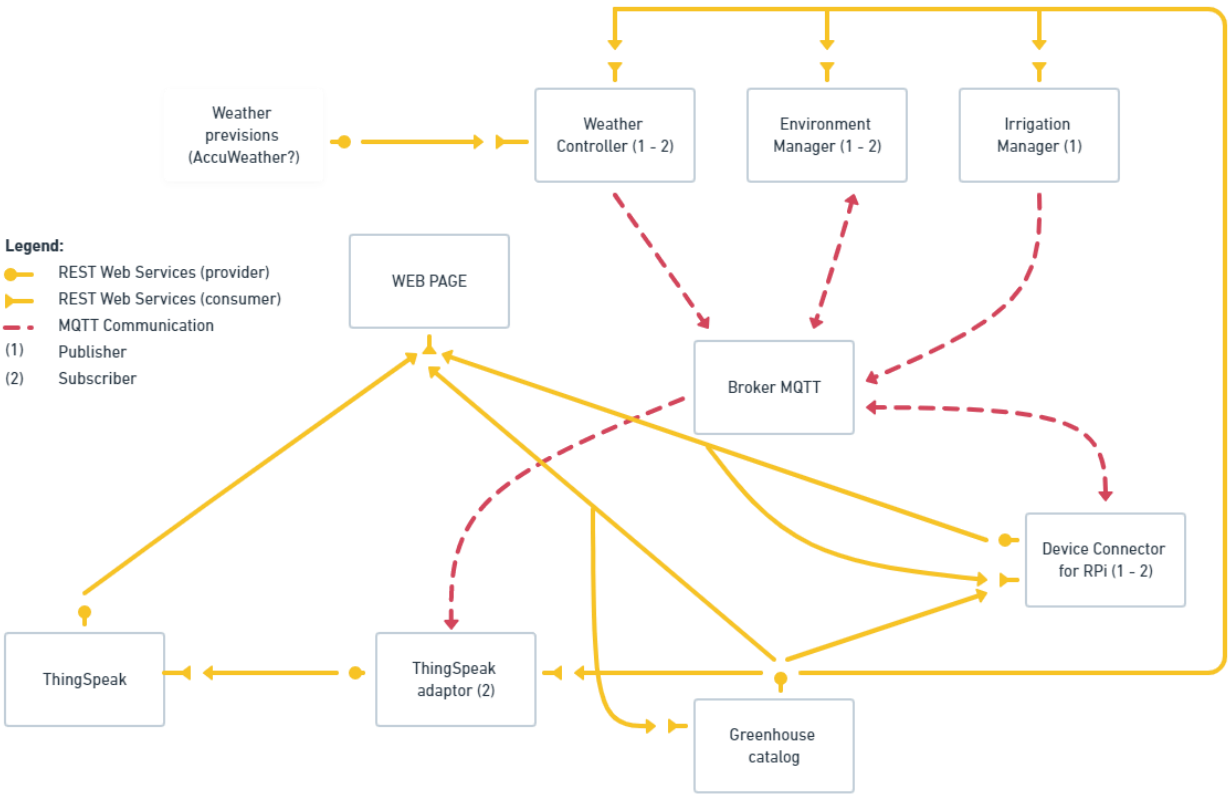
1 Name of Use Case

Name of the Use Case	**IoT platform for Smart Greenhouse**
Version No.	V1.0
Date	08/12/2021
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 a smart greenhouse management and improvement.
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 either control strategies. The overall platform provides unified interfaces, through both REST and MQTT. Finally, the platform provides end users with detailed knowledge of the consumption and statistics on parameter 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, lighting, irrigation and window systems 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). I. Data obtained by sensors II. Statistics and graphs about consumption and costs; control of parameters

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, 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 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. It also allows you to edit the configuration settings of the control strategies through the web page.
- The **Raspberry Pi Connector** is a *Device Connector* that integrates into the platform raspberry pi boards. Some boards are equipped with temperature and humidity sensors to provide environmental information about the status of a greenhouse, others are equipped with two relays to control the connected devices. It provides Rest Web Services to retrieve (from the catalog or web page) and change the status of the devices. It also works as an MQTT publisher, sending environmental data (every 5 minutes), 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 **Home Catalog**. Each greenhouse is managed by an instance of this strategy. It works as an MQTT publisher to send actuation commands to IoT Devices.

- The **Environment Manager** is a control strategy to manage temperature, humidity and light devices depending on the schedules provided by the **Home 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 to control the opening of the windows of the greenhouses according to the schedules provided by the **Home 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 that receives measurements on 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).
- The **Web Page** provides simple visual interfaces to monitor and change the parameters of the smart greenhouses besides statistics on consumption and variations. It exploits the **Thinkspeak** Web Services to import plots about environmental measurements. It also retrieves measurements from IoT devices exploiting the REST Web Services provided by **Raspberry Pi** and **Arduino Connectors** and allows users on sending actuation commands to IoT devices again exploiting REST.

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

Device Name	Quantity	Needed for…