) decentralized control deployment (control is deployed on each light), and (2) centralized control deployment (control is deployed on the broker). The system should provide a run time setting that can be used to switch execution from decentralized control deployment to centralized control deployment, and vice versa. For each type of control deployment, perform a measurement of the delay from User 1's face detection until light response (Time to Light). This measurement result should be included in the practical assignment report of the relevant teams.