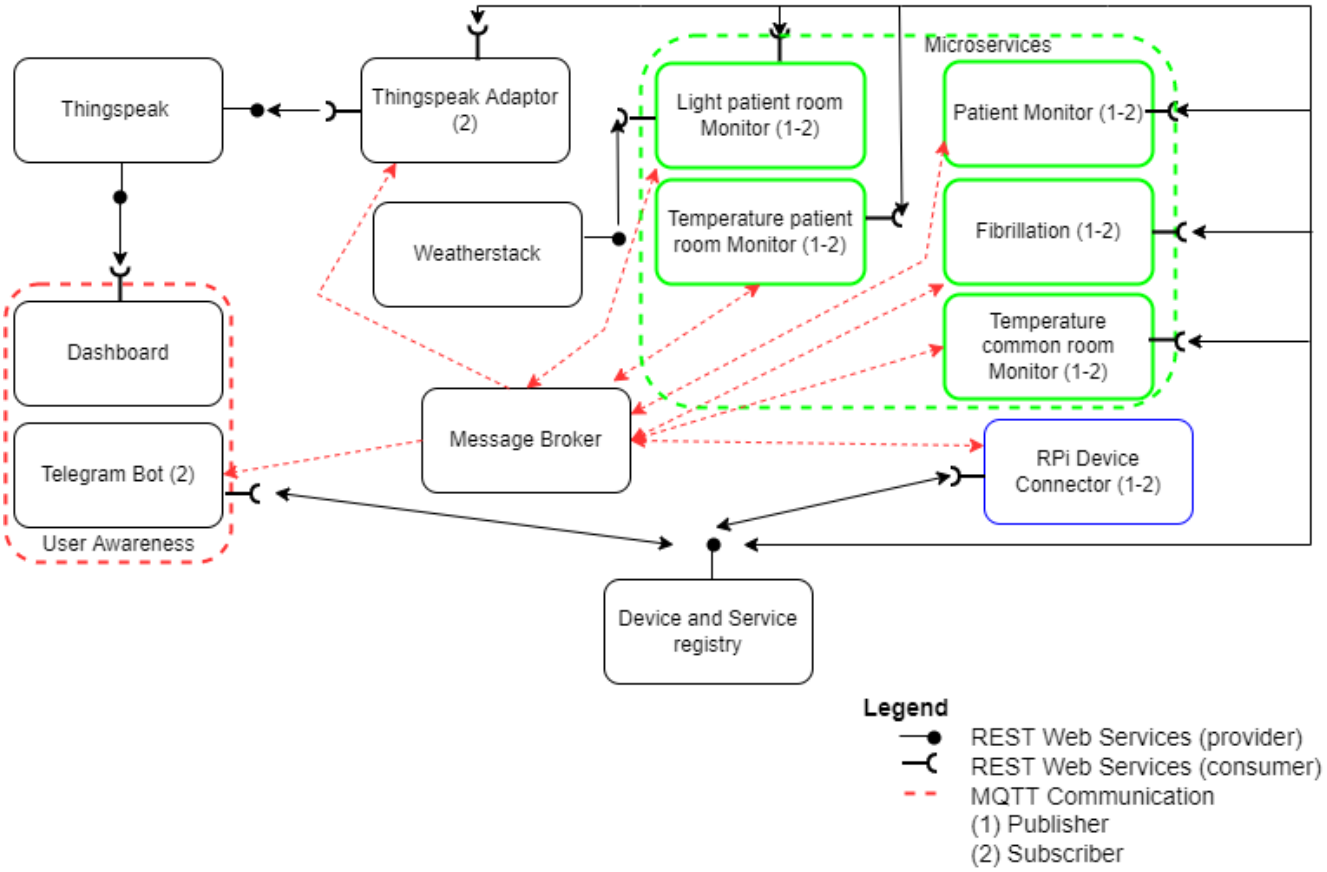
1 Name of Use Case

Name of the Use Case	**IoT Patient Care**
Version No.	V1.0
Submission Date	12/12/2021
Team Members (with student ids)	Centamore Lorenzo s289305 D'Apice Alessandro s287049 Palmucci Leonardo s288126 Sarpa Amedeo s281638

