1 Name of Use Case

Name of the Use Case	Smart Greenhouse
Version No.	v0.1
Submission Date	10/12/2022
Team Members (with student ids)	Ravera Stefano (271156), Redi Alessandro (310471), Scardi Alessia (317628), Volponi Federico (309709)

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 objective is to fully automate a greenhouse and release essential information to the owner thanks to the placement of sensors and user-awareness applications.
Domain(s)	Agriculture 4.0, Smart Grids
Stakeholder(s)	Farmers, growers, agricultural companies
Short description	The proposed IoT platform aims at automating the main functions of a greenhouse. It is possible to monitor the humidity, temperature, and CO_2 levels of the greenhouse through the integration of IoT devices. Moreover, in order to monitor humidity, sensors are inserted in the topsoil to check if irrigation is needed. If so, water is provided using an actuator. Finally, humidity and temperature are controlled by vaporizers and fans. The farmer has easily access to the data thanks to a third-party application (Freeboard). They also can receive real time information about the plants status through a telegram bot. The key features of the IoT platform are: • Humidity and temperature monitoring (for each batch of plants and for the whole greenhouse) • Air monitoring and fans control • Irrigation control • Plants variety datasheet • Applications for user-awareness • Prevision of future harvest time

3   Diagram of Use Case
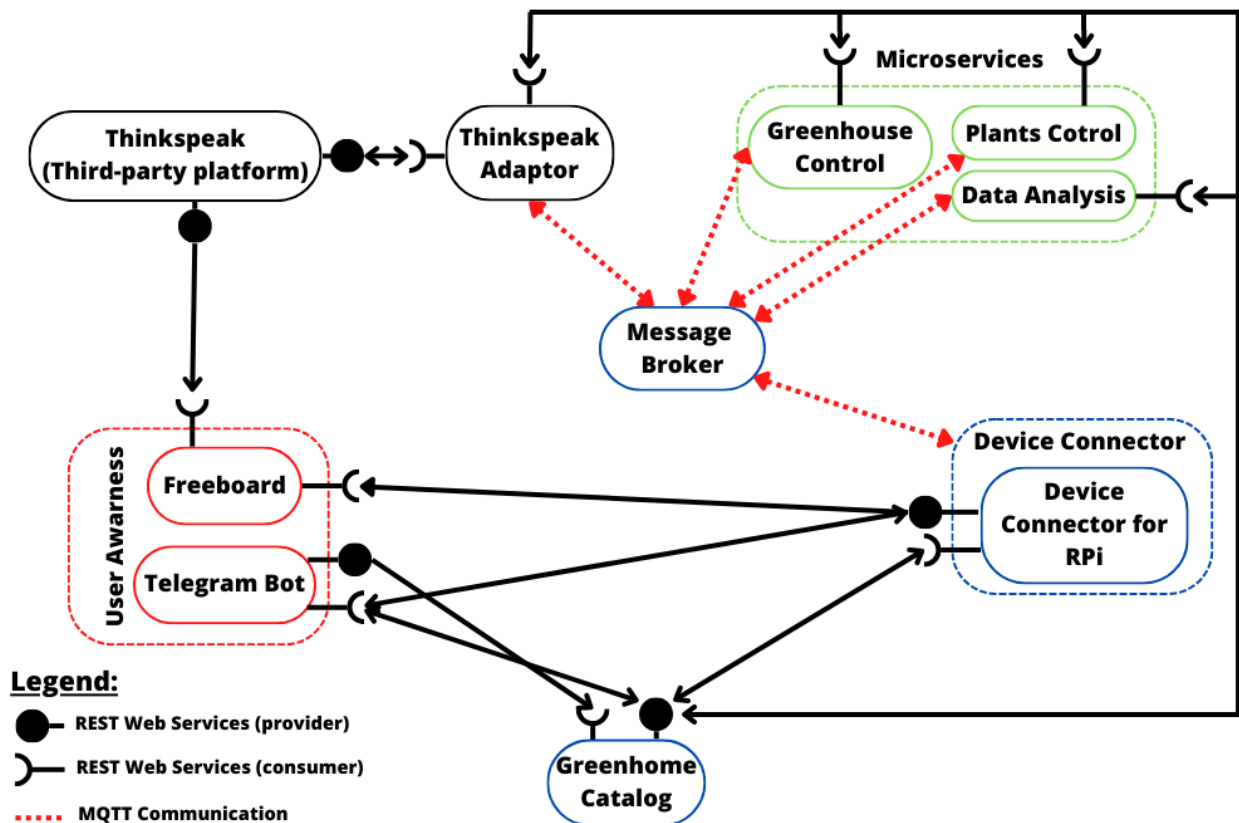
4   Complete description of the system

The Smart Greenhouse platform is designed using a micorservices approach. The communication between actors occurs through two communication paradigms:

(1) publish/subscribe based on MQTT protocol.

(2) request/response based on REST Web Services.

The actors involved in the IoT platform are the following:

- The **Message broker** is a software module that provides an asynchronous communication among applications. It exploits the MQTT protocol based on the publish/subscriber approach.

- The **Greenhouse Catalog** works as a registry system both for devices and services. It operates as an entry point for all the actors in the system. For the proposed IoT platform the Greenhouse Catalog contains: configuration of humidity and temperature thresholds used by the Greenhouse Control and humidity levels needed for each variety of plants used by the Plants Control. It also provides a registry of the IoT devices and holds information about the variety of the plants assigned to each batch of pots. It is implemented as JSON-based RESTful API.

