Distributed application of a set of smart parkings

Students	Camille FAUCHIER
	Sergio Arturo SILVA GONZALEZ
	Hussein RAMMAL
Supervisor	Dr. Georgios BOULOUKAKIS

Master 2 Computer Science For Networks

Fall 2022

Content

1. Introduction

In big cities like Paris, there are currently many difficulties finding available parking spots. This represents a waste of time for the car users because of the traffic (mainly in the downtown) as well as the energy and gas consumption, and also considering that sometimes there are some parking rules violations. This causes a lot of stress to drivers because they feel forced to look for a spot during a lot of time and many places.

Nowadays, with IOT technologies, it is possible to monitor, manage and control the parking space and services provided which help not only the managers and the owners of the parkings but also represent a good opportunity for the users to find these available spots.

Smart Parkings are built to solve the mentioned problems by using sensors, actuators devices, cameras. With those, the system will notify the drivers if there are any available slots, if not it will notify them of the nearest available parking. However, all the parking must interact with each other to share this information among them but maybe each one of them is using a different protocol. So, to resolve that the middleware concept should be present.
Here are the services to implement:

- Display number of spots availables in a given parking and if possible their exact localisation - **S1** ;
- Indicate the nearest parking with available parkings spots - **S2** ;
- Manage reservations and automate them with the license plate - **S3** ;
- Check if people are parked properly, and fine the owner if they are not (using cameras) - **S4**

2. Project Modelisation

Figure 1 : Campus de palaiseau

We focus on 5 parking areas on the campus. We assume that they are equipped with some IoT devices.

Parking	Number of parking spots	IoT devices
P1	350	cameras to check available spots, deals with reservation, well parked
P2	400	devices at the gate that count entry/exit
P3	100	one device per spot that indicates the status (free/occupied) of the spot
P4	215	devices at the gate that count entry/exit
P5	10	one device per spot that indicates the status (free/occupied) of the spot

Table 1 : Description of the parking and their equipments

We focus on three type of IoT devices :

- Sensors that can give the status of a slot, there is one sensor by spot.
- Sensors that can count the number of cars in a parking, by counting the entry/exit of the parking
- Cameras that send images of slots.

3. Used Technologies

Smart parking system is a very important system to facilitate controlling, managing the services provided by the parking. So, it is important that the parking must have the most efficient function and performance, for that we used some technologies that will be explained in details why we used them like message queuing service that is lightweight and a MQTT server that handles clients subscribe/publish and that is most scalable server.

The technologies used to accomplish the proposed solution are as follows:

3.1. MQTT

We chose Mqtt in our project to see the available spots, compute the available spots in the parking, image recognition that send the state of the parking so it is like a minimal code footprint which in our case it is efficient and more performant. No matter how many devices are linked to the internet, it is the perfect approach to post and subscribe to a certain topic since it is lightweight. Also, Mqtt is reliable because of the quality of services that assures message delivery.

3.2. EMQX

In our project, we chose to use EMQX MQTT broker, because it provides the cluster mechanism spread equally the load across the workers where there is load-balancing and high availability, if we need another node, it easy to scale where adding extra worker to the cluster allows for capacity additions without requiring any downtime.

3.3. Telegraf - Influxdb - Grafana

- InfluxDB is a data store for any use case involving large amounts of timestamped data.

- Telegraf is a plugin-driven server agent for collecting and reporting metrics. This can collect data from a wide variety of sources. In our project, it subscribes to the relevant topics in the same way a user would, and them write the collected data into InfluxDB.

- Grafana makes it easy to create dashboards for displaying data from many sources, particularly time-series data. It will be used as a front-end part for the demo.

