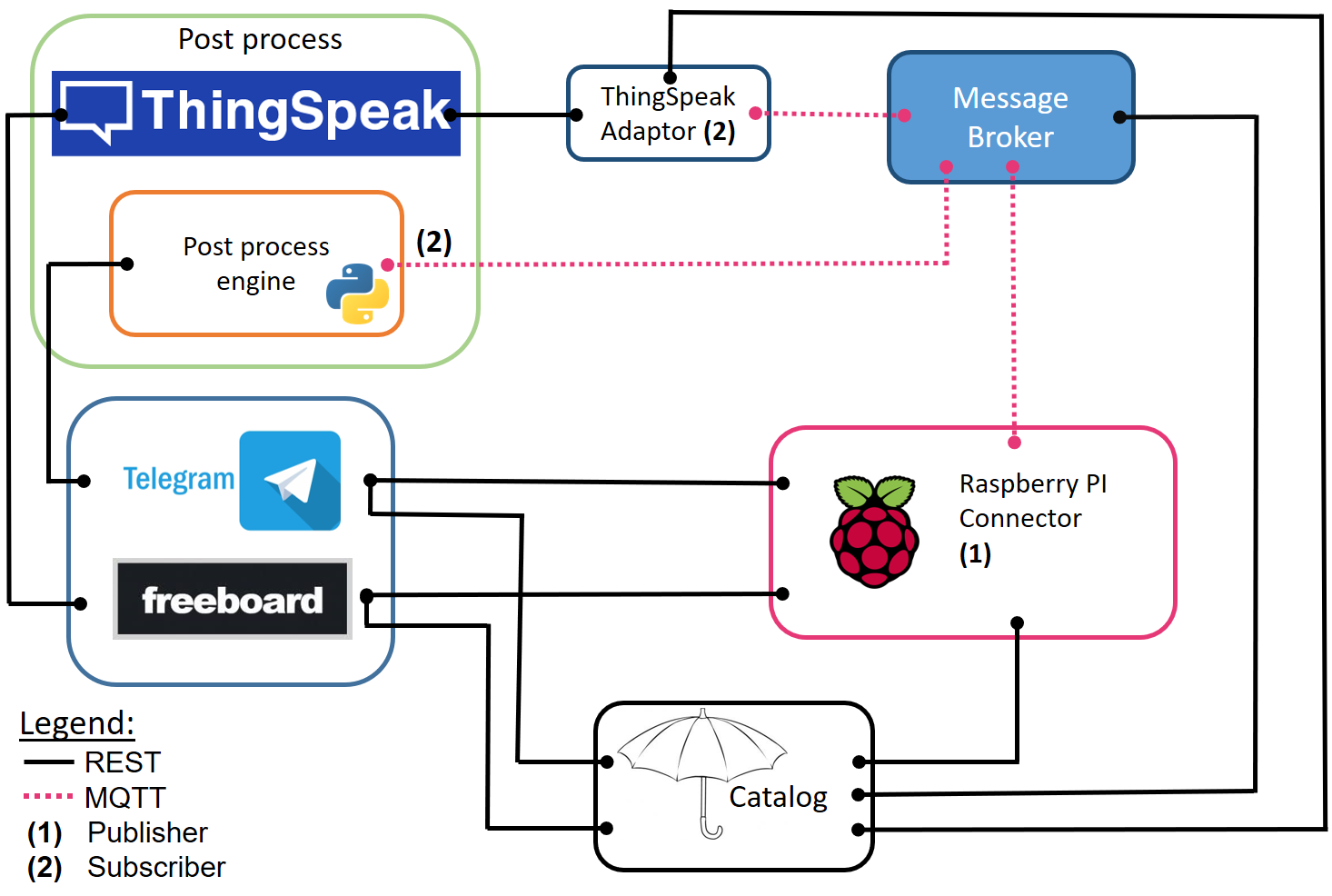
1. **Name of Use Case**

|  |  |
| --- | --- |
| **Name of the Use Case** | **IoT platform for Smart Umbrellas** |
| **Version No.** | V0.2 |
| **Submission Date** | 12/01/2018 |
| **Team Members (with student ids)** | 242541, Luca Colaci  250714, Louise Pelle  251622, Alexandre Proy  251897, Alexandru Tarna  250678, Johanna Von Hammerstein |

