1 Name of Use Case

Name of the Use Case	**IoT platform for Smart Home**
Group	24
Version No.	V1.1
Submission Date	11/01/2024
Team Members (with student ids)	Ali Mohammad Alizadeh (308885) — s308885@studenti.polito.it Mohammad Eftekhari Pour (307774) — s307774@studenti.polito.it Mehdi Rajaee (308497) — s308497@studenti.polito.it Hamid Shabanipour (314041) — s314041@studenti.polito.it

2 Scope and Objectives of Function

Scope and Objectives of Use Case	
Scope	The proposed IoT platform aims to provide additional services for Smart Homes for having a control over air conditioning of environment.
Objective(s)	The expected results include: ● Enable precise control over environmental conditions (including temperature and humidity) for a consistently comfortable living space. ● Gather Historical Data for in-depth Statistical Analysis and informed decision-making. ● Monitor Home Environment Conditions, identifying patterns and fluctuations over the previous periods. ● Implement both Manual and Automatic Commands for Air Conditioning Devices.
Domain(s)	Smart Home
Stakeholder(s)	Home Inhabitants, Air Conditioning System
Short description	The proposed IoT platform integrates various IoT devices to optimize air-conditioning systems, offering control strategies to minimize energy consumption and enhance overall living conditions. The platform provides unified interfaces (REST and MQTT) to seamlessly integrate the home into a Smart Building, allowing for flexible Demand/Response policies. End-users gain detailed insights into their environment conditions. Summarizing, the main features it offers are: • Monitoring of House temperature and humidity. • Control strategies for temperature, humidity, and air conditioning systems. • Unified interfaces (REST Web Services and MQTT queues) to support Demand/Response scenarios. • End-user applications for an enriched experience, offering detailed information on the current environment. • Collection of Historical Data for future Data Analytics and Decision-making. • Comprehensive Historical data and Data Analytics support.

