AutoLight App

-

IoT Project

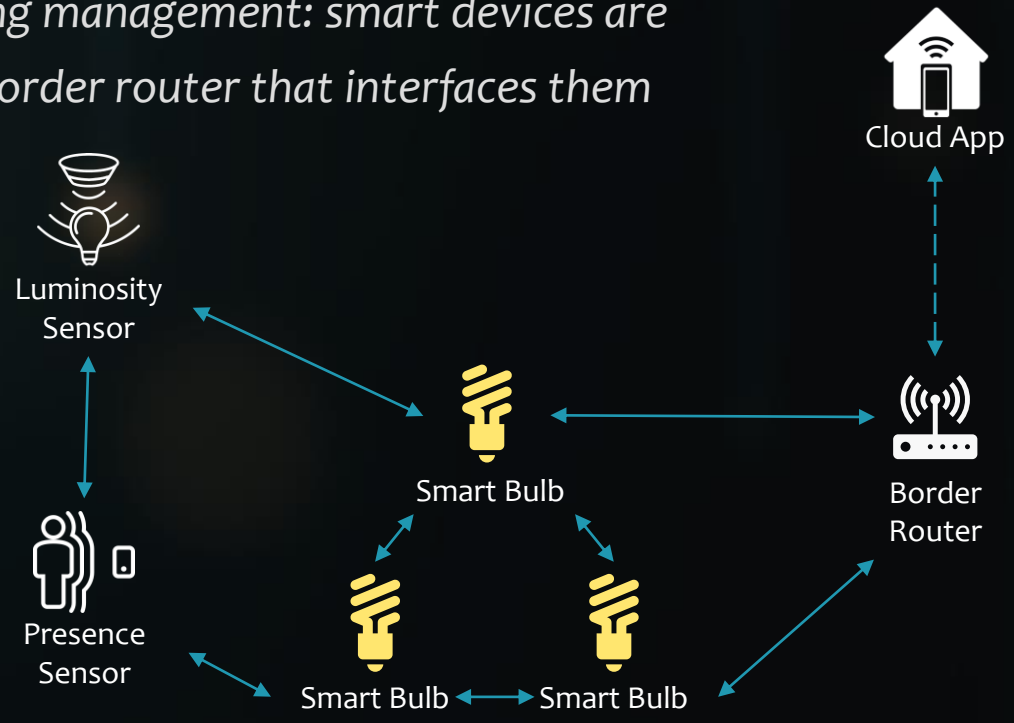
Stefano Petrocchi

Architecture

*Home automation application for smart lighting management: smart devices are connected by **multi-hop RPL protocol** with a border router that interfaces them with a cloud application via **CoAP***

Devices:

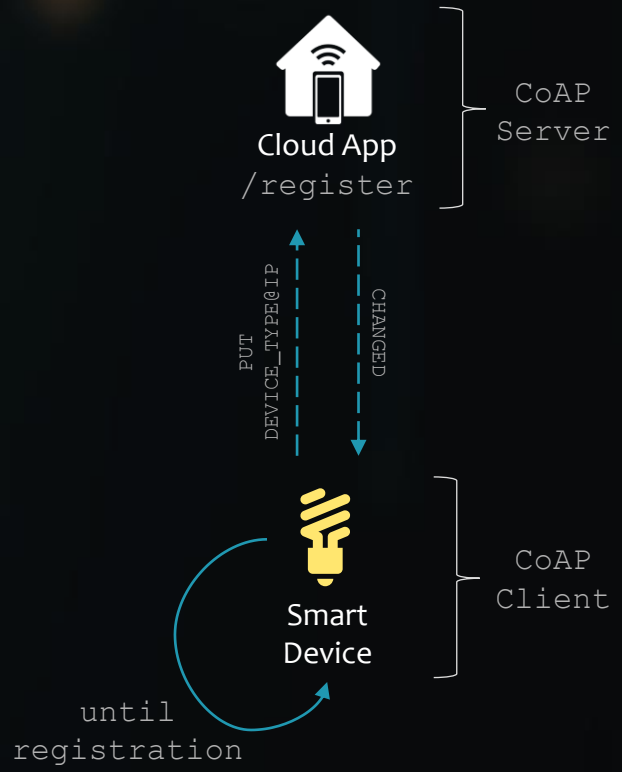
- ❑ *Cloud App*
- ❑ *Border Router*
- ❑ *Smart Bulbs*
- ❑ *Presence Sensors*
- ❑ *Luminosity Sensors*



Devices Registration

To interact with the application, the devices must first register:

