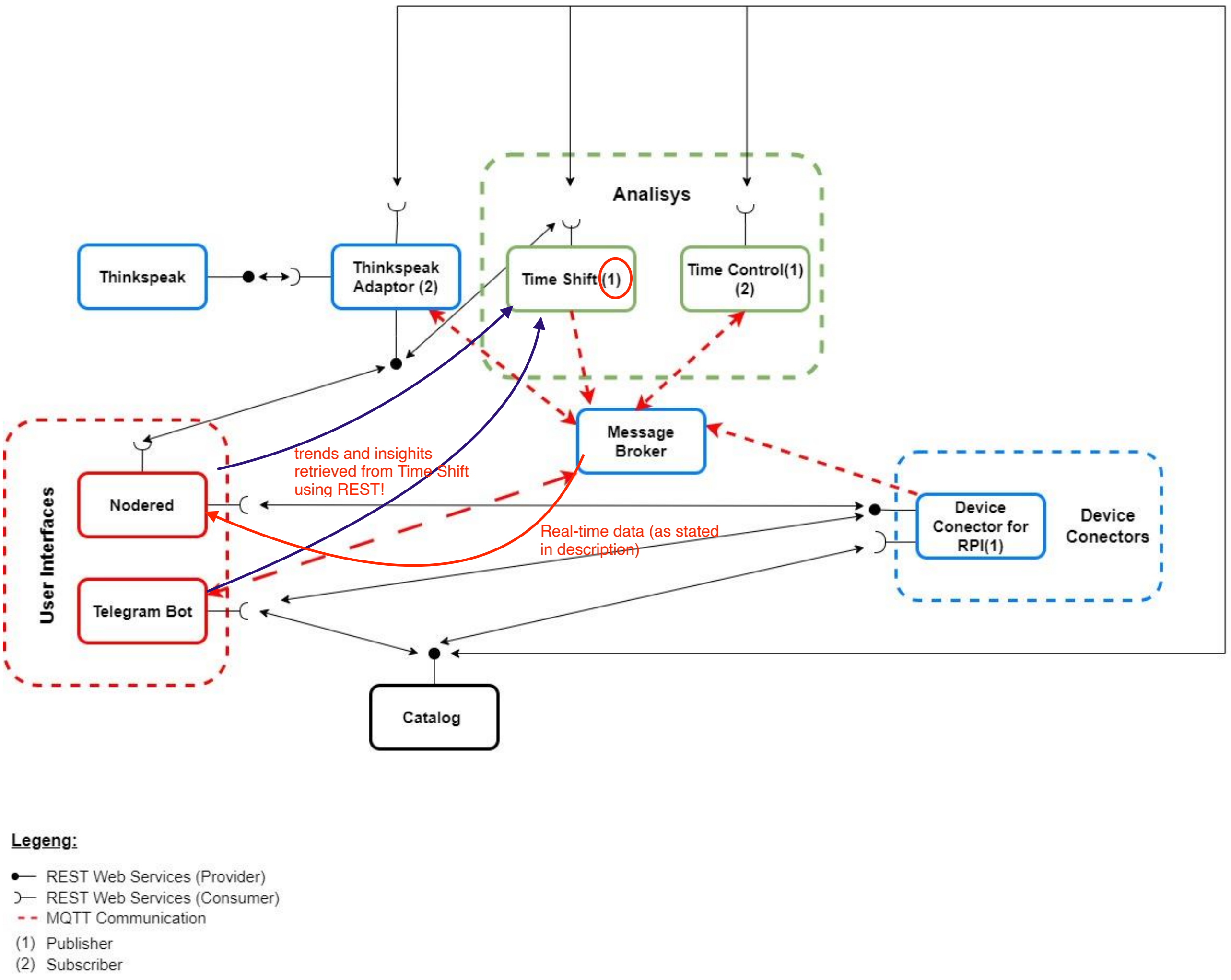
1 Name of Use Case

Name of the Use Case	**Integrated IoT Platform for Elderly Care**
Version No.	V0.2
Submission Date	06/01/2024
Team Members (with student IDs)	Thiago Baldassarri Levin 334838 Barbara Valdete Correia da Cunha 287436 Francesca Fornasier 338688 Asia Ferri 331439

2 Scope and Objectives of Function

Scope and Objectives of Use Case	
Scope	The proposed IoT aims to integrate safety features and health monitoring for elderly individuals.
Objective(s)	The system will monitor vital signs and personal safety, providing real-time alerts to caregivers as needed.
Domain(s)	Elderly care, smart healthcare
Stakeholder(s)	Private caregivers, Family Members, Nursing Homes
Short description	The IoT platform integrates wearable sensors to provide real-time safety and health monitoring for elderly individuals. It includes accelerometers for fall detection, health sensors for tracking vitals, an emergency button, and a caregiver notification system. Using MQTT and REST APIs, the platform ensures efficient communication between devices. Key features include: • Fall detection and alerts. • Vital signs monitoring. • Scalable integration through a Catalog. • Multi-channel notifications via apps.

3 Diagram of Use Case



trends and insights retrieved from Time Shift using REST!

Real-time data (as stated in description)

4 Complete description of the system

- **Device Connectors**: These interface non-IoT devices, including sensors and actuators and the platform. The Device Connector for Raspberry Pi integrates sensors to collect health data. These connectors use REST APIs to retrieve data and MQTT to transmit sensor readings to the Message Broker, ensuring real-time, low-latency communication.

What do you intend with use REST APIs to retrieve data? In the outputs you state is MQTT (which is fine).

  - **Inputs**: Data from sensors (wearable accelerometers for fall detection, thermometers for body temperature, heart rate sensors and oximeters).
  - **Outputs**: Sensor data published to the Message Broker via MQTT.

