Group 7 – Team 3

TEAM REQUIREMENT DOCUMENT

Sensor Management

Controller Management

Sensor Data Binder

- Team-3 Members:
  - Ashish Chauhan (2021201044)
  - Utkarsh Tripathi (2021202007)
  - Aditya Vishnu Tiwari (2021202029)

1. OVERVIEW

A. Introduction

- This module is responsible for handling the sensors, starting from registering it for the very first time to fetching and preprocessing data and sending it to application.
- Module also deals with the registration and configuration of the controllers.

B. Intended Use & Scope

- We begin with the registration of all the sensors, providing unique ID to each, that could be used for data input to any of the AI Model deployed onto the platform.

- The sensor along with its id is then available on the platform to be made use of. The user gets to choose the sensor for data collection in his model.

- Sensors are configured accordingly post registration, and then the data is being fetched from sensors using Kafka. Also, data collected from the application is being sent to the controller specified by the user.

- To make things smooth for the application developer, this module hides the complexities of sensor and controller technologies providing a single format using abstraction.

- This module intends to provide scalability to our project wherein we can add the sensors and/or controllers to the list of our existing ones, as and when required.

C. Assumptions & Dependencies

- Before the setup of the platform, all the sensors are available to be fetched in case of their registration.
- Whenever new application executes, the sensors required by it are up and running in normal fashion.
- Configuration files sent by the user for any sensor must be in the fixed format as required.

2. FEATURES AND REQUIREMENTS

A. Functional Requirements

1. Registration:
    a. *Sensor Registration* – On initialization of the platform, the sensors required by the application need to be registered. Also, in the case of new addition of sensors we need to register the newly added onto our database.

    b. *Controller Registration* – All the controllers that ought to collect data from the deployed application, must be registered beforehand in the same fashion as the sensors.

2. Configuration:
    a. *Sensor Configuration* – All the sensors registered need to be configured in a fashion that they could take the input and provide the data in some specific format for all the application.

    b. *Controller Configuration* – In similar way, we need to configure the controller types and all their instances.

3. *Sensor Identification*:
    - Upon registration of the sensor onto the platform, all its details are saved in the repository. We provide all the sensors with some unique id based on its properties, which are different for each.

    - Application Manager then uses this unique id to identify the sensor based on the parameters/ properties provided by the user, after sensor manager receives data binding request.

4. *Data Binding*:
    - After the required sensor has been identified, data is to be sent in the specified – fixed format.
    - Certain node will be servicing some instance of the application which made the request to the sensor manager with a sensor id.
    - Using the ids, the required sensor(s) for receiving/sending data are made available. The data is then sent in the specified format.

B. Non-functional Requirements

1. *Scalability*: Module deals with addition (registration and configuration) of any number of sensors and controllers to the platform.

2. *Monitoring*: The Health Check Service of the Monitoring System keeps a check on Sensor and Controller Manager. This is being done by using a heart-beat signal, which keeps ensuring in certain period time that sensors and up and functioning properly.

3. *Accessibility*: Sensor or list of sensors that any application or its instance provide in the configuration file, will collect data for it. The data collected is made accessible to that particular node for further use.

4. *Abstraction*: Data collected from various sensors might be of different type and could baffle the Application manager if it's sent in the same manner. It is made sure that all data is made available in single format.

C. External Interface Requirements

- Some external sensors and/or controllers might require manual setup or configuration before they could be actually made available on the platform.

3. INTERACTIONS ACROSS OTHER MODULES

A. Application Manager

- Application manager provides with the unique IDs of the sensor(s) which it requires to interact with for data collection.
- Sensor manager provides with the information for the specified sensor(s) which is then stored at application's end.

B. Node Manager & Nodes

- Node manager assigns a node for each application instance. Node might interact with sensor manager to send and/or receive the data.

C. Monitoring System

- The Health Check Service of the Monitoring System keeps a check on Sensor and Controller Manager, which is being done by using a heart-beat signal.

D. Fault Tolerance

- Fault tolerance module interact with Sensor manager module and in case of any discrepancy or fault, the instance of fault tolerance module related to Sensor or Controller manager gets triggered.

4. BLOCK DIAGRAMS

A. Sensor & Controller Manager :