Project 01 [20 points] Deadline November 25th

Introduction

Nowadays, Internet of Things (IoT) applications and solutions have emerged and are gaining importance in the modern world.

IoT devices, spread all over the world, are capable of collect a large amount of data in many different fields: home, mobility, energy consumption, real-time monitoring, etc.

With all this data, we can create advanced analytics, using various types of modeling: statistical models, neural networks, etc… and provide the corresponding summaries and verified information to provide the necessary knowledge to improve the way things are used.

However, the large scale in the amount of data requires the use of distributed computing to process all the data intake in an acceptable time.

Project Proposal

In this project we will simulate the creation of an IoT platform to gather information, optimize, and control the climate and comfort of remote homes.

In this project, we will have **several users** that have contracted your IoT platform.

You have installed a **set of sensors at their home**, connected through a **Gateway** using **MQTT**.

The sensors will provide **timeseries information to your cloud services**, via a public MQTT, where you will perform data analytics and aggregations and provide the necessary signals to control the devices.

In this project we will simulate the readings of data at 1/60 scale, meaning that **1 second in the real world will be 1 minute in the simulation**.

<u>**User's home:**</u>

Each user will have a set of **IoT *devices*** at home that will periodically send information to the system.

The sensor's simulation implementation can be found in the project's files.

Notice the ***devices*** are generic and have no configuration other than the specified in the README **it is not allowed to modify the sensor.**

Each user will have a ***Gateway*** installed at their home. The *gateway* is used as a hub of communication between the ***Devices*** and the ***Cloud****.*

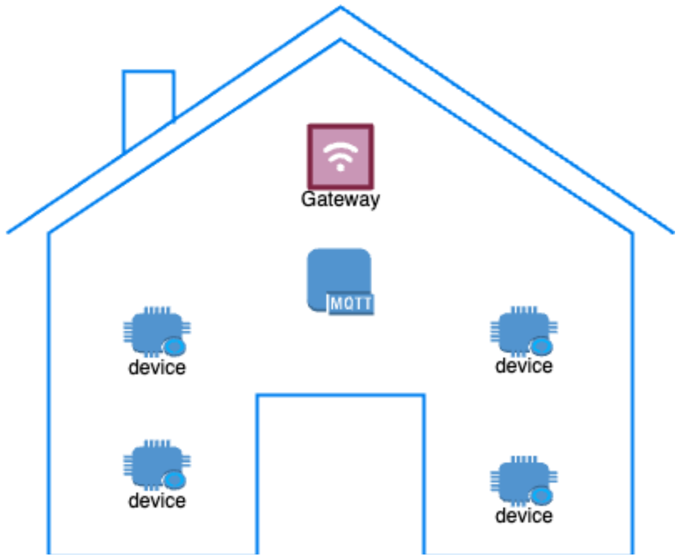

Figure 1 Home architecture

<u>Cloud</u>

The data of the ***gateway*** will be sent using MQTT to the ***cloud***. Then the data will be introduced to a Kafka system to apply some pipelines.

P1 – Save Raw Data:
This will only save the raw data, exactly as it was read by the sensor. Useful to review and recalculate in case of error.

P2 – Save Clean Data:
This will save the data after it is cleaned by a cleaning process. The cleaning is performed by removing values of exactly 100 degrees

P3 – Actuate:
This service will analyze the clean data and decide if a command have to be sent to the user.

- Start the heat pump when the temperature is lower than 20
- Stop the heat pump when the temperature is over 25

P3 – Aggregate:
This service will analyze the clean data and calculate aggregations (mean) for user sensors (obtain the user mean temperature of the whole home)

