

Communication Module , Sensor Manager And Monitoring

Internals Of Application Server

Team Report

Group 6 Team 1

Submitted By -

- Pratik Nashine (2020201021)
- Mudit Kumar Bhansali (2020201067)
- Ayushi Maheshwari (2020201053)

Index

1. Introduction.....	2
1.1 Intended use	
1.2 Assumption and dependencies	
2. Sensor Manager.....	3
2.1 Flow Diagram	
2.2 Purpose	
2.3 Subsystems	
3. Monitoring.....	5
3.1 Flow Diagram	
3.2 Subsystem, Endpoint and Capability	
4. Communication manager.....	6
4.1 Flow Diagram	
4.2 Subsystems	
4.3 Services Provided	
4.4 Interactions	

Introduction

Communication module: This will provide communication between all the modules present on the platform

Sensor Manager : Collects data from sensors present at different locations and binds it with respective kafka topics and then binds data to algorithms.

Sensor Registration : This module helps in registration of sensors, and creating instances of sensors as per the need of the applications.

Monitoring: Monitors all the modules present in our platform and checks if they are working properly , detects problems or abnormal behaviour and informs about it to the fault tolerance manager.

Intended use -

- Connect external sensor inputs with the working algorithms associated to it and then send the result of the algorithm for further processing.
- Monitoring keeps track of the status of every component included in our platform.

Assumption and dependencies:

- Algorithms corresponding to sensors should be present.
- Configuration of the system must be set up according to the config file.

Sensor Manager :

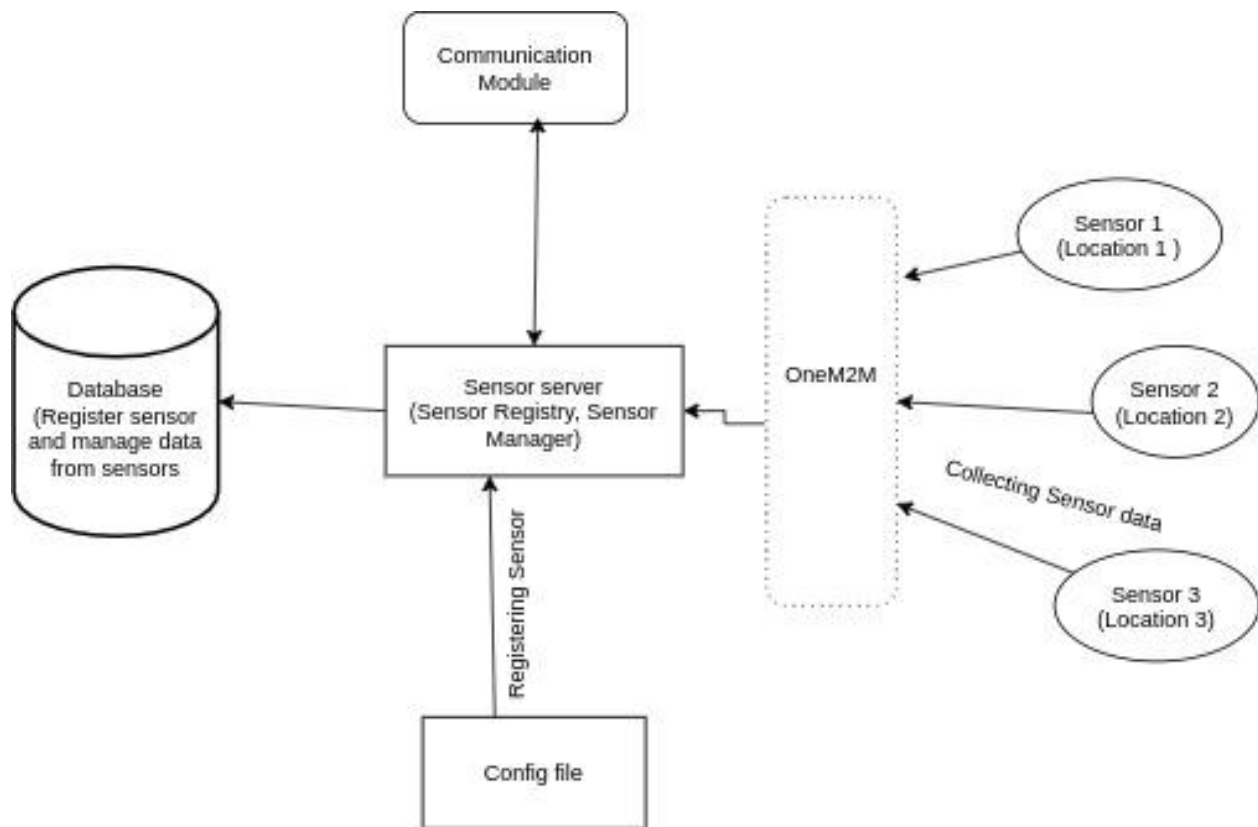


Fig.1 Flow diagram of sensor manager

The **purpose** of the sensor module is -

- to manage interaction with the sensors present at different geographic locations and collecting different data from the environment
- collecting data from the sensors
- processing sensor data so that it can be used by different applications implemented on top of our platform

- storing data from sensors in a database.
- Giving information about sensors being active or not.
- Binding data collected from server to the algorithms used by applications as per their access permissions.