2 Scope and Objectives of Function

Scope and Objectives of Use Case	
Scope	The proposed platform aims to collect and analyze patients' data and manage the rooms of a nursing home.
Objective(s)	While medical staff assists patients, the IoT Patient Care will overseer patients' parameters and notify nurses about anomalies. It will also monitor room temperature and light and maintain them in a desired range.
Domain(s)	Smart Building, Smart Health, Smart Nursing Home
Stakeholder(s)	Patients, Doctors, Nurses, Technicians
Description	The proposed IoT platform aims to monitor and manage the rooms of the nursing home and patients' data. It is able to control the lightness and temperature of the room in a smart way. Furthermore, it can measure during the night the patient's body temperature, heartbeat rate and blood oxygen saturation; it sends real-time notifications to the nurse in case of anomalies, addressing the source of the problem. It performs analysis on heartbeat rate samples, acquired over a certain period of time, to alert the medical staff for a possible condition of fibrillation. Nurses can interact with the system through a telegram bot, setting the desired temperature of the rooms and insert or delete a patient. Nurses also receive Telegram notifications each time an alert rise. Both patient and sensor status data will be stored and organized by a third part service and ready to be shared with doctors (patient data) and technicians (sensor data).

3 Diagram of Use Case



4 Complete description of the system

The proposed IoT platform for patient and room monitoring follows the micro-services design pattern. It also exploits two communication paradigms: i) publish/subscribe based on MQTT protocol and ii) request/response based on REST Web Services.

The following actors have been identified and introduced:

- The **Message Broker** provides an asynchronous communication based on the publish/subscribe approach. It exploits the MQTT protocol.

- The **Device and Service Registry** 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. Each actor, during its start-up, must retrieve such information from the Catalogue exploiting its REST Web Services.

- The **Raspberry Pi** connector is a *Device Connector*. The system is composed by one raspberry per room. Each raspberry is able to communicate via BLE with temperature and pulse oximeter sensors to provide information about patient in the room and is equipped with motion and temperature sensors in order to observe and control these parameters. It is connected to the heating management system and to the light management system (e.g. to know if the light is on). It works as an MQTT publisher sending information about both patient and room (that will be further stored and analysed) and as an MQTT subscriber to receive actuation commands from the actors in order to pilot actuators (air conditioning and lights system).

- **Fibrillation** is a microservice used to alert nurses about a possible state of fibrillation in a patient during the night. It takes the perfusion index and pulse rate from the oximeter via a subscribe. For each sample arrived, it evaluates the perfusion index: if this is too low, it does not perform any calculation and eliminates the records acquired, otherwise, after having registered n samples, it evaluates the standard deviation and, if this is above the threshold, sends an alarm to the nurse, via a publish, on the patient's possible state of fibrillation. After analysing the n patient samples, it deletes the collected records and starts a new analysis. It works i) as an MQTT subscriber to receive patient heart beat and perfusion index ii) as an MQTT publisher to send notifications about a possible status of fibrillation.

- **Patient monitor** is a control strategy that analyses each sample arriving from the various sensors worn by the patient during the night, performing threshold checks and sending notification via MQTT in case pulsation, temperature or saturation are out of range. It also sends a notification if the sensor is removed or badly positioned (evaluated for the oximeter by perfusion index, while for the thermometer, if it returns a temperature below 35 ° C). It works i) as an MQTT subscriber to receive information on patients ii) as an MQTT publisher to send notifications about anomalies.

- **Light patient room monitor** is a control strategy that manages the lightness in the patient room. It receives messages, via a subscribe, to find out if the user has the light on or off. If the user has the light on, the microservice launches a GET Http request through the API of the web site 'weatherstack.com' to get external environmental conditions (e.g. ‘is_day’: no), and based on them it sets, via an actuation command sent by publish message, the percentage of brightness of the room. It is supposed that, when light is on, the monitor is active and able to adapt the brightness of the room (light has not only two on/off states). Instead, when light is off, it means that the user deliberately wants to keep lights off so the microservices does notingh. (e.g. duting the night). For example, if the sun has set, it sets the brightness percentage to 85%. It works i) as an MQTT subscriber to know if the light of the room is on; ii) as an MQTT publisher to send actuation commands to light management system.
- **Temperature patient room monitor** is a microservice used to manage patient room temperature in a smart way. The algorithm, basing on the current time and data get from a motion sensor, is able to recognize through a hourly scheduling too, if it is night or if, during the day, the patient is present in his room or is expected to be in. It also receive temperature data, and evaluating all the inputs, it sends actuation command to switch on or switch off the air conditioning system in order to keep the temperature in the desired value.