1. The java application includes a *CoAP server* with a known IP address and `/register` resource
2. At startup, each smart device, after connecting to the network, sends a *CoAP* `PUT`* message as *client* to `/register` resource, inserting `DEVICE_TYPE@IP` in the payload
3. The application receives the message, handles it by creating an appropriate data structure to represent the device according to the *type*, stores the *IP* and sends a `CHANGED` response code if the operation is successful
4. The smart device attempts registration until success



* in this implementation the registration is idempotent, that's why was chosen a PUT instead of a POST

Functionalities

After smart devices ***registration***, the following modes are available:

- Manual Mode:
  - Change bulb luminosity
  - Set all bulbs luminosity
  - Switch bulb on/off
  - Switch all bulbs on/off
- Auto Mode:
  - Eco mode: *The lights turn on only if someone is present inside the room*
  - Constant room luminosity: *The luminosity inside the room is kept constant at the preferred value regardless of external luminosity*

Cloud App

Java application that manages the smart home's operations and provides the user with a command line interface

Available commands :

- `!exit|!quit` : to exit
- `!reg` : lists all registered devices
- `!info` : refresh and shows bulbs info
- `!mode [manual|auto]` : to pass to manual/automatic mode
- `!des_lum [value]` : to set desired luminosity value
- `!sw [Bulb ID] [ON|OFF]` : to switch bulb on or off
- `!sw ALL [ON|OFF]`: to switch all bulbs on or off
- `!lum [Bulb ID] [+|-] [value]` : increase/decrease bulb luminosity value
- `!lum [Bulb ID] [value]` : set bulb luminosity value
- `!lum [ALL] [value]` : set all bulbs luminosity value

CoAP Resource:

- `/register`:
  - **GET:** get registered devices count
  - **PUT:** performs the device registration operation

Smart Devices Interactions: all interactions with devices are managed *asynchronously* via CoAP

Smart Bulb

Smart dimmable light with intensity ranging from 0% to 100%

LEDs:

- YELLOW: The device is not yet connected to the network
- GREEN: The light is turned ON and the device connected to the network
- RED: The light is turned OFF and device connected to the network

Button: toggle light ON <-> OFF

CoAP Resources:

- `/luminosity`:
  - **GET:** get current luminosity value
  - **POST:** increase (`+`) or decrease (`-`) luminosity value
  - **PUT** (`lum`): set luminosity value
- `/switch`:
  - **GET:** get the current state (ON/OFF)
  - **POST:** toggle light ON <-> OFF
  - **PUT:** set light ON or OFF

Presence Sensor

Presence sensor that simulates the presence or absence of someone inside the room

LEDs:

- YELLOW: The device is not yet connected to the network
- RED: Presence detected inside the room
- OFF: Nobody present inside the room

Random Presence Simulation: Every *random* seconds toggle presence status

CoAP Observable Resource:

- `/presence`:
  - **GET:** get current presence value (T/F)
  - **EVENT:** Notify observers of the presence's status change

Luminosity Sensor

Luminosity sensor that simulates the luminosity value in the room by adding the external Luminosity to that of the smart bulbs

LEDs:

- YELLOW: The device is not yet connected to the network
- OFF: The sensor is connected to the network

Random External Luminosity Cycle: Every *random* seconds increase external luminosity by $[1,10]\%$, when it reaches 100%, every *random* seconds decrease external luminosity by $[1,10]\%$, when it reaches 0% it resumes the cycle

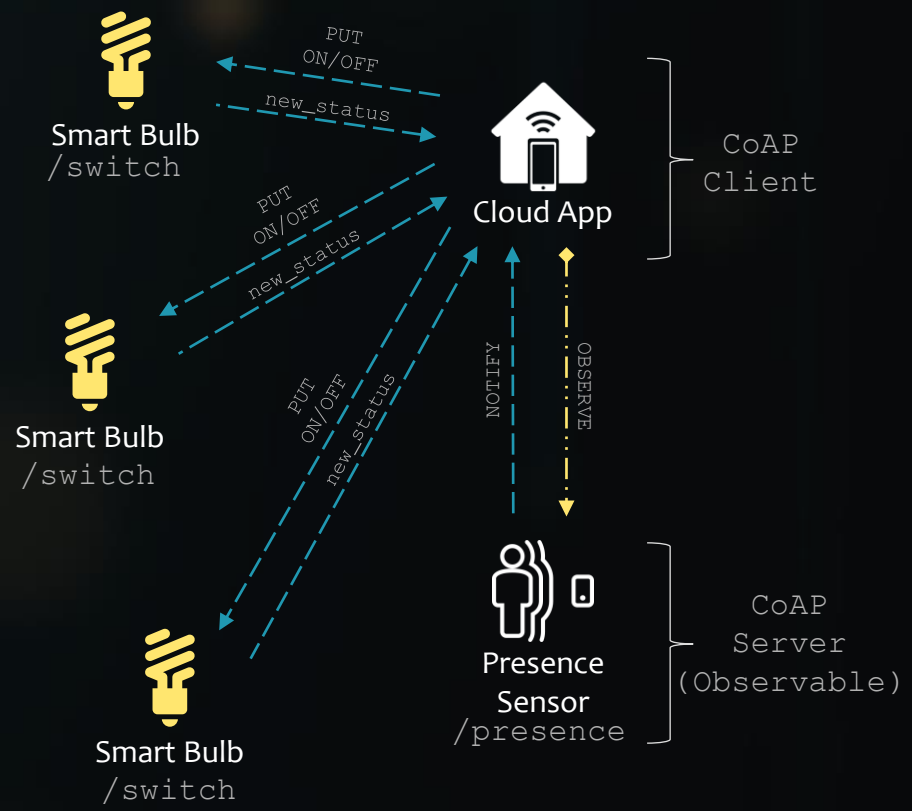
CoAP Observable Resource:

- `/luminosity`:
  - **GET:** get current luminosity value in the room
  - **PUT:** set bulbs mean luminosity value to simulate sensor perception
  - **EVENT:** Notify observers of the change in luminosity

ECO Mode

This mode turns on smart bulbs only when needed:

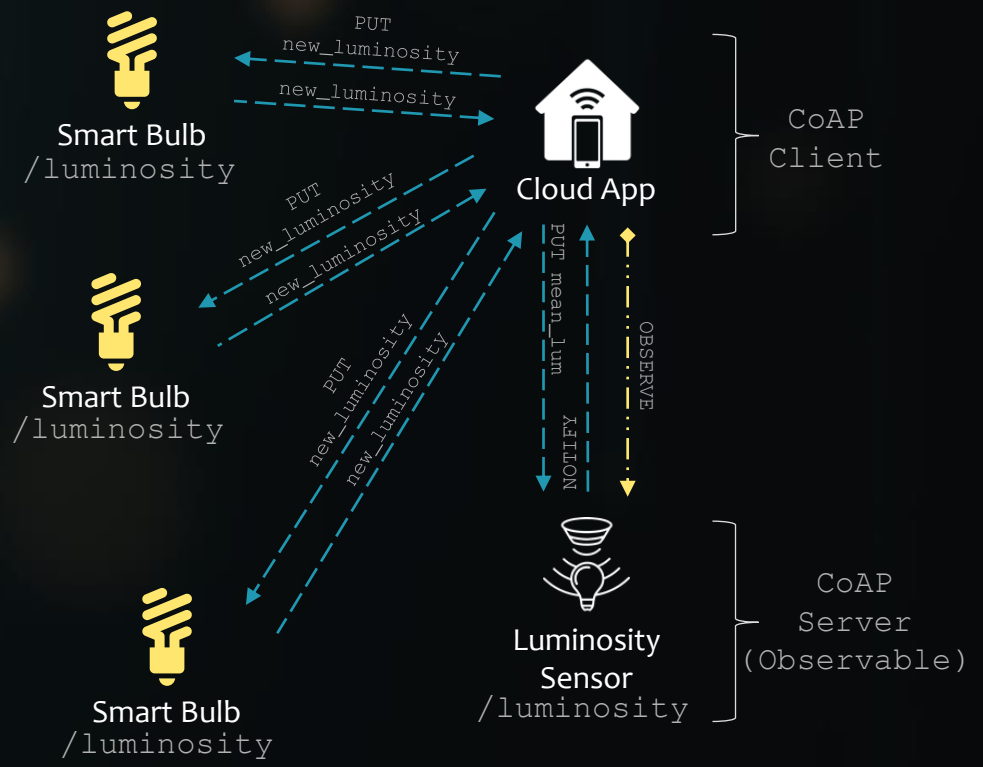
1. During presence sensor's registration, cloud application immediately starts to observe its `/presence` resource
2. If the sensor status changes, the application is notified with the new status
3. The application performs a `PUT` on the resource `/switch` of each smart bulb registered with it, indicating the new status
4. If the `PUT` is successful, the smart bulbs respond with their new status, in order to update the app values



Constant Luminosity Mode

This mode keeps the luminosity in the room constant:

1. During luminosity sensor's registration, cloud application immediately starts to observe its `/luminosity` resource
2. If the sensor status changes, the application is notified with the new status
3. The application calculates the new bulbs' luminosity to keep that of the room constant, after which it performs a `PUT` towards the `/luminosity` resource of all the smart bulbs indicating the new value
4. If the `PUT` is successful, the smart bulbs respond with their new luminosity value, in order to update the app values
5. In parallel, the application performs a `PUT` on the sensor `/luminosity` resource to allow it to distinguish between natural light and that generated by smart bulbs (coherence mechanism)



Luminosity Coherence Mechanism