Università di Pisa

Mobile and Cyberphysical Systems

IoT street lighting

Alex Colucci, Domenico Romano

2020/2021

Index

Problem **2**

Solution (conceptual description) **2**

Functional Requirements **2**

- Web app - Modes definition 2
- Web app - Cluster visualization/association 2
- Web app - Fault visualization 3
- Street lamps behaviour 3

Scenarios **3**

- Optional requirements/ future works 4

Solution proposed **4**

- Hardware 4
- Software architecture 5
- State 5

Planned demo **5**

Smart street lighting

Problem

The main problem is the waste of electricity due to the fact that usually the street lamps are turned on at sunset and turned off at dawn, remaining on at their maximum brightness. For this purpose, it is therefore inevitable to turn off the street lamps or reduce their brightness when people and vehicles do not pass, but at the same time guaranteeing safe situations.

Another problem concerns the maintenance of the street lamps. Usually, faults can only be identified through people's reports, and sometimes it is necessary to leave all the street lamps on during the day in order to have the repairers identify the faulty one.

Solution (conceptual description)

The idea is to allow the administrator, through a web app, to manage in a customizable way the various clusters of street lamps (groups of nearby street lamps), based on the specific characteristics of the area in which they are located.
Faults will also be notified to the administrator in real time via the web app.

[1] Motion sensors will be placed on the street lights, as well as a brightness sensor for each cluster. On the base of the environmental brightness and the parameters set by the administrator, the street lamps (if necessary) will turn on or increase their brightness when people or vehicles pass.

Functional Requirements

1. Web app - Modes definition

The administrator can create, modify or delete a mode.
In the first two cases he defines the parameters for that specific mode:

- A variable number of associations of range of luxes (environmental brightness) with specific percentages of illumination brightness (both in the cases when something is passing and when it is not);
- the number of following lamps to be switched on, on the base of the current position of the car/person and the number and the number of seconds which they must remain on.

2. Web app - Cluster visualization/association

The administrator can visualize the clusters information and he can choose the mode for every cluster.

3. Web app - Fault visualization

The administrator can visualize the list of unfixed faults.
Faults not yet visualized are highlighted.

4. Street lamps behaviour

See [1].

Scenarios

The administrator, in the web app, visualizes the list of clusters of his city, which have an id between 0 and 30. Specifically, to clusters from 20 to 25 (extra-urban fast-flowing road) and to those between 26 and 30 (city center pedestrian areas) a modality has yet to be assigned and defined. Then the administrator creates two new modes: "**extra-urban road**" and "**pedestrian area**".

In the **pedestrian area**, for safety and aesthetic reasons, the street lamps can never be completely turned off when it is dark, but they must be turned on at least at 20% of their maximum intensity, even if nobody is passing. Then the administrator sets the following parameters:

illumination table:

environmental brightness	*street lamp illumination - movement*	*street lamp illumination - no movement*
> 50 lux	off	off
3.4 - 50 lux	30%	10%
< 3.4 lux	100%	20%

following lamps: 1
illumination time: 5 (seconds)

On the **extra-urban road**, on the other hand, the street lamps can be turned off when it is dark and no vehicles pass, but having high speed and a need for greater visibility for road safety reasons, the number of street lights to turn on must be greater. So the administrator sets the following parameters:

illumination table:

environmental brightness	*street lamp illumination - movement*	*street lamp illumination - no movement*
> 50 lux	off	off
3.4 - 50 lux	30%	off
< 3.4 lux	100%	off

following lamps: 3
illumination time: 10 (seconds)

So the administrator assigns to the clusters 20 to 25 the "**extra-urban road**" mode and to clusters 26 to 30 the "**pedestrian area**" mode.

Then the street lamps will behave accordingly to the current environmental brightness and to the parameters set. They will also send messages to the web app in case of faults occurring.

Optional requirements/ future works

- Collection/processing of data for the purpose of traffic analysis;
- Warning of dangerous car overtakings.