- The **Raspberry Pi Connector** is a Device Connector that introduces the Raspberry Pi boards into the system. The Greenhouse is equipped with temperature, humidity and CO_2 sensors in order to monitor the Greenhouse's climate conditions and with humidity sensors for each batch of plants. They are all controlled by the Raspberry Pi which also manages the actuators controlled by the microservices. It exploits the REST Web Services in order to provide to the users the information they need, when asked, and it also works as an MQTT publisher to send sensor data and as a MQTT subscriber to collect actuation commands from the Control Strategy.

- **Thingspeak** is a third-party software (https://thingspeak.com/) that provides REST Web Services. It is an IoT analytics platform service that allows the user to aggregate, visualize and analyse data in the cloud.

- The **Thingspeak adaptor** behaves as an MQTT subscriber that links the selected outputs to the cloud platform using REST Web Services.

- The **Greenhouse Control** is a control strategy that manages temperature, humidity and CO_2 inside the Greenhouse. Based on the comparison between the measurements received from the Device Connector and the thresholds stored in the Greenhouse Catalog, it turns on the actuation system. In the first case, when receiving data, it works as an MQTT subscriber and in the second case, when sending actuation commands, it works as an MQTT publisher. It is also a REST Client getting information from the Greenhouse Catalog.

- The **Plants Control** is a control strategy that manages irrigation of each batch of pots of the Greenhouse independently. It receives the data from the Device Connector about the humidity of a batch of pots. Then the Plants Control decides whether to irrigate the specific batch of pots by comparing the current wetness of its soil with the threshold related to the water needed by each plant variety. It works as an MQTT subscriber to read the data and as an MQTT publisher to send actuation commands. It communicates with the Greenhouse Catalog to get the information about the thresholds of the plant varieties by using the REST Web Services.

- **Data Analysis** is a long term analysis that summarizes the data related to the plants growth in the last weeks/months. Moreover it reports a statistic of the future growth of the plants and the mean harvest time for each plant variety.

- **Freeboad** is an open source dashboard creation tool that simplifies tracking feeds collected by multiple devices in the IoT environment. It also helps to visualize the data by exploiting the REST and the Thingspeak Web Services to import plots about environmental measurements.

- The **Telegram Bot** is the user-side service needed to create a fast communication system between the final user and the proposed infrastructure. It can retrieve measurements from IoT devices exploiting the REST Web Services provided by the Device Connector. It uses the REST Web Services also to communicate with the Greenhouse Catalog by associating every batch of pots with the plants it has to store. Moreover it sends information to the Greenhouse Catalog about the plant varieties that have to be added or modified.