IAS - Group - 5

Team - 4

Requirement Document

Sensor Manager

Madhusree Bera (2022202007)

Prateek Pandey (2022201035)

Amitabh Paliwal (2022201030)

May 2023

Sensor Manager

Overview

This module is designed to retrieve sensor data from OneM2M API and subsequently transmit the data to applications upon request through Kafka. Additionally, the module facilitates sensor registration for the platform by binding sensors to application services. This functionality enables the transmission of data to the application requests.

Functional Overview

1) **Functional Requirements**
    a) Sensor Registration
        i) When a sensor is introduced to the platform for the first time, or if the platform is reset, the sensor must be registered in order to access its data i.e either creating a schema for sensor or fetching the existing schema stored in sensor registry
        ii) Registration involves storing information about the sensor, such as its ID, input/output type, data rate, and location, in the sensor repository.
    b) Interaction with IoT sensors
        i) APIs are designed to offer abstraction to developers, allowing them to interact with sensors via Kafka.
    c) Identification of sensors for data binding
        i) Upon sensor registration, the Database stores all significant sensor details, including their unique IDs for easy identification.
        ii) Whenever a data binding request is received by the sensor manager, it identifies the sensor based on the given parameters and sends the data in the specified format for binding.
    d) Storing the Real-Time Sensor data
        i) Once sensors are bound to a system, a schema is generated for the corresponding sensor instance. Real-time streaming data produced by the sensors is stored in the schema specific to that instance, which is optimized for analytics. Additionally, the incoming data is duplicated and forwarded to Kafka streaming services for further processing and analysis.
2) **Non Functional Requirements**
    a) Scalability
        i) The platform can accommodate registration of multiple sensors.
        ii) At any given moment, the platform is capable of handling data from multiple sensors concurrently.

b) Flexibility with Sensors
    i) The platform is designed to be flexible in working with updated sensor types and instances. When the sensor manager is initialized, it will query Sensor Api Gateway which itself queries OM2M to determine all available sensor types and instances. If any changes are made to the sensors in OM2M, the old sensor data will be removed, and the new instance and type data will be utilized instead.

c) Accessibility of Data
    i) At any given time, sensor data is available to all services that request it through different channels. This allows multiple services and applications to access the data independently through their own channels.

d) Monitoring
    i) Sensor manager is monitored by the heart-beat module by continuous message signals sent by these components.

e) Logging
    i) To provide a comprehensive report of the current status of the sensor manager, logging is performed at two levels: module level and platform level.