- **Message Broker**: This is the platform's central communication hub, facilitating asynchronous data exchange between device connectors, control strategies, cloud platforms, and user interfaces. It ensures seamless communication by publishing sensor's data from wearables to all relevant components, enabling quick analysis and action.

  - **Inputs**: Sensor data from Device Connectors via MQTT.
  - **Outputs**: Sensor data distributed to subscribing services like Time Control, ThinkSpeak Adaptor, and Node-RED.

- **Time Control**: Monitors short-term patterns in real-time data to detect anomalies or events that require immediate attention. It uses techniques like moving average smoothing to filter noise and Z-score analysis for anomaly detection. For instance, if a slightly elevated heart rate is detected but remains within a safe range, Time Control may delay notifying caregivers until further observations confirm a trend or anomaly. The module prioritizes actionable alerts using thresholds that adapt dynamically based on recent data trends, minimizing false alarms and reducing notification fatigue for caregivers.

  For recent data trends, it might be useful to retrieve historical data from the Thingspeak Adaptor

  - **Inputs**: Real-time sensor data from the Message Broker via MQTT. Maybe historical data from Thingspeak Adaptor?
  - **Outputs**: Alerts or insights sent to the Telegram Bot via REST API for caregiver notifications.

    I suggest you to employ MQTT for notifications rather than REST...

- **Time Shift**: Focuses on long-term historical data analysis to identify recurring patterns or anomalies. It uses simple statistical techniques such as moving averages and rolling window analysis to detect trends in activity or health data. Additionally, basic clustering (k-means) groups historical behavior into normal and abnormal patterns. For example, it might detect repeated falls at specific times (e.g., early mornings) or irregular activity during nighttime. By analyzing historical data in time buckets and aggregating deviations, Time Shift provides caregivers with actionable insights, such as suggesting targeted interventions during high-risk periods.

  - **Inputs**: Historical data retrieved from ThinkSpeak via REST API.
  - **Outputs**: Analyzed trends and insights to be sent to the Telegram Bot and Node-RED via REST API.

- **ThinkSpeak Adaptor**: Bridges the platform to the ThinkSpeak cloud for data storage and visualization. It receives environmental and health data via MQTT and uploads it to ThinkSpeak using REST APIs, enabling historical data analysis and integration with third-party tools.

  Exposes REST APIs to enable other services (e.g. Time Shift) to retrieve historical data!!

  - **Inputs**: Real-time sensor data from the Message Broker via MQTT.
  - **Outputs**: Sensor data uploaded to ThinkSpeak via REST API.

- **ThinkSpeak**: (https://thingspeak.com/) is a third-party software that provides REST Web Services for storing, post-processing, and visualizing IoT data. It serves as an open-data platform for the Internet of Things, allowing the system to leverage its historical data storage and analysis capabilities. Through ThinkSpeak, real-time sensor data is uploaded via REST APIs, enabling caregivers and the system to generate actionable insights. For example, caregivers can visualize trends through intuitive plots and access long-term patterns for better decision-making. The integration of ThinkSpeak enhances the platform's ability to analyze historical data effectively, supporting features like Time Shift and data visualization in Node-RED.

  - **Inputs**: Sensor data uploaded from ThinkSpeak Adaptor via REST API.
  - **Outputs**: Historical data provided to Time Shift and Node-RED via REST API for analysis and visualization.

    This data is retrieved from the Thingspeak Adaptor and not directly from Thingspeak!

- **Node-RED**: Acts as a real-time and historical data visualization dashboard, providing intuitive graphs for health readings and alerts. Retrieve measurements from Device Connector using REST ?? Or why is the REST connection between Node-RED and the Device Connector?

  - **Inputs**: Real-time data from the Message Broker via MQTT; historical data and insights from ThinkSpeak and Time Shift via REST API. Thingspeak Adaptor
  - **Outputs**: Graphical visualizations for caregivers, accessible via a web-based dashboard.

- **Telegram Bot**: This multi-functional tool allows caregivers to manage various platform aspects efficiently. Through the bot, caregivers can schedule medications by providing details such as the medication name, dosage, and timing. At the scheduled times, they will receive reminders to give medications. Additionally, the Telegram Bot delivers real-time notifications about health alerts, ensuring caregivers stay informed about critical issues as they arise.

  - **Inputs**: Alerts from Time Control and Time Shift via REST API; caregiver inputs for medication scheduling.

I suggest you to employ MQTT for alerts!!

For insights and analysed trends from time shift is fine to use REST… but for alerts, i suggest you to employ MQTT!!

Retrieve measurements from Device Connector using REST ?? Or why is the REST connection between Telegram Bot and the Device Connector?

  - **Outputs**: Notifications and reminders sent to caregivers.

- **Catalog**: A centralized registry for all devices, services, and configurations. It dynamically provides MQTT topics, REST endpoints, and configuration settings for all components. This ensures that new devices or services can be added without requiring changes to the platform's source code, allowing the system to be scalable.
  - **Inputs**: Configuration requests from microservices.
  - **Outputs**: MQTT topics, REST endpoints, and device configuration details provided to requesting components.

The system's data flow begins with sensors collecting real-time information on motion, temperature, and health metrics. The Device Connector processes this data and publishes it to Message Broker via MQTT. Time Control processes real-time data to detect immediate anomalies, while Time Shift retrieves historical data from ThinkSpeak via REST API to analyze long-term patterns. Data is simultaneously uploaded to ThinkSpeak for historical storage and visualization, which Node-RED accesses for real-time and historical monitoring. Alerts and critical information are transmitted to caregivers through the Telegram Bot, ensuring timely responses and effective management of elderly safety and health.

5 Desired Hardware components (only among those we can provide)

Device Name	Quantity	Needed for…