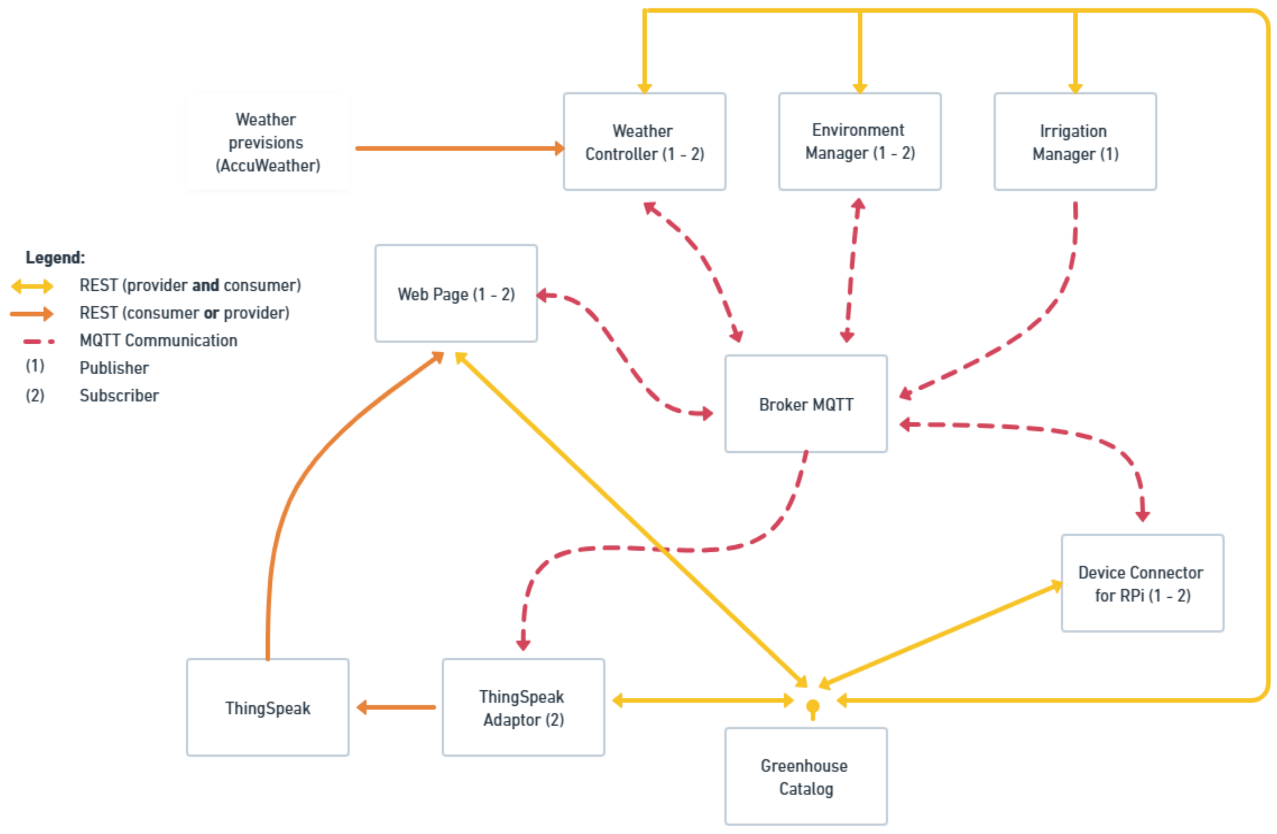
1 Name of Use Case

Name of the Use Case	**IoT platform for Smart Greenhouse**
Version No.	V2.0
Date	10/06/2023
Team Members (with student ids)	Impieri Matteo s302543, Bogoni Matteo s302917, Valentina Mendoza Zamora s301368, Alejandro Ayala Gil s294246