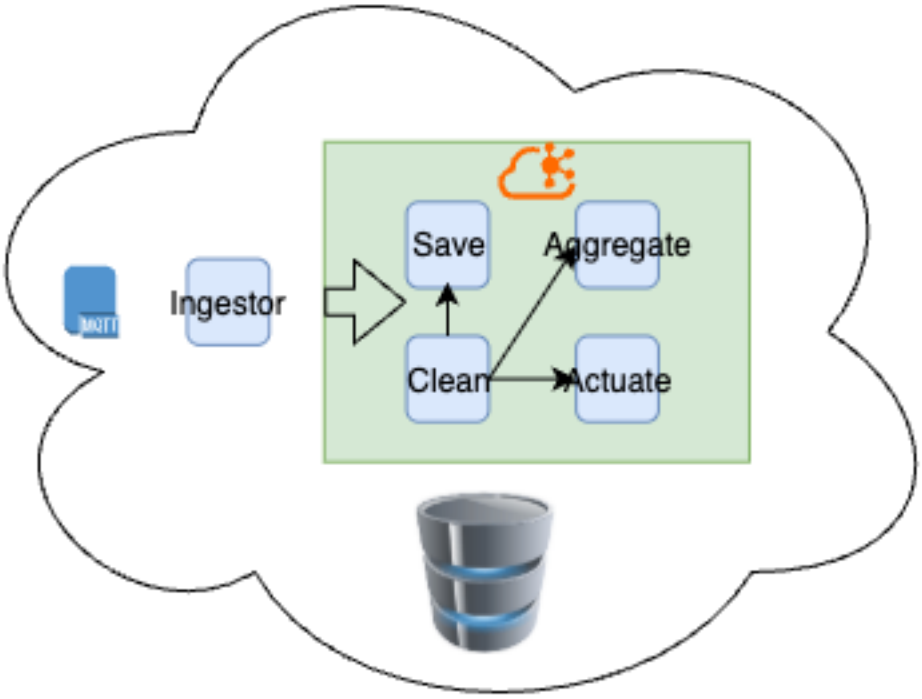

Figure 1 Cloud architecture

Architecture overview

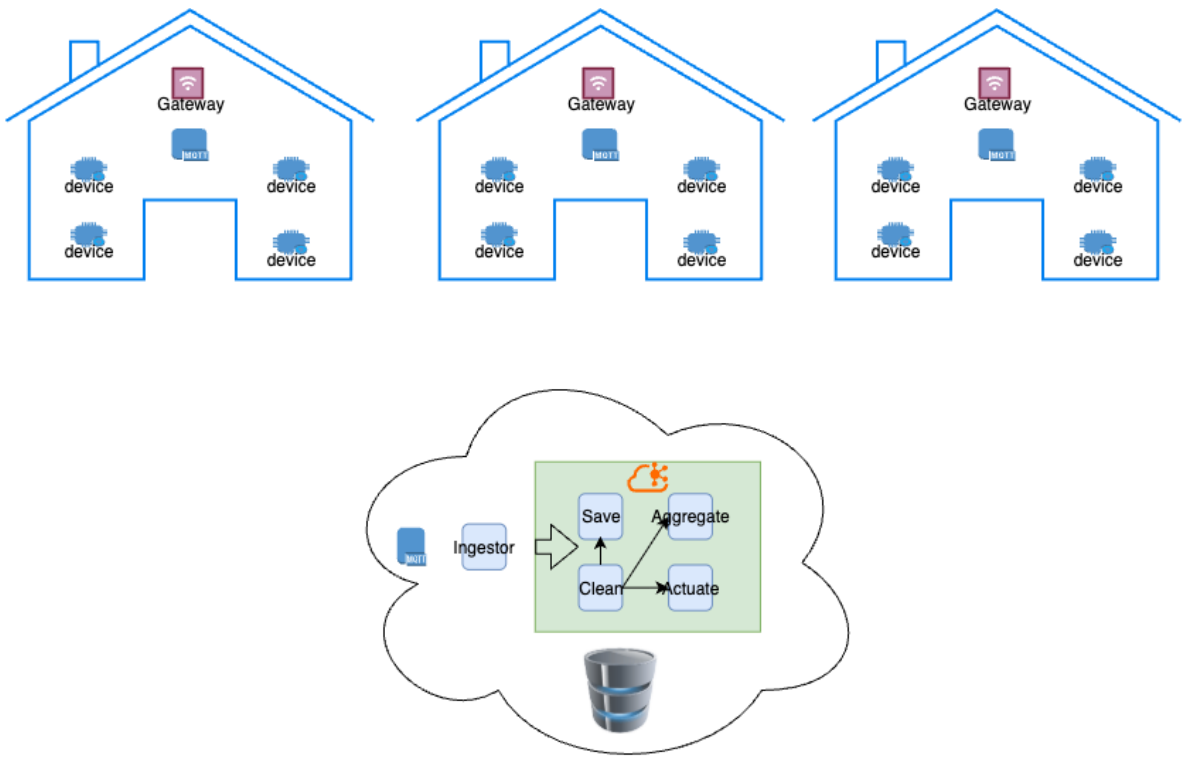

Figure 1 Project architecture

1st level Functions [50%]

- The devices are implemented following the project specification.
- The Gateway will receive the information from MQTT and will send it to the Cloud Service.
- The Save pipeline should be implemented and store the data to an Influx database (25%)

2on level Functions [25%]

- The Save clean pipeline should be implemented, cleaning the data and saving it to a new clean element in the database (5%)
- The Actuate pipeline should be implemented, sending the data to the device in order to actuate. (5%)
- The Aggregation pipeline should work and store the data to the database. (15%)

3rd level Functions [25 %]
- Add a sleep in the actuate to simulate a slow process, solve the Backpressure generated.

- Set the cloud services in minikube (mqtt, ingestor, kafka and pipelines)

Quality Features:

- The deliverable is well redacted and includes the correct reasoning and explanation of why it is implemented this way.
- The code is well documented and has clear instructions on how to run it

Considerations

You face the development of a distributed application. Your solution should provide correct implementation of the functionalities and consider efficiency, fault tolerance and performance.

Deliveries

Report content

Create a report with the following contents:

- Description of each component.
- Summarize the main design decisions done in this project such as technology used, database used and data model.
- A well detailed documentation on how to run the project.

Instructions

Work in pairs to develop the distributed application project. Submit a report with the contents specified above. The final source code and deliverable should be upload to the virtual campus. Do not forget to indicate in the report the time spent on the activity. **There will be a face-to-face presentation with the professors. In the case a group deliver the project after the deadlines there will be penalizations on the score.**