1. **Scope and Objectives of Function**

|  |  |
| --- | --- |
| **Scope and Objectives of Use Case** | |
| **Scope** | The proposed IoT platform aims at providing tailored weather forecast services through a network of smart umbrellas. |
| **Objective(s)** | The expected results consist in providing more accurate and fine-grained weather forecasts, namely rain warnings, to the platform users so that they never go out in the rain without their umbrella. |
| **Domain(s)** | Smart cities, wearables |
| **Stakeholder(s)** | citizens |
| **Short description** | The proposed IoT platform aims at making our umbrellas smart and helpful to others. It integrates different IoT sensors to provide useful feedbacks to the platform so that it can notify other nearby users in case it is raining. The overall platform relies both on REST and MQTT in order to keep track of all its users and to communicate with them. Finally, the Smart Umbrella IoT platform is able to provide its end-users with accurate weather forecasts at every moment of the day.  To sum up, the main features offered by our platform are:   * rain warnings to concerned users will be sent by a post pocess engine implemented in Python. The post pocess engine will receive information from all the umbrellas and based on the processd data, it will send notifications only to the impacted users of the platform. * unified interfaces (i.e. REST and MQTT services) * end-user application to have a summary of her/his umbrella usage as well as see an interactive map of the other umbrellas roaming the city. For this, we will use ThingSpeak to post-process the data and communicate the processed data to freeboard through REST. |

1. **Diagram of Use Case**

****

1. **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, we have identified 8 actors that are defined as follow:

* The Message Broker provides an asynchronous communication based on the publish/subscribe approach. It exploits the MQTT protocol.
* The Umbrella Catalog works as a catalog 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 (umbrellas), resources (umbrella open, humidity, temperature) and services in the platform (Thingspeak, Telegram). It also provides configuration settings for the application (e.g. list of sensors). Each actor, during its start-up, must retrieve such information from the Umbrella Catalog exploiting its REST Web Services.
* The Raspberry Pi Connector is an implementation of the Device Connector that integrates into the platform raspberry pi boards. Each raspberry is equipped with a push button (the same you press to open the umbrella, a temperature and humidity sensors to provide environmental information about the weather conditions around the users. It provides Rest Web Services to retrieve environmental information (i.e. temperature and humidity). It also works as a MQTT publisher sending as information when there’s a pression on the button (when you open the umbrella) and environmental data.
* 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. The inputs will be the temperature and humidity values, and the location of opened umbrellas. As an output, we will have graphs representing temperature and humidity values throughout the day, and a map with the location of the open umbrellas.
* The Python post process engine will receive as input: ID, location and the status (open/closed) for each umbrella. The script itself will be implemented in a MQTT fashion and it will act as a subscriber to receive the data. It will then send notifications to the interested users based on the processed data. Using the post process engine we can isolate the raw data that will be sent by umbrellas and the gui on wich notifications will appear (Telegram). So, the system is more flexible and new features (i.e. new interface for notifications) can be added without modifying the entire platform. This will be the actor in charge of sending rain warnings to the users.
* Freeboad is a dashboard to retrieve data from IoT devices and visualize them exploiting the REST Web Services provided by the Raspberry Pi Connectors. It also exploits the Thingspeak Web Services to import both the plots about environmental measurements (humidity and temperature) as well as the map with the umbrellas’ location.
* Telegram Bot is a service to integrate our proposed IoT infrastructure into the Telegram platform, which is a cloud-based instant messaging infrastructure. It will be used in combination to Thingspeak to send users sharing the location of a high number of open umbrellas a notification to remind them to take their umbrella.

1. **Desired Hardware**

|  |  |  |
| --- | --- | --- |
| **Device Name** | **Quantity** | **Needed for…** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |