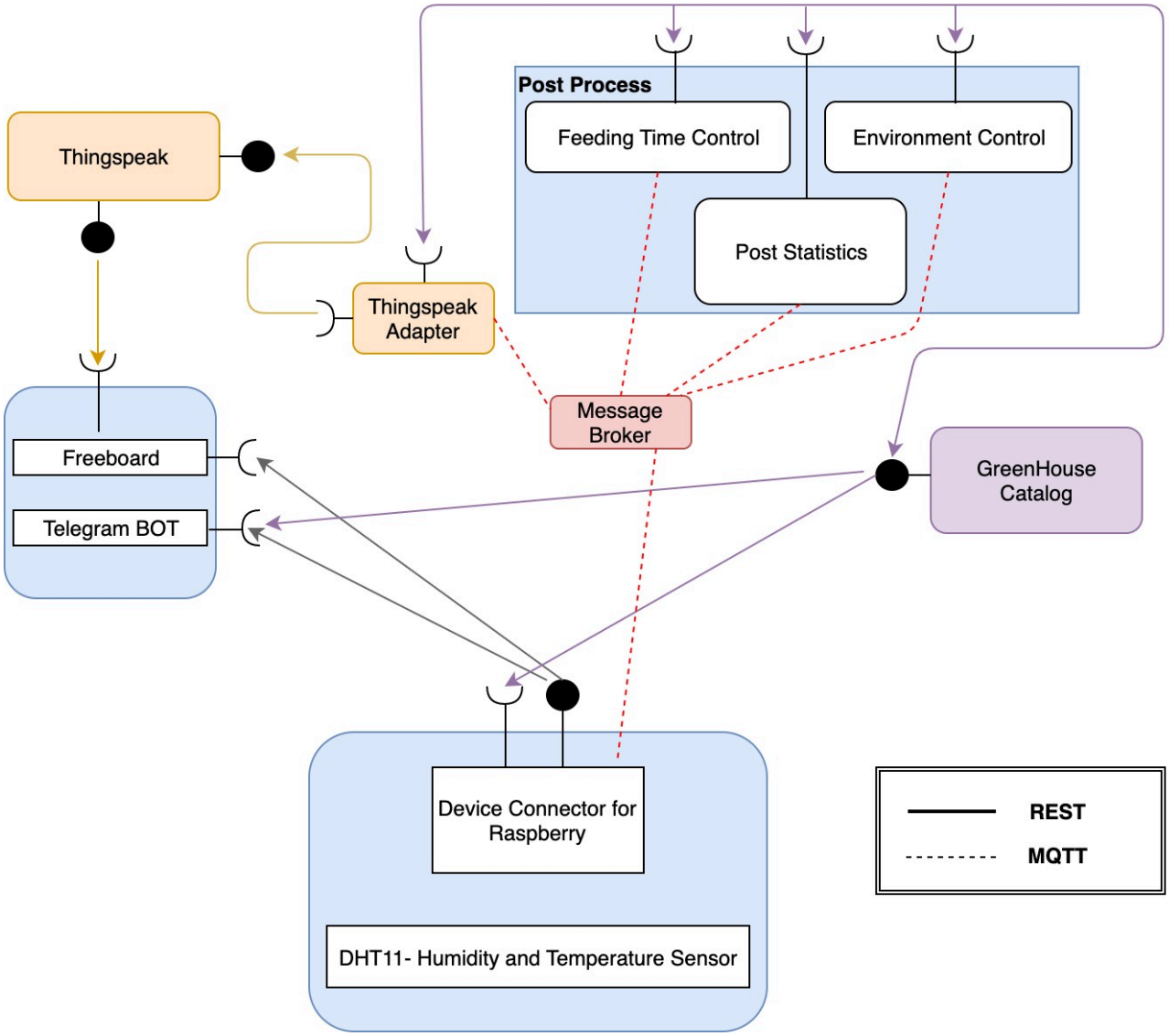
1 Name of Use Case

Name of the Use Case	**IoT platform for Smart Green House**
Version No.	v0.2
Date	21/12/2022
Team Members (with student ids)	s300424 Bekzod Sirojdinov, s277278 Umair Ahmed Memon, s296224 KUEJO Michel Sardou, s299284 Zhang Jiaxing