The **subsystems** present are -

- **Sensor Registration** - takes config file as input from user containing details about sensor and then makes it's entry in the database. And then connect to that sensor to collect its data.
- **Sensor Server** - Sensor server has all the information about sensors present and is connected with interface to receive sensor data and database storing sensor data.
- **Data Binding** - It binds collected data to the algorithms which are implemented on top of the platform.
- **Config File** - It contains information about all the sensors present on the platform, along with their input and output data types, their geographic location , input output range, data rate.

Endpoints - Sensor module is connected with communication module which takes and receives sensor data from sensor module.

Monitoring:

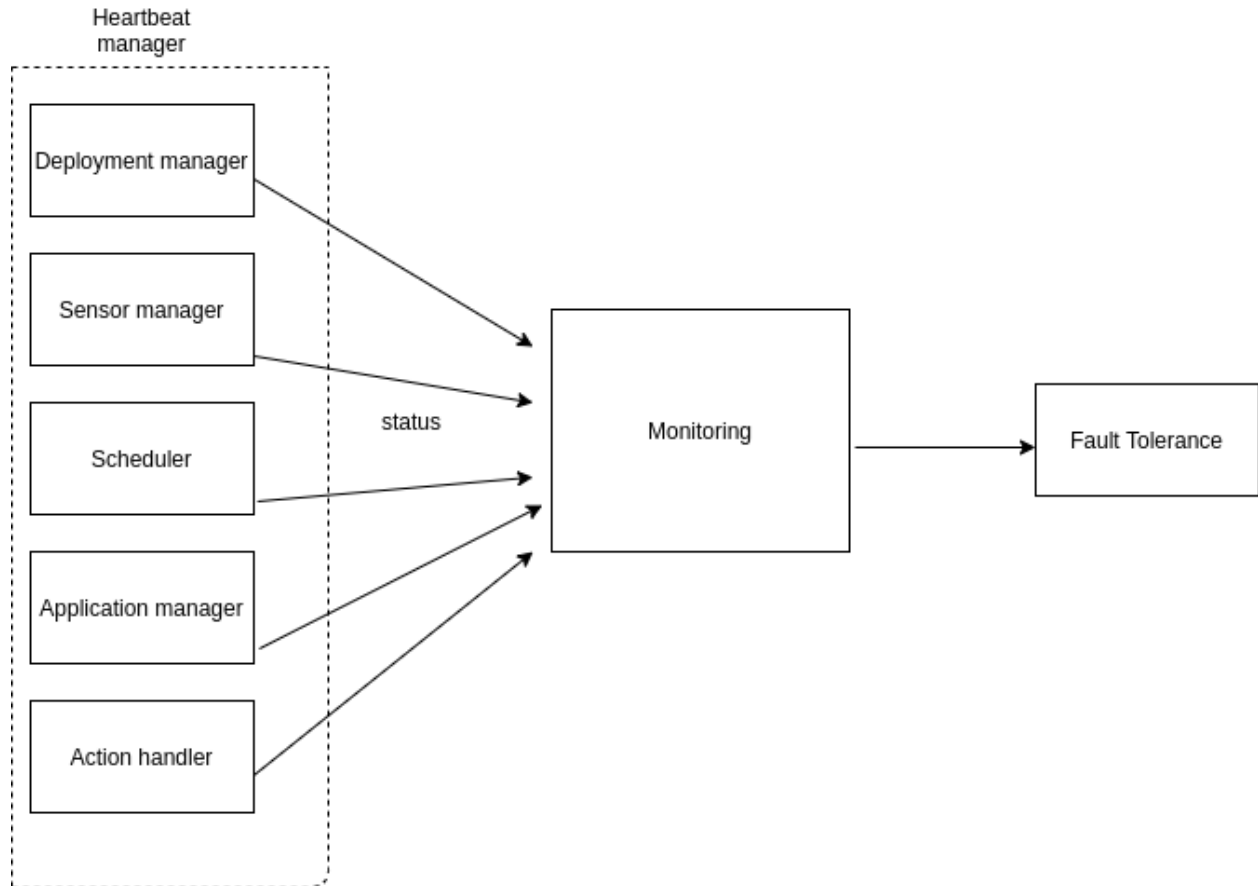


Fig 2. Flow diagram of monitoring

- **Heartbeat manager** is the **subsystem** in this part which will keep track of health i.e proper functionality of different components in our platform like of Deployment manager, sensor manager,scheduler,application manager and action handler.
- Now Monitoring keeps status of all the heartbeat from different components and sends the status to fault tolerance and now if there's any problem associated with any components the fault tolerance will take care of that.
- **End points:** Monitoring takes input as status from different components from heartbeat manager and sends status to fault tolerance if there's any issue.

- **Capability:** Keeps track of the status of proper functioning of all the different components , so that if there is some issue we can resolve and our platform works smoothly.