3   Diagram of Use Case
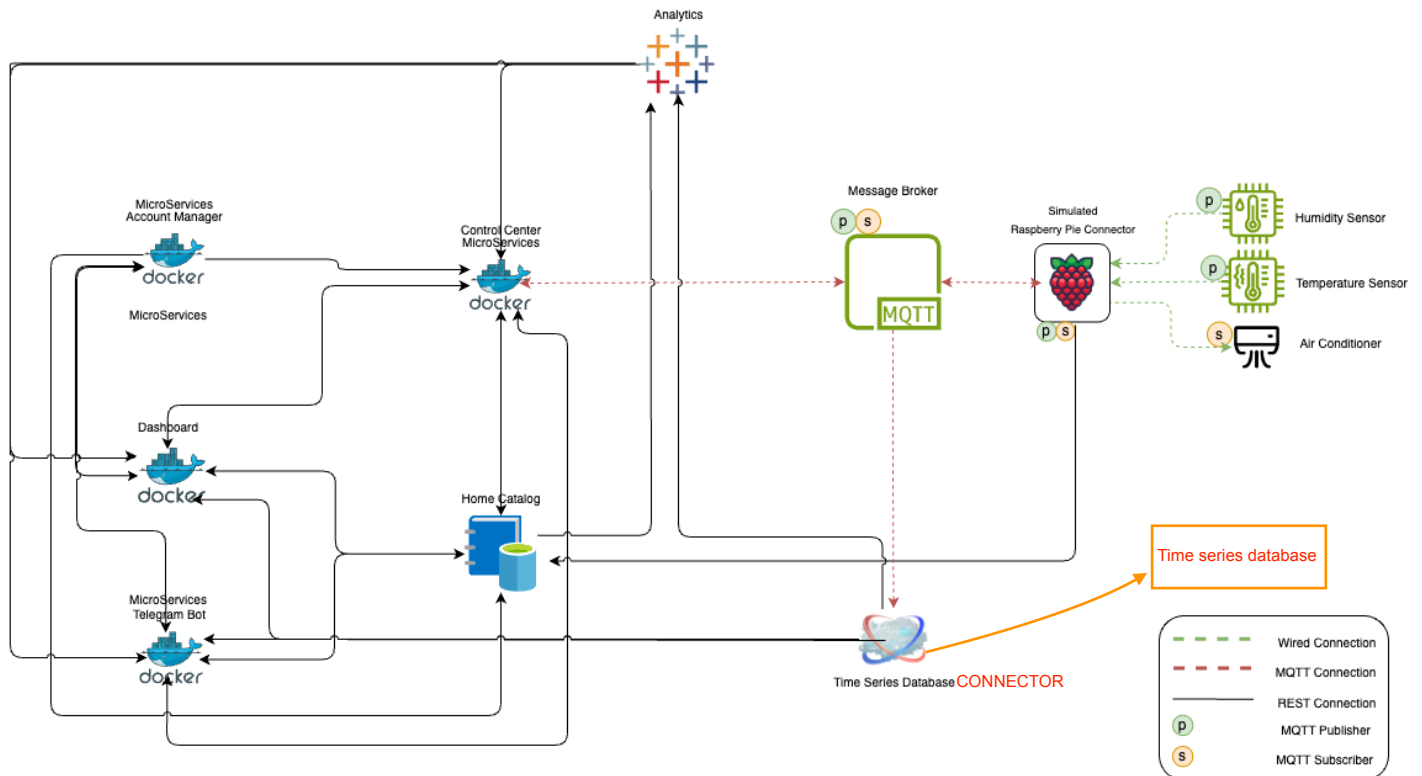
4   Complete description of the system

The proposed IoT platform for Smart Home follows the microservices designing pattern. It also exploits two communication paradigms:
  i)   publish/subscribe based on MQTT protocol
  ii)   Request/response based on REST Web Services.
In this context, some actors have been identified and introduced in the following:

● **Raspberry Pi Connector** is used for connecting humidity and temperature sensor to provide information about the environmental conditions in order to monitor and maintain the appropriate conditions required for inhabitants. Another important task is based on the MQTT protocol: working as a MQTT publisher it is able to send information about environment data (from humidity and temperature sensors), progressively analysed and finally showed on the Web-Dashboard. Also, Manual or Automatic Commands will be sent to Air-Conditioning system to be set with needed information or be turned Off and On.

● **Data Analytics,** one of the uses of historical data is having some analysis over timeseries and make predictions for future home environment, in this case we will use timeseries such as hourly temperature and humidity, hourly comfortable setting for Airconditioning system, also considering the manual command of the user to predict better configurations for Air-Conditioning system, to implement this feature, Moving Average over the data or ML algorithms like Regression or ARIMA model can be used since we have timeseries. In case of using Thingspeak, it can handle visualizing data and performing queries over the data and giving back the resuls and otherwise this
otherwise

microservice will handle the calculation over the data, for example it can provide dashboard with weekly or monthly average of temperature and humidity.

- **The Home Catalogue** works as service and device registry system for all the actors in the system. It provides information about end-points (i.e., REST Web Services and MQTT topics) of all the devices, resources and services in the platform. It also provides configuration settings for applications and control strategies (e.g. historical Data, list of sensors and actuators). Each actor, during its start-up, must retrieve such information from the Home Catalogue exploiting its REST Web Services. So, in a nutshell it is a service and user/house catalogue.

- **Sensors and Actuators,** two sensors namely Humidity and Temperature is used to get the environmental data which will be sent over MQTT microservices. Also, Air Conditioning devices are integrated to be turned off and on depending on the data provided by analytic service for the user and it could be done manually by the user.

- **Message Broker** provides an asynchronous communication based on the publish/subscribe approach. It exploits the MQTT protocol.

- **Control Center,** is Responsible for communicating With Analytics service and other REST APIs to send MQTT messages and receives data provided by MQTT communication and send them to other microservices. based on the information's taken from the sensors, is a control protocol created to maintain the ideal conditions to provide consistent appropriate environment. The tool and the sensor work with MQTT communication, like subscriber and publisher, comparing the data generated by the sensors to the specifications assigned by the home owner in order to manage the air condition of the area. So briefly it takes the commands from REST APIs and send them through appropriate MQTT connection. Also, if humidity and temperature get out of a specific range it will send a notification to the user to make them aware of the change. Also, automatic commands are run here on a schedule or other methods.