Block Diagram

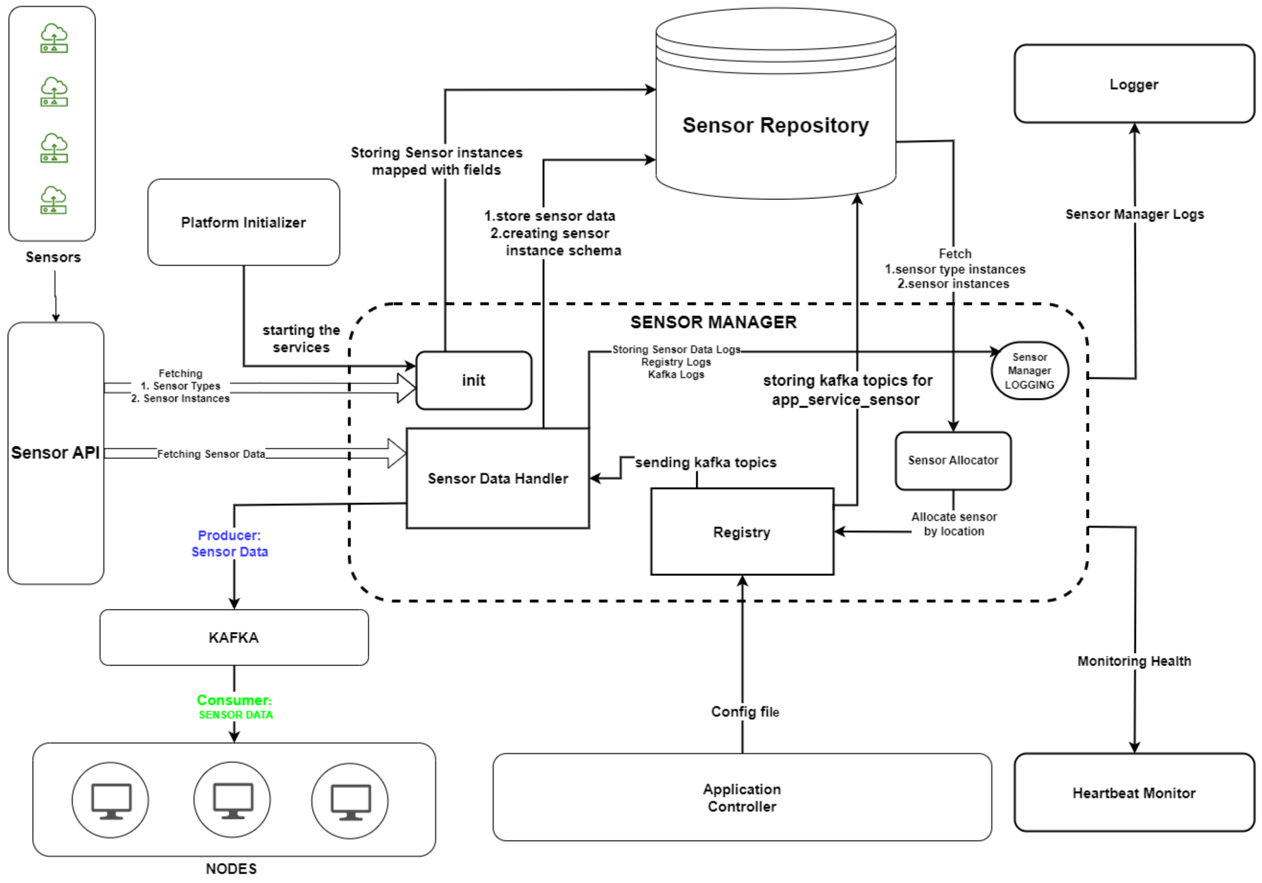

Fig 1: Block Diagram of Sensor Manager

Components of the Subsystem

- **Init**
  - Enquiring available sensors and instances at initialization
  - Storing mapped sensor instances with fields
- **Sensor Data Handler**
  - Each sensor will be uniquely identified by a sensor ID.
  - Forwards data from sensors to their respective Kafka partitions.
  - Creates sensor instance schema.
  - Generating event logs.
- **Sensor Registry**
  - Creates a kafka topic for sensor instances bound to service and application ID
- **Sensor Allocator**
  - Allocating sensor to Application filtered by sensor type location
- **Logging**
  - Logging sensor events

- **Sensor data streaming**
  - Real-time data from sensors stored in MongoDB and produced in Kafka.

Artefact

- **Sensor Configuration file**
  - A file with metadata of sensors required for the services of each Application

Services

1) **Sensor Registration**
   a) Accepts sensor config file from Application Controller.
   b) For each service of the application, the required number of sensors of the specific type is mapped with the service and a Kafka channel is created.
   c) The list of created Kafka topics is sent to the Application Controller.

2) **Providing Live Sensor data**
   a) Live sensor data fetched by OneM2M API is produced in the allocated Kafka topic for each registered service.
   b) Sensor Data is stored for each sensor instance in the Sensor Repository.

Interactions with other modules

- Sensor API Gateway sends sensor Instances at the time of initalization
- Application controller sends config file for sensors.
- Sensor type instances stored in sensor repository.
- Streams sensor data to other modules using Kafka.
- Initializer bootstraps the module and gets the sensor manager up and running.
- Logger logs the events of the module.
- Sending continuous Heartbeat status to Monitoring

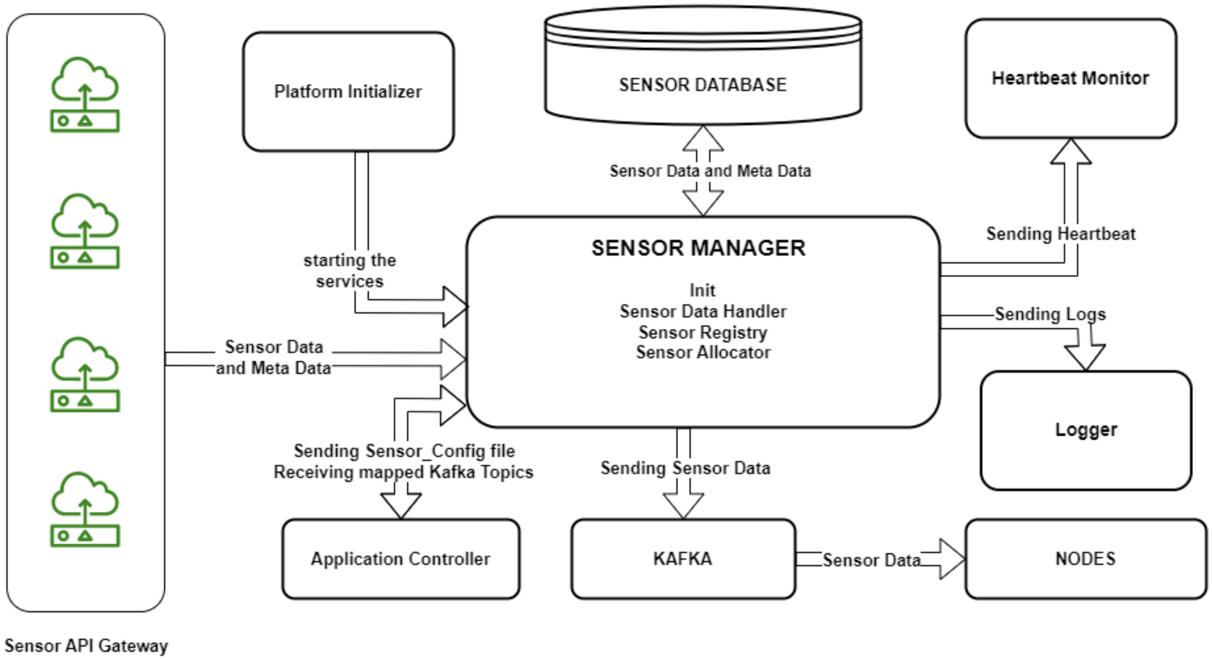

Fig 2: Interaction of Sensor Manager with other Subsystems