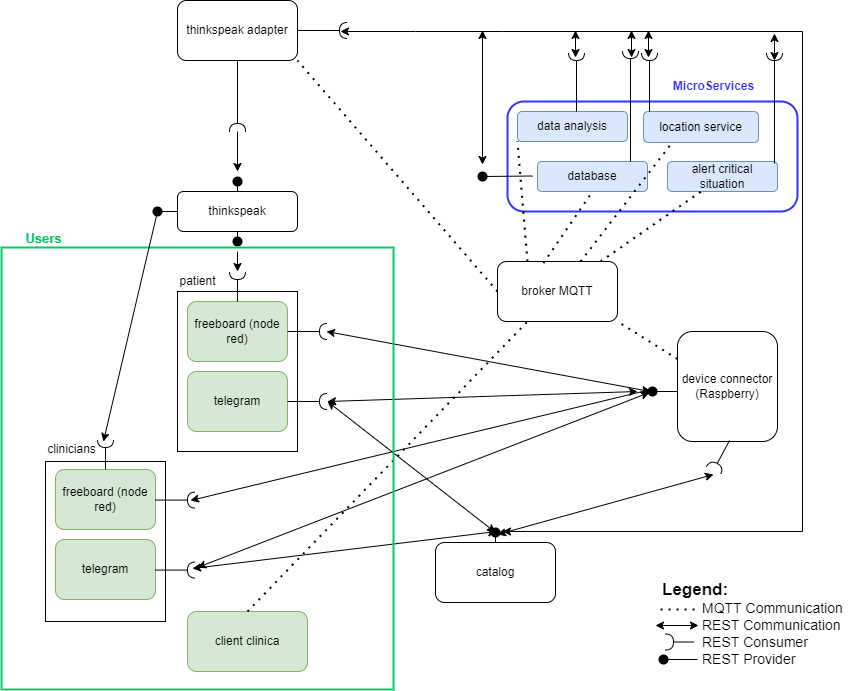
1. Name of Use Case

|  |  |
| --- | --- |
| **Name of the Use Case** | **IoT platform for continuous monitoring of patients** |
| **Version No.** | 1 |
| **Submission Date** | 10/12/2021 |
| **Team Members (with student ids)** | Gagliardi Marco (284171)  Filippi Edoardo (292207)  Alessandria Domenico Emanuele (292216)  D’Adamo Davide (289326) |

1. Scope and Objectives of Function

|  |  |
| --- | --- |
| **Scope and Objectives of Use Case** | |
| **Scope** | The aim of this platform is to guarantee a 24/7 monitoring for patients with assessed medical issues |
| **Objective(s)** | Provide a smart control to:   1. prevent critical events and/or guarantee an immediate response (such as stroke in cardiovascular patients) 2. simplify personal therapies (such as insulin self-injection for diabetic people) 3. automate signal acquisition for easier quantification of disease evolution (such as sensorized orthesis for postural disease patients) |
| **Domain(s)** | e-Health |
| **Stakeholder(s)** | Patients, clinicians and clinics |
| **Short description** | The proposed IoT distributed platform aims to make patients' life safer and easier, while making clinicians' assistance more immediate and cost-efficient. In particular, the system manages the communication between personal wearable devices and reference doctors, storing historical interesting data (depending on the disease being monitored) and providing alarm to clinicians in case of critical events detected. Moreover, the system can help patients in self-therapies (where possible and needed) through a Telegram bot. We are going to develop detailed code to manage three exemplary disease-specific services, but the platform will be structured (standardized data format and communication paradigm) in well-defined blocks, in order to easily add new monitoring system, only adding message content, graphical objects, or specific needed actions.   1. The first service explains the capability to react to incoming critical events with immediate clinician’s acknowledgement and intervention: cardiac rhythms monitoring through wearable ECG system and cardiac frequency, which detects incoming stroke and alerts personal doctor and nearest subscribed local clinic with ECG data and location of the patient. 2. The second service explains the capability to trigger self-therapy in patients: glycemic index smart device, which monitors real time and communicate with the patient through Telegram bot, ipo/iper-glycemia status, in order to induce him/her to assume appropriate medicines. 3. The third service explains the capability to store quantitative data useful for future clinic observations: sensorized orthopedic corset, which detects when the device has been weared and vice versa, and collects quantitative data of interest for next therapy phases. |

1. Diagram of Use Case



1. Complete description of the system

The proposed IoT platform for patient monitoring follows the micorservices 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, these actors have been identified:

* The **Message Broker** provides an asynchronous communication based on the publish/subscribe approach. It exploits the MQTT protocol.
* The **Raspberry Pi Connector** is a *Device Connector* that integrates into the platform raspberry pi boards. Each raspberry is connected to the sensors assigned to the patient. It provides Rest Web Services to retrieve sensor data (i.e. heart rate). It also works as an MQTT publisher sending data readings (every 5 minutes).
* **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 access privileges informations oof the different users in the system, and the location of clinics registered. Each user, during its start-up, must retrieve such information from the Home Catalog exploiting its REST Web Services.
* **Thingspeak Adaptor** is an MQTT subscriber that receives measurements from the patients sensors and uploads 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).
* **Freeboad (patient)** is a dashboard to retrieve data from IoT devices and visualize them exploiting the REST Web Services provided by **Raspberry Pi.** It only has access to the private data of the patient.
* **Telegram Bot (patient):** is a service to integrate the proposed infrastructure into Telegram platform, which is cloud-based instant messaging infrastructure. It retrieves measurements from the IoT sensors of the patient exploiting the REST Web Services provided by **Raspberry Pi.** In addition to theis service a message in sent in the case of a critical reading by the critical situation alert system.
* **Freeboad (clinician)** It has similar functionality to the patient freeboard but shows a review of the patients under the clinician care.
* **Telegram Bot(clinician):** critical messages are sent to this service by the critical situation alert microservice if needed. Furthermore, the Telegram bot with clinician privileges can read data from all the sensors of the patients under their care.
* **Client(clinic)**: It is a python runnable program on the pc of the clinic that exploits the MQTT protocol, it is subscribed to the broker and receives messages about potentially dangerous situations in patients located near the clinic.
* **Data analysis service:** is an mqtt subscriber that receives data from all the sensor and every hour saves a summary of each patient sensor readings for that period in the database using the MQTT protocol.
* **Database service:** using the MQTT protocol reads the data sent by the data analysis microservice and saves them formatted in a json file. It can be directly consulted by the user for the history of the sensor readings using REST webservices.
* **Critical situation alert:** receives data from the sensors in real time using REST and if there is an out of safe range reading alerts the correct users (assigned clinician and the patient itself). It also publishes a message to be read from the location service with the last known user position.
* **Location service:** If a critical message is sent it retrieves the location of the patient and finds the nearest clinic and aletrs it using the MQTT protocol (each clinic is a subscriber to the correct topic)