Solution proposed

Hardware

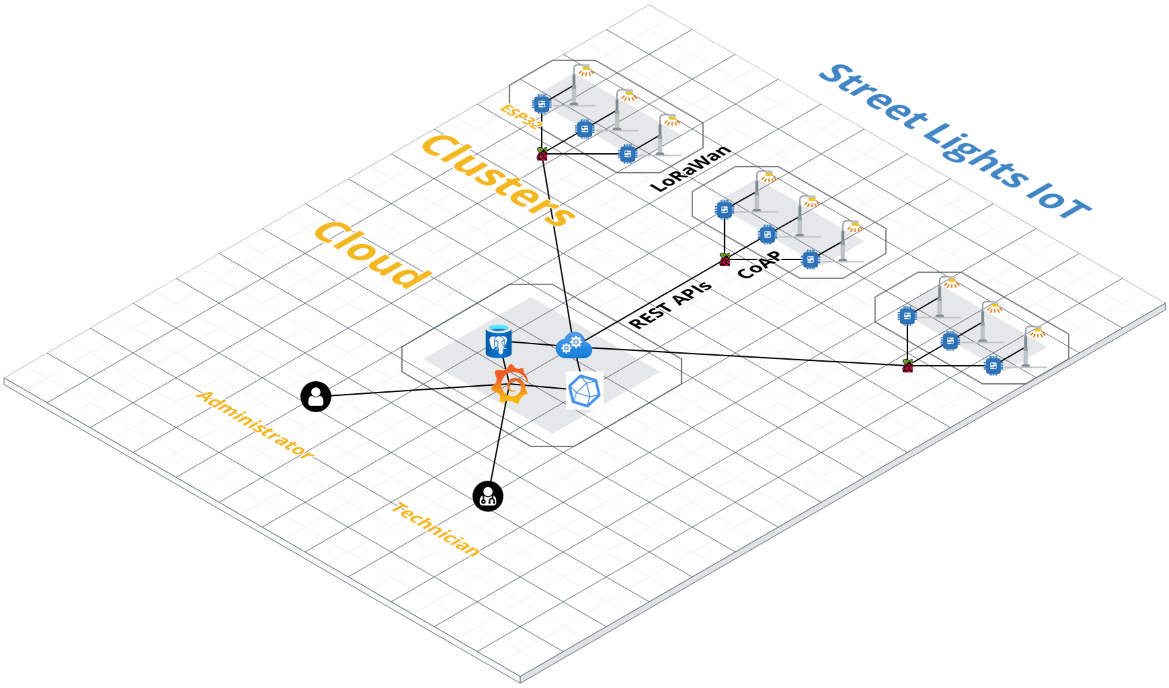
Street light

- ESP32
- Luminosity sensor
- IR LED sensor

Edge

- Raspberry

Software architecture



- The street lights that are equipped with an ESP32 microcontroller, are connected together with the Lo-RaWan protocol that allows up to 10 km. Another possible option that may be considered, could be the IEEE.802.11ah.
- The set of street lights communicates with the Edge by using COAP..
- The cloud receives data from the Edge and from the Administrator and can save the state over Timescale DB that offers an optimized solution for time series data over Postgres SQL.
- The Administrator can use the Grafana Dashboard to visualize data in real time and manage the cluster that he owns.

State

ADMINISTRATOR DATA: ID, Name, Password, Clusters, Modes

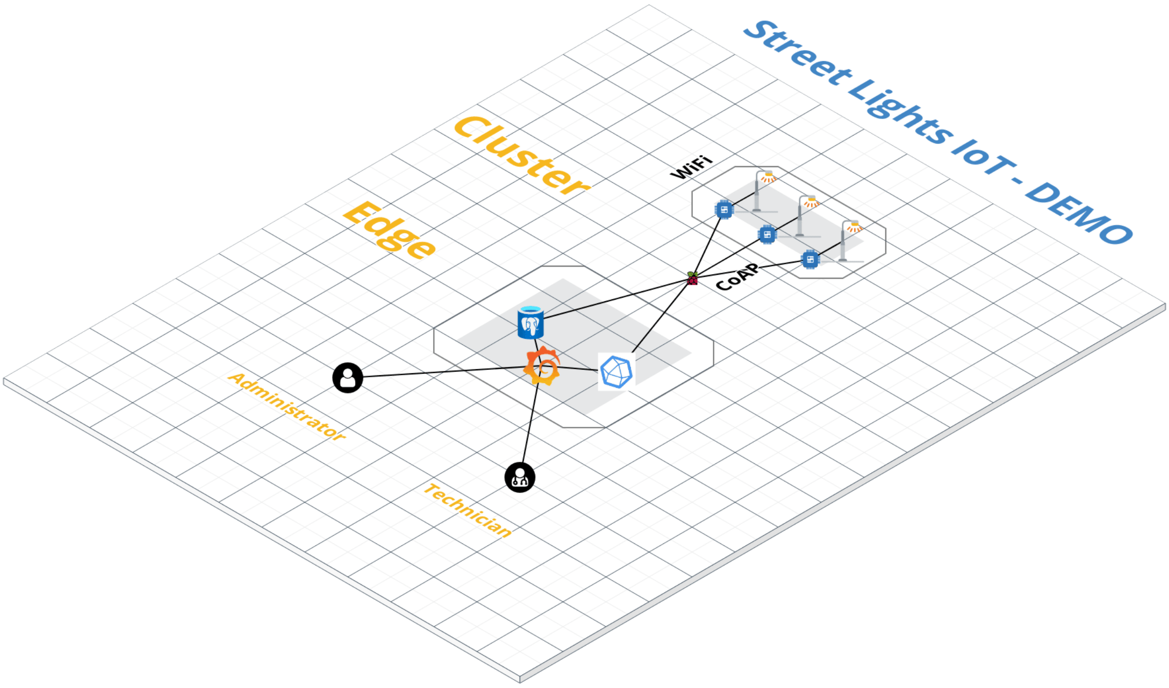
CLUSTER DATA: ID, Cluster name, Street lights, Expected mode

- computed: Number of street lights, Number of faults, Geographical area, Actual Mode

STREET LIGHTS DATA: ID, location (latitude, longitude), Actual Mode, Expected mode, Brightness (time, lux), Proximity (time, value)

MODE: ID, Name, Groups (Lux Range, Illum. with movem., Illum. without movem.), Following lamps, Illumination seconds

Planned demo



The way in which we intend to present the project is by using the following hardware:

- 3 Arduinos MKR 1010 Wifi with IR led sensor and led to simulate the behavior of the street lights.
- PC connected to the same Wifi network of the Arduinos behaving as edge.
- Brightness sensor on the top of the lamps to detect the environmental brightness.
- Photoresistor under the bulbs in order to measure if the street lamps illuminate properly and therefore if faults occur.
- Toy car.

The demo will use a simulation of data for a specific part of the communication even though there is a predisposition for CoAP calls to communicate with the edge.

Source code

https://github.com/Street-Lights-IOT/street-lights-iot