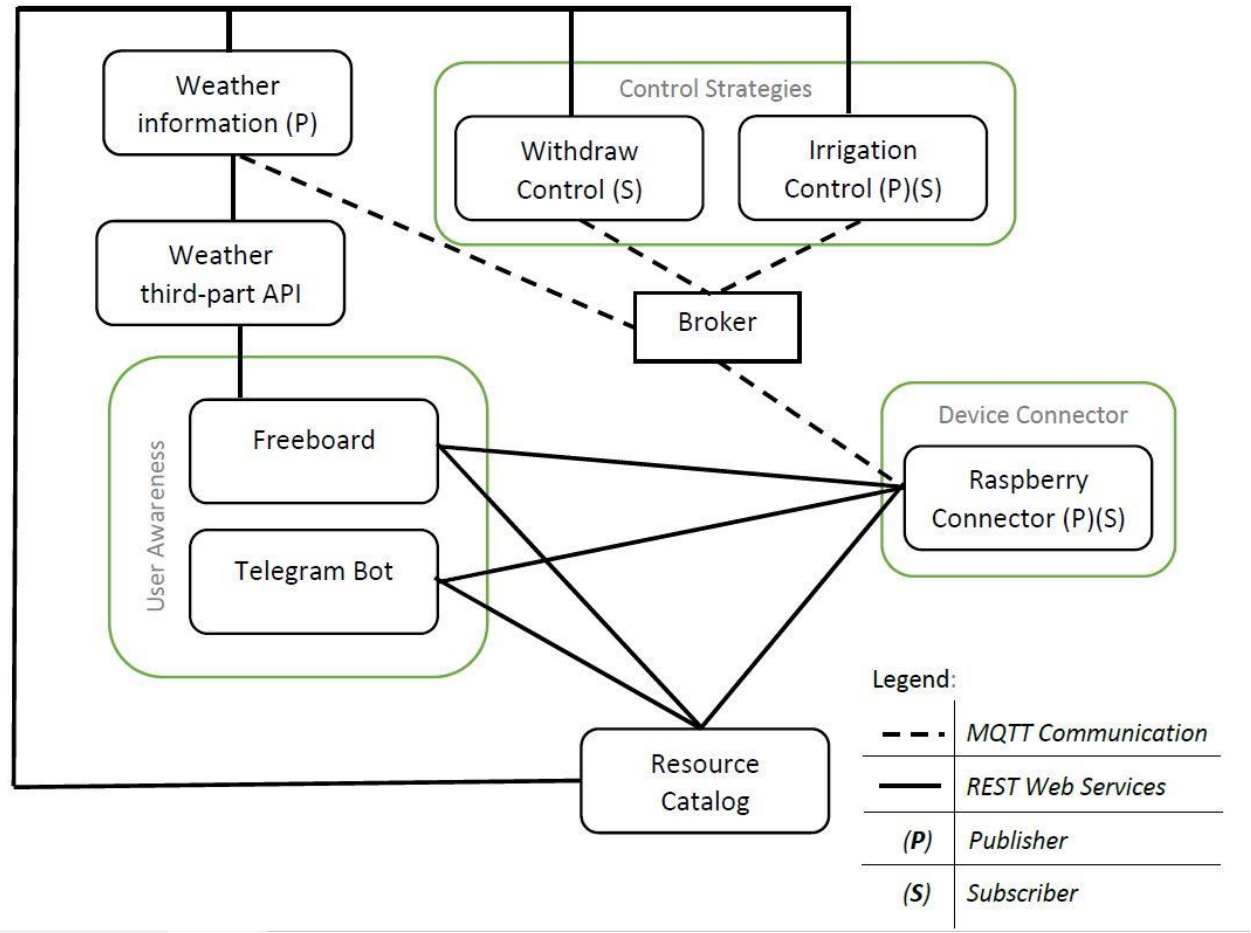
1 Name of Use Case

Name of the Use Case	**See U later Irrigator**
Version No.	v 0
Submission Date	05/12/2016
Team Members (with student ids)	Eros Filippi S194300, Enrico Suria S236170, Federica Sgrò S191570, Francesca Davi S194016

2 Scope and Objectives of Function

Scope and Objectives of Use Case	
Scope	The proposed IoT platform is willing to manage a smart device for irrigation purpose.
Objective(s)	The main idea is to minimize water waste and to reduce expenses through a smart network. It provides the best watering for any kind of plant.
Domain(s)	Smart Agriculture and Environment.
Stakeholder(s)	Farmers, Home inhabitants, City Services.
Short description	The platform helps the user to irrigate a garden in an optimal way. Every given time (for example every day), based on expected rainfall information, a pump picks up a certain amount of water from a source and collects it into a tank. The pump maintains a fixed level of water inside the tank: if no rainfalls are expected during the days, it doesn't pick up any water; otherwise it takes the amount used during the previous day. The water will remain in the tank to reach the optimal temperature, controlled by a sensor. A group of humidity sensors are set into the ground. At a certain time of the day, if the average humidity of the terrain gets below a certain value, given by default or by the user based on the type of plant, a second pump takes water from the tank and an irrigator waters the ground.

3 Diagram of Use Case



4 Complete description of the system

This IoT platform for a smart irrigation is designed following a microservices approach. It also exploits the following communication paradigms: request/response based on REST Web services and publish/subscribe based on MQTT protocol.
The main actors in this smart network are:

- **Resource Catalog** is a registry of available IoT devices and the resources they expose. Through a JSON-based RESTful API it provides:
  - Information about end-point of each actor like REST Web Services and MQTT topics;
  - Configuration settings for control strategies (list of sensors and actuators, important values) and for applications.

  Such information will be retrieved by each actor during its start-up phase.
- **Message Broker** is the heart of the publish/subscribe approach. It manages the topics and it is responsible for receiving MQTT messages, filtering and forwarding them to the right subscribers.
- **Raspberry Pi Connector** is an implementation of the *Device Connector* that integrates into the platform Raspberry Pi boards.
  - *Withdraw Raspberry* acts on the relay that controls the first pump. It also works as a MQTT subscriber.
  - *Irrigation Raspberry* is equipped with a temperature sensor for the tank water and ground humidity sensors. It also works as a MQTT client (pub/sub).

- **Withdraw Control** is a strategy that decides if it is necessary to take water from the source. From the *Resource Catalog* it takes information about the working hour and the desired water level in the tank. The amount of water to be retrieved is evaluated according to the previous consumption. The decision about taking this amount of water is based on the weather forecast information obtained from a weather service.
  Previous consumption and weather forecast are received as MQTT messages since this control strategy also acts as a MQTT subscriber.
- **Irrigation Control** strategy manages the smart watering. Its decision is taken according to the water temperature in the tank and the (average) humidity of the ground. It acts on the irrigation pump when the water temperature is over a desired level and if the (average) humidity is under a minimum value. If the water temperature is too cold, the system waits for the reaching of the threshold. The working hour is near the sunset time (to avoid plant damaging).
  It works as a MQTT client because it is a subscriber that receives information about the sunset time from the broker and it is a MQTT publisher to provide news about the water consumption.
- **Weather Adaptor** is an MQTT publisher that takes useful information about weather forecast from a third-part weather API (as a client) and forwards a formatted message to the broker with the needed data.
- **Freeboard** is a dashboard used to show data from IoT devices and visualize them exploiting the REST Web Services provided by **Raspberry Pi Connector.** It also displays weather information through a third-part weather API.
- **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 Connector**.

5 Desired Hardware

Device Name	Quantity	Needed for…
Raspberry pi	1	Irrigation Raspberry