Communication Manager:

The efficiency and speed of any distributed system depends greatly upon its communication manager which handles how different components of the system will communicate with each other and transfer the requests and responses among them based on some communication standard.

Block Diagram of Communication Manager

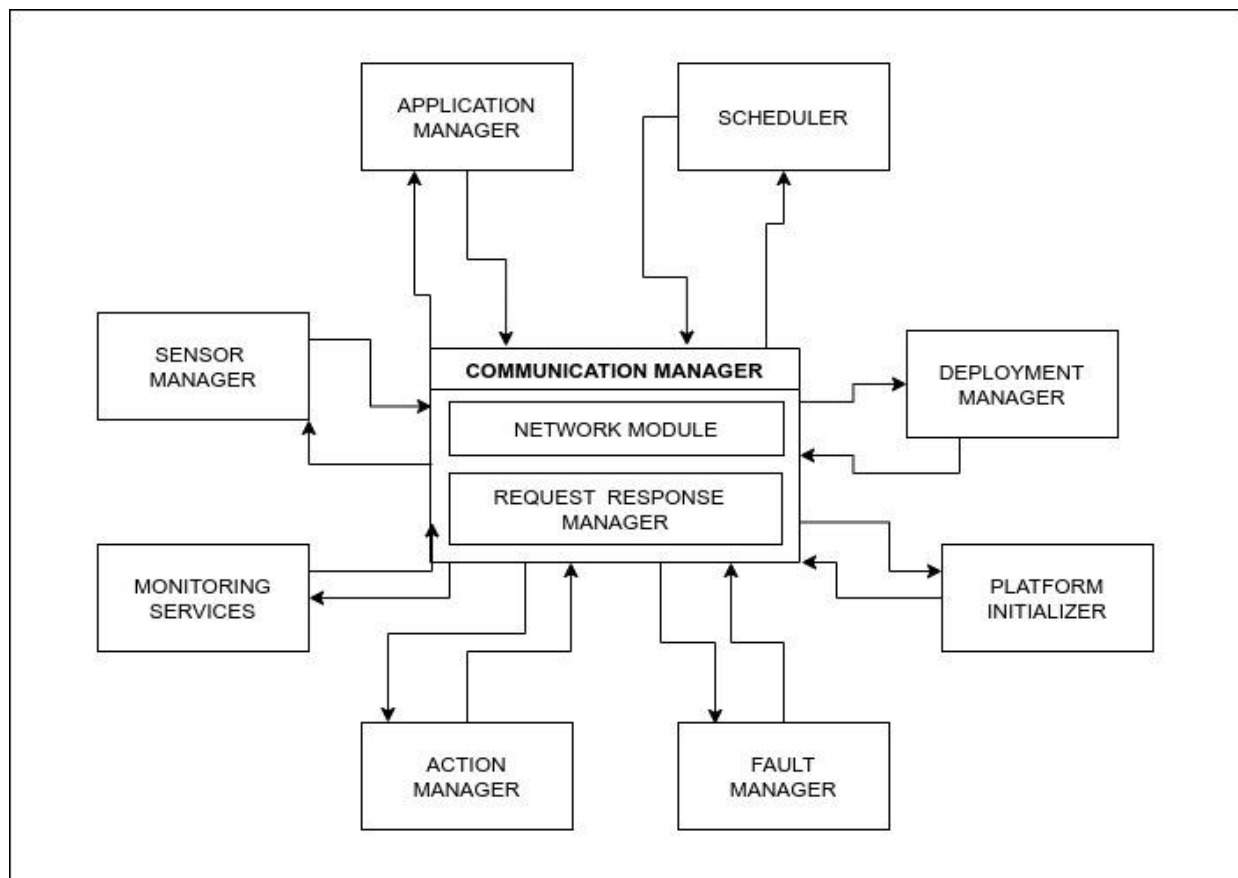


Fig.3 Communication Manager..

Sub Systems of this Module are:

- **Network Module :** The network module will basically implement our communication standard using Apache Kafka which will help in the setup of the communication channels between all the modules.

- **Request Response Manager** : This module will deal as a traffic router and will help with the routing of requests and responses to their respective senders and receivers.

Services Provided by this Module are:

- Handling the communication channel and communication framework of the whole system.
- Handling the transfer of request and response among various modules.
- Keeping communications online.
- Secure and virtually lag less transfer of data among the various components.

Interactions with other Modules are:

All the modules will be integrated with this module to be able to communicate among each others natures of such interactions are as follows:

- **Action manager** accepts actions to be performed and sends the responses required.
- **Sensor manager** accepts requests for data binding and sends the responses to the application managers and other modules.
- **Fault manager** requests the stat report from Monitoring services and send the response based on the report to the deployment manager and other modules
- **Monitoring services** pings all the modules for their stats and stores it in the log database and responds to requests from fault manager.
- **Deployment Manager** accepts requests for applications to be deployed from scheduler and fault manager.
- **Application manager** sends the request for required data to the sensor manager and application details to deployment manager and scheduler.
- **Scheduler** sends the request to the application manager to fetch the application and sends the request to the deployment manager to deploy an application.
- **Platform Initializer** initializes all the components and sends their respective configurations to them.