- **Time Series Database,** in order to store sensors time series measurements, we will use ThingSpeak or InfluxDB.
  - **ThingSpeak** allows us to aggregate, visualize, and analyse live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices or equipment. Execute MATLAB code in ThingSpeak, and perform online analysis and processing of the data as it comes in.
  - **InfluxDB** is an IoT-friendly database, specifically designed for seamlessly storing and querying time-stamped sensor measurements. It excels in managing dynamic data streams from IoT devices, especially sensors, where accurate timestamps are crucial for contextual analysis and decision-making. Its optimized architecture ensures efficient handling of continuous, high-frequency data updates typical in IoT scenarios, making it a preferred choice for applications demanding real-time insights and historical trend analysis based on sensor readings.

- **Account Manager,** is Responsible for managing users and houses corresponding to them and associated device.

- **Web-Dashboard:** Enables users to access their profile and generated data by the system will be demonstrated to the user through graphics and plots by using Rest APIs. Using REST web services statistical information about the status of the House will be retrieved in order to show it on Web-Dashboard. For Realtime plots there might be a need to use Thing Speak but this needs further study.

- **Telegram Bot** latest house environment and weekly temperature average also has the ability to send a command to Air-Conditioning device.

5 Desired Hardware

Device Name	Quantity	Needed for…

Raspberry Pi	1	Broker
DHT11	1	Collect temperature and humidity data
Relay	3	Turn on and turn off the air conditioner

Over View: a simple overview of flow and what happens from beginning to getting a result as an end user

- All microservices when they start running, they get their data such as URL and port from Home Catalogue which has the configuration. (REST)
- All the sensors will check their configuration and the endpoint they want to connect to from Home Catalogue. (REST)
- User registers using web-dashboard and creates a house and its sensors, for example a humidity and a temperature sensor and an air conditioning system. So now he is registered and can login also he can create a telegram bot connection to the house. Creating a user is done using Account manager microservice using REST connection and the user related data will be stored in Home Catalogue. Each service implements a CRUD operation.
- When the telegram bot connection is stablished, related data will be stored in Home Catalogue and the record related to the user will be updated. (REST)
- The user profile and data will be retrieved from Home Catalogue to be shown in Dashboard and Telegram Bot. (REST)
- When sensors start working, they first register themselves and status in Home Catalogue and retrieve their Topic from Home Catalogue (REST). And after that they will start sending measurements which is stored in ThingSpeak. (MQTT)
- Sensors current status and latest measurements and live chart of readings will be retrieved from ThingSpeak and will be shown in web-dashboard (REST). Telegram bot will simply show latest stored measurements and sensors status. (REST)
- Sensor's measurements will be sent over Message broker and will receive commands through Message Broker. How commands will be sent to RPI Connector will be explained.
- User can send Off/On commands and preferred temperature and Humidity configuration using Rest to control center and control center will send them using MQTT. The command with details like date, time and … will be stored in Home Catalogue for further use.
- The data stored in ThingSpeak and the commands stored in Home Catalogue will be retrieved using REST in Analytics service and then some calculation is done to predict the future preferred conditions and air conditioning system's Off/On status, and then Send though REST stored in Home Catalogue. Control service will read the configuration and can automatically send commands to message broker to be delivered to air conditioning system. Maybe the automatic commands are done 3,4 times a day or they are activated after a period of time of reading user commands and learning some pattern.
- Control service is also a subscriber to sensors measurements and controls a threshold for the readings and if there is an abnormality in readings, will notify the user. Also using this we can understand when a sensor turns off and can update the Home Catalogue.
- About the Command, there are two types:
    - o User Manual Commands: in web-dashboard or telegram bot a REST API is called to control center and then control center sends the request through MQTT.
    - o Automatic Command are done inside Control center directly using MQTT connection.