4. Software Architecture

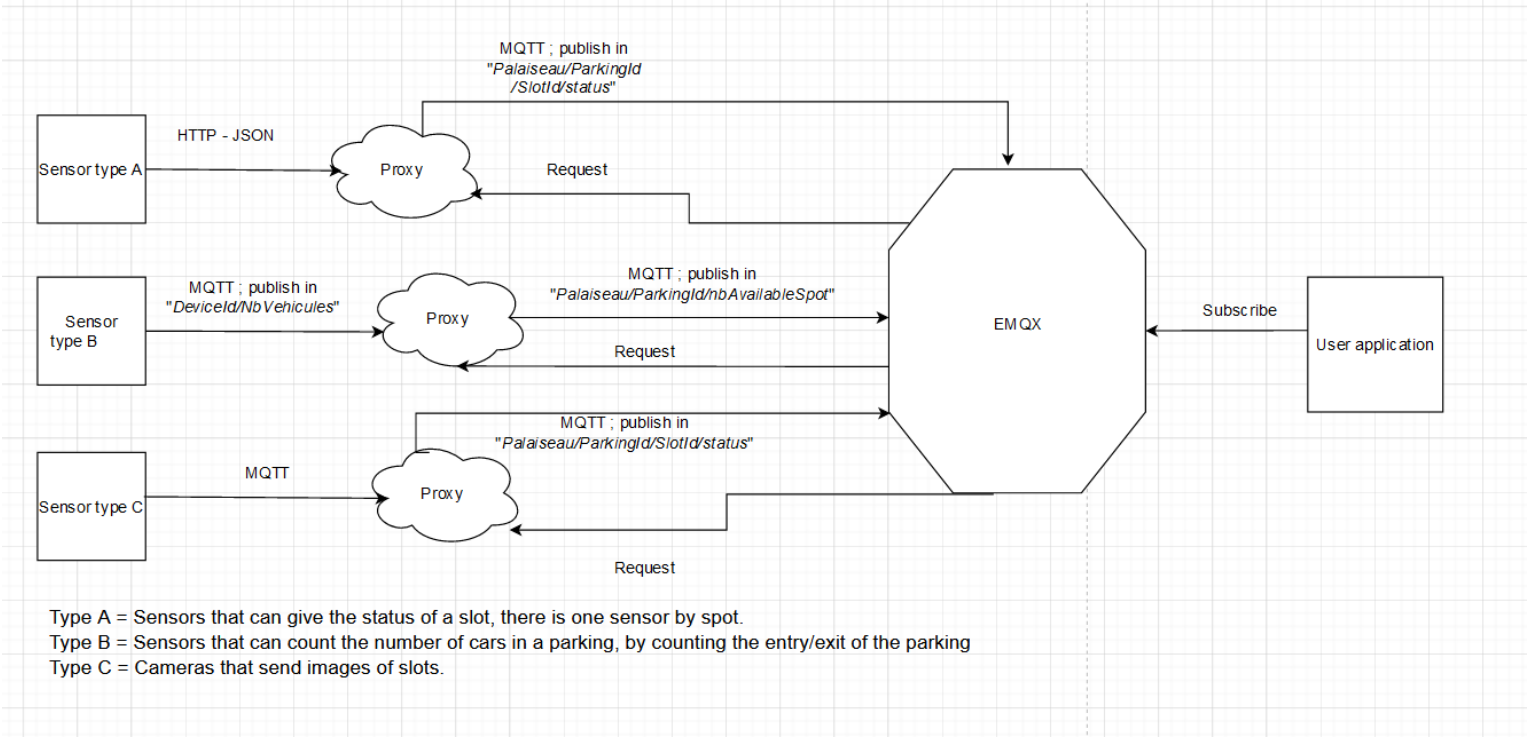

figure 2 : System architecture

The figure above describes our system architecture. Sensors use different protocols to communicate. It's up to the proxies to harmonize data, and publish to the right topics. Mqtt Broker Emqx stores the data into a database.
We constructed our topic this way :

- "/Palaiseau/**$ParkingId/**nbAvailableSpots"
- "/Palaiseau/**$ParkingId/SlotId/**status"

Then a user can subscribe to this topics if they want information about spot / parking.

Since we don't have real devices, we mocked them.