RUGBY-360

MQTT Hierarchy

Cristian De Nicola, MAT: **744954**; Rosario Catalfamo, MAT: **744951**

1. Introduction

This document outlines the MQTT topic hierarchy used in the Python project for collecting and publishing real-time sensor data related to rugby players. The structure is designed to facilitate communication between the Python simulation script, the MongoDB database, and the MQTT broker.

2. MQTT Topic Hierarchy

The MQTT topic hierarchy is organized to manage and distribute player metrics effectively. Each player has their own set of MQTT topics for publishing detailed real-time sensor data.

Topics used in Sim.py

- **rugby/players/**{player_id}**/sensors**: Publishes the complete set of sensor data for each player.
- **rugby/players/***{player_id}***/sensors/coordinates:** Publishes simplified GPS coordinates for each player.

Topics used in Realtime_metrics.py

- **rugby/players/***{player_id}***/sensors/metrics:** Publishes summary metrics like average velocity, distance traveled, calories consumed, and max heart rate, which are calculated from the data stored in MongoDB.

3. Detailed description of Topics

1. rugby/players/*{player_id}***/sensors**

- **Purpose:** This topic publishes all sensor data related to a specific rugby player, including physiological, positional, and performance metrics. Each player has a unique player_id that is replaced with the player's specific ID.

- **Usage:** During the simulation, for every second of gameplay, a full set of sensor data for each player is published to this topic. This data can be used for real-time analysis or stored in a database for future processing.

2. rugby/players/{player_id}**/sensors/coordinates**

- **Purpose:** This topic is used to publish the simplified GPS coordinates (x, y) of a specific player. It focuses solely on the player's position on the rugby field.

- **Usage:** This topic is updated alongside the main sensors topic, allowing for real-time tracking of the player's position on the field without the need for all other sensor data.

3. rugby/players/{player_id}**/realtime/metrics**

- **Purpose:** This topic publishes summary metrics for a specific player, such as average velocity, distance traveled, calories consumed, and heart rate. Each player has a unique player_id that is replaced with the player's specific ID.

- **Usage:** This topic is used to publish periodically summarized data calculated from stored MongoDB data. This provides an overview of the player's performance and condition over time.

4. Field description

- **timestamp**: The current timestamp in ISO format.

- **elapsed_time**: The total time elapsed in the simulation (seconds).

- **player_id**: Unique identifier for the player.

- **role**: The role of the player on the rugby field (e.g., wing, prop).

- **heart_rate**: Current heart rate of the player (bpm).

- **temperature**: Current body temperature of the player (ºC).

- **blood_pressure**: Current blood pressure with systolic and diastolic values (mmHg).

- **calories_consumed**: Total calories burned by the player during the simulation.

- **gps**: Positional data including coordinates (x, y), unique ID (unic), current speed (velocity), and top speed.

- **impacts**: Number of impacts the player has experienced, along with the average impact force.

- **impact_to_play_ratio**: The ratio of impacts to playing time.

- **velocity_variability**: The variation in the player's speed compared to the mean speed.

- **max_heart_rate**: The maximum heart rate the player has reached.

- **impact_severity_index**: A severity index derived from the number and intensity of impacts.

5. Summary

The MQTT topic hierarchy designed for the rugby Python project is simple yet effective, allowing for the publication of detailed metrics for each player on dedicated topics. This structure facilitates real-time data access and analysis, supporting accurate monitoring and timely interventions based on player performance and physical condition.

This organization of topics can be further expanded to include additional metrics or divide data into more specific categories as the project evolves.