2 Scope and Objectives of Function

Scope and Objectives of Use Case	
Scope	The proposed IoT platform aims at providing services for a smart greenhouse management.
Objective(s)	The expected results consist of providing a smart control of appliances to minimize the energy waste and promoting production effectiveness via user-awareness applications.
Domain(s)	Smart greenhouse, Smart Agriculture.
Stakeholder(s)	Farmers, Energy aggregators.
Short description	The proposed IoT platform aims at making smart greenhouse. It integrates different IoT devices for managing appliances in agriculture environments. It provides control strategies for watering and heating systems to minimize the energy waste. The overall platform provides unified interfaces (through both REST and MQTT) to integrate the greenhouse into Smart greenhouse and Smart Grid environments. Hence, Demand/Response policies can be applied. Finally, the platform provides end-users with detailed knowledge of the greenhouse consumption for each appliance. Summarizing, the main features it offers are: ● Remote control of appliances; ● Strategies to Control Temperature, Soil humidity and Fertilizer intake; ● unified interfaces (i.e. REST Web Services and MQTT queues) available to enable Demand/Response; ● end-user applications for energy-awareness and monitoring purposes.

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 and ii) request/response based on REST Web Services.
In this context, ten actors have been identified and introduced in the following:

- The **Message Broker** provides an asynchronous communication based on the publish/subscribe approach. It exploits the MQTT protocol.
- The **GreenHouse Catalog** works as a 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., timers, list of sensors and actuators). Each actor, during its start-up, must retrieve such information from the Home Catalog exploiting its REST Web Services.
- The **Raspberry Pi Connector** is a *Device Connector* that integrates into the platform raspberry pi boards. The raspberry is equipped with temperature and humidity sensors to provide environmental information about the status of the greenhouse. It provides Rest Web Services to retrieve

environmental information (i.e., temperature and humidity). The Raspberry implements both MQTT and REST communication paradigms: the first one is used to send information about the microclimate as a publisher, while the second allows the user to retrieve the measurements of the sensors.

- **Feeding Time control** is a control strategy to manage appliances depending on time-schedules provided by the GreenHouse Catalog, it also enables users to control the limit of intake regardless of the timely manner already set by the Catalog. For example, it allows users to switch on the Fertilizer intake for a certain limit. It works as an MQTT publisher to send actuation commands to IoT Devices.
- **Environmental control** is a control strategy which controls temperature and humidity of the greenhouse, for example if the temperature of the greenhouse is lower than the threshold it turns on lights to heat and if the temperature is higher than the threshold it turns on fans for cooling. It is also responsible for irrigation in the green house depending on the humidity level of soil. It works i) As an MQTT subscriber to receive information from sensors; ii) As an MQTT publisher to send actuation commands to IoT Devices.
- **Post Statistics:** Its purpose is to monitor the information provided by the sensors to better understand the behaviour of the climate overtime. For evaluating how the climate change effects the health of plants to plan the next batch of plants according to the predicted data. It predicts the temperature for the following day by using past 3 days data. System can do heating or cooling and watering in advance, according to the prediction of temperature and humidity. For instance, if according to the prediction, the system finds temperature will go down to the low threshold, the system turns lights on in advance, in case the actual low temperature damage the plants. It works as an MQTT publisher and as a subscriber.
- The **Thingspeak Adaptor** is an MQTT subscriber that receives environmental measurements and uploads them on **Thingspeak** through REST Web Services.
- **Thingspeak** is a third-party software (https://thingspeak.com/) that provides REST Web Services. It is an open-data platform for the Internet of Things to store, post-process and visualize data (through plots).
- **Freeboard** is a dashboard to retrieve data from IoT devices and visualize them exploiting the REST Web Services provided by **Raspberry Pi**. It also exploits the **Thinkspeak** Web Services to import plots about environmental measurements.
- **Telegram Bot** is a service to integrate the proposed infrastructure into Telegram platform, which is a cloud-based instant messaging infrastructure. It retrieves measurements from IoT devices exploiting the REST Web Services provided by **Raspberry Pi**. It also allows users to send actuation commands to IoT devices again exploiting REST.

5 Desired Hardware components (only among those we can provide)

1.

Device Name	Quantity	Needed for…