## Project 01 [28 points] Deadline December 22 th

## Introduction

Nowadays, 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.

**If some students have a custom project proposal different than the one presented here, they can explain it and the teacher will verify if it can be used for the subject. (A different proposal will be always graded better due to the complexity of working in a real environment)**

## Project Proposal

In this project we will simulate the creation of an IoT platform to gather information, analyze, and control the devices from domestic homes.

In this project, we will have different users that will be identified using a unique ID. In the following table you can find the list of users:

*Tabla 1: Users*

| User | ID |
|---|---|
| **Albert** | |
| **Tommy** | |
| **Dakota** | |
| **Tifany** | |

Each user will have a set of IoT devices at home that will periodically send information to its gateway by means of MQTT. In this project we will simulate the readings of data (you can choose the best way to simulate it).

*Tabla 2: Sensors*

| Sensor/Actuator | Type | Value range |
|---|---|---|
| **Light bulb** | Actuator | 1/0 (discrete) |
| **Presence sensor** | Sensor | -10~110 (continuous int) |
| **Temperature sensor** | Sensor | 15~30(continuous int) |
| **Heat Pump** | Actuator | 15~30(continuous int) |

The sensors will send information about the readings, these readings will be the data between the max and min values specified.

The actuators will send their state information and receive the order to change its state.

The gateways will act as Kafka producers, sending the data received by the different sensors and creating different pipelines. For these pipelines we will create the following microservices:

*Tabla 3: Microservices*

| Microservice | Function | |
|---|---|---|
| **Save** | Saves the information to the database | |
| **Clean** | Cleans the data following simple rules | |
| **Actuation** | Detects if actuation is required | |

**SAVE**: Will save the information received to the correct place in the database.
**CLEAN**: Will remove outliers following the rules:

- PresenceSensor (-0 or +100)
- TemperatureSensor(-18 or + 28)

**ACTUATION:** Will actuate on the devices following the rules:

- PresenceSensor-> LightBulb: +50 -> 1 / -50 ->0
- TemperatureSensor -> HeatPump: 18-20 -> 20/ 24-28 -> 24

That will conform the following pipelines

*Tabla 4: Pipelines*

| Pipeline | Microservices | |
|---|---|---|
| **Save Raw Data** | Save | |
| **Save Clean Data** | Clean, Save | |
| **Actuate detection** | Clean,Actuate | |

The following image shows a diagram of the final architecture for one user.

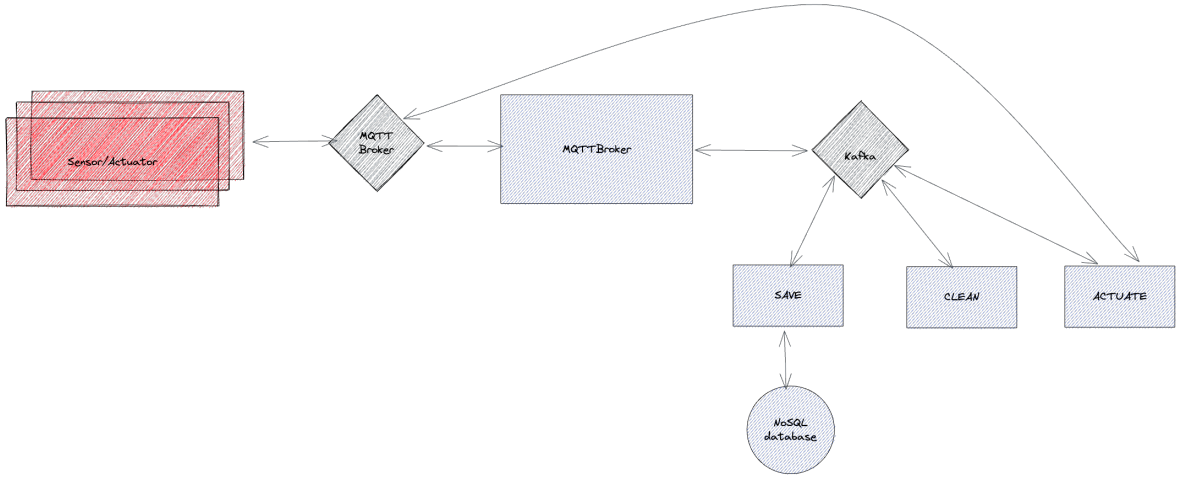


*Figure 1 Diagram of architecture*

## 1st level Functions [50%]

- The Sensors can be simulated with dockers. It will generate the readings in the values specified every second, the status will be reported onChange. All data will be sent using MQTT (25%)

- The Gateway will receive the information from MQTT and will send it to the Kafka Service.
- The Save pipeline should be implemented and store the data to a database of your choice (25%)

## 2on level Functions [30%]

- The Save clean pipeline should be implemented, cleaning the data and saving it to a new clean element in the database (15%)
- The Actuation pipeline should be implemented, adding the MQTT connection and send the actuation back to the gateway. The actuations can be implemented by just printing the new state on the output.

## 3rd level Functions [10 %]

- Create a docker-compose file that contains all components required. And All the steps to configure the dockers
- Add authentication to Communications in MQTT and KAFKA
- Distribute pipelines so no botelneck is created by

## Quality Features [10%]

The next features provide extra points.

- Data storage is done considering optimality
- The deliverable is well redacted and includes the correct reasoning and explanation of why it is implemented this way.
- Use consumer groups to read the same data for different purposes

- Reuse the microservices in order to keep the functionalities in one place
- A docker-compose file is delivered that runs the project automatically

## Considerations

**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.

You can choose the technology to develop the project, but all decisions must be justified. The technology you choose must consider facilities to efficiently develop distributed applications with access to remote objects or services, etc. The technology chosen cannot be an excuse neither to satisfy the project requirements nor for a correct implementation of the functionalities.

## 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. It is better if all modules are run using a single docker compose.
-

## 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.**