  - if it is “night”, checking the current temperature, the microservice sends the command to switch on or off the air conditioning system in a way to maintain the temperature as desired : considering the calendar date, if "winter-autumn" season, the microservice sends a switch on command if the temperature is lower than the set one; if "summer-spring" season, the microservice sends a switch on command if the temperature is higher than the set one .
  - if “not night”, it evaluates the temperature value and the signal from the motion sensor (e.g. let's consider it Boolean) in the room and evaluates: -- if the patient is present in the room, it sends command in a way to maintain the temperature fixed at the temperature set by the nurse (as described before,

checking the current season). -- if the patient is not in the room:
--- if during "expected presence", considering the calendar date, if in "winter-autumn" season the microservice , observing the current temperature, sends actuation command to switch on or off the air conditioning system in a way to maintain the temperature at 2 degrees lower than the one set by the nurse; if "summer-spring" season it sets the command to keep the temperature at 2 degrees higher than the temperature originally set. --- if not during "expected presence", considering the calendar date, if in "winter-autumn" season the microservice, observing the current temperature, sends actuation command to switch on or off the heating system in a way to maintain the temperature at 4 degrees lower than the one set by the nurse; if "summer-spring" season it sets the command to keep the temperature at 4 degrees higher than the temperature originally set. It works i) as an MQTT subscriber to know the current temperature and if there is a presence in the room; ii) as an MQTT publisher to send the actuation command to switch on/off the air conditioning system.

- **Temperature common room monitor** is a microservice used to manage common room temperature in a smart way. The algorithm, basing on the current time and data get from a motion sensor, is able to recognize through an hourly scheduling too, if it is night or if, during the day, someone is in the room or is expected to be in. It also receives temperature data, and evaluating all the inputs, it sends actuation command to switch on or switch off the air conditioning system in order to keep the temperature in the desired value set by the nurse.

  - if it is “night”, considering the calendar date, if in "winter-autumn" season the microservice , observing the current temperature, sends actuation command to switch on or off the heating system in a way to maintain the temperature at 4 degrees lower than the one set by the nurse; if in "summer-spring" season it sets the command to keep the temperature at 4 degrees higher than the temperature originally set.

- if “not night”, it evaluates the temperature value and the signal from the motion sensor (e.g. let's consider it Boolean) in the room and evaluates:

  -- if someone is in the room, it sends command in a way to maintain the temperature fixed at the temperature set by the nurse : considering the calendar date, if "winter-autumn" season, the microservice sends a switch on command if the temperature is lower than the set one; if "summer-spring" season, the microservice sends a switch on command if the temperature is higher than the set one.

  -- if nobody is in the room:

  --- if during "expected presence", considering the calendar date, if in "winter-autumn" season the microservice , observing the current temperature, sends actuation command to switch on or off the heating system in a way to maintain the temperature at 2 degrees lower than the one set by the nurse; if in "summer-spring" season it sets the command to keep the temperature at 2 degrees higher than the temperature originally set.

  --- if not during "expected presence", considering the calendar date, if in "winter-autumn" season the microservice, observing the current temperature, sends actuation command to switch on or off the heating system in a way to maintain the temperature at 4 degrees lower than the one set by the nurse; if in "summer-spring" season it sets the command to keep the temperature at 4 degrees higher than the temperature originally set.

  It works i) as an MQTT subscriber to know the current temperature and if there are presences in the room through motion sensors; ii) as an MQTT publisher to send the actuation command to switch on/off.

- **Weatherstack** is a web site that provides, trough API, information about external environmental conditions of a certain city. For example information like weather conditions (e.g. partly cloudy) , cloud cover (0-100%) or is_day (yes or no).

- The **Thingspeak Adaptor** is an MQTT subscriber that receives measurements on patients and sensor status and upload them on Thingspeak through REST Web Services.

- **Thingspeak** is a third-party software (https://thingspeak.com/) that provides REST Web Service. It is an open-data platform for the Internet of Things to store, post-process and visualize data.

- **Dashboard** exploits the Thingspeak Web Services to import plots about sensors status and patients and roomsdata.

- **Telegram Bot** is a service to integrate the proposed infrastructure into a Telegram platform, which is cloud-based instant messaging infrastructure. It can:
  - receive warnings in case of anomalies (via MQTT)
  - register patients (via Rest Web Service)
  - set desired rooms temperature