2 Scope and Objectives of Function

Scope and Objectives of Use Case	
Scope	The IoT platform aims at providing services for improve greenhouses management and make them smart.
Objective(s)	The expected results consist in providing an intelligent control of the growth conditions of the plants to optimize their production, also improving their quality, minimizing the waste of resources via user-awareness applications.
Domain(s)	Smart agriculture, Smart Building.
Stakeholder(s)	Farmers, Agriculture companies.
Short description	The objective of the proposed IoT platform is making smart cultivation in greenhouses. It integrates different IoT devices to monitor and regulate the fundamental parameters for plant growth. To control these parameters automatically and efficiently in order to reduce waste it integrates also some control strategies. The overall platform provides unified interfaces, through both REST and MQTT. Finally, the platform provides end users with detailed knowledge on the fundamental parameters along with statistics on their changes during the growth cycle of the plants. Summarizing, the main features it offers are: • remote control of temperature, humidity, light and window I. real time set parameters. II. commands executed by the devices. • control strategies for heating, humidification, irrigation and windows opening: I. default user-set parameters II. commands executed by the devices. • unified interfaces (i.e., REST Web Services and MQTT queues) available to enable Demand/Response • end-user applications for energy and parameter variations awareness (ThingSpeak along with the web page). I. data obtained by sensors. II. statistics and graphs about fundamental parameters and their behaviour/changes.

3 Diagram of Use Case


Weather previsions (AccuWeather)
Weather Controller (1 - 2)
Environment Manager (1 - 2)
Irrigation Manager (1)
Legend:
REST (provider and consumer)
REST (consumer or provider)
MQTT Communication
(1) Publisher
(2) Subscriber
Web Page (1 - 2)
Broker MQTT
Device Connector for RPi (1 - 2)
ThingSpeak
ThingSpeak Adaptor (2)
Greenhouse Catalog

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, nine 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 service and device registry system for all the actors in the system. It provides information about endpoints (i.e., REST Web Services and MQTT topics) of all the devices, resources and services in the platform, and moreover it can continuously register and refresh those services and resources. 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 Greenhouse Catalog exploiting its REST Web Services. It also allows you to edit the settings of the control strategies and managing users and greenhouses through the web page.
- The **Device Connector** is integrated in Raspberry Pi boards (at this phase is deployed into a Docker container). The boards are equipped with temperature and humidity sensors, to provide environmental information about the status of a greenhouse, and with relays to control the connected devices. Since no real devices are implemented at this phase their functions are fulfilled by a simulator (python script). It provides Rest Web Services to retrieve basic information about the system from the **Greenhouse Catalog**. It also works as an MQTT publisher, sending environmental

data, and as an MQTT subscriber to receive actuation commands from other actors that exploit the MQTT protocol (e.g., Control Strategies).

- The **Irrigation Manager** is a control strategy that manages the irrigation system in greenhouses according to the schedules received by the **Greenhouse Catalog**, through REST interface. It works as an MQTT publisher to send actuation commands to IoT Devices.
- The **Environment Manager** is a control strategy used to manage temperature and humidity devices depending on the schedules provided by the **Greenhouse Catalog**. It works i) as an MQTT subscriber to receive environmental data; ii) as an MQTT publisher to send actuation commands to IoT Devices if changes are needed.
- The **Weather Controller** is a control strategy used to control the opening of the windows of the greenhouses according to the schedules provided by the **Greenhouse Catalog**. It works as an MQTT publisher to send actuation commands to IoT Devices. It also provides Rest Web Services to retrieve information about the weather from an API provided by AccuWeather (https://developer.accuweather.com/).
- The **ThingSpeak Adaptor** is an MQTT subscriber which receives environmental measurements and upload 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). In our platform it is used as storage for the measurements and parameters which will be shown in the **Web Page**.
- The **Web Page** provides simple visual interfaces to monitor and change the parameters of the smart greenhouses (starting from the users and greenhouses registration) besides visualizing statistics on measurements and variations. It exploits the **ThingSpeak** web services to import environmental measurements and other parameters. It also allows users on sending real time actuation commands to IoT devices exploiting MQTT.

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

Device Name	Quantity	Needed for…