

IAS - Group - 7
Team - 4
Design Document
Sensors, controllers communication module

Aviral Sharma (2020201062)
Shashank Kumar (201502183)

April 2021

Contents

1 Overview	1
2 Block Diagram	1
3 Technologies Used	3
4 Environment used	3
5 Components/ Subsystems	3
5.1 Sensor and controller	3
5.2 Sensor Manager	3
5.3 Controller Manager	4
5.4 Sensor/Controller type registration	4
5.5 Sensor/Controller instance registration	4
5.6 Config files	4
6 Actors involved	4
6.1 Application configurer	4
7 Interaction between this and other component	6
7.1 Sensor/Controller manager and Application manager	6
7.2 Sensor/Controller manager and Node manager	6
7.3 Sensor/Controller manager and Heart-beat manager	6
8 Persistence	6
9 Test Cases	6
9.1 Case 1:	6
9.2 Case 2:	6
10 File Formats	7

1 Overview

This module deals with Sensor/Controller manager. Application configurer gives configuration files to platform and it verifies the file and code format after which a unique sensor/controller id is generated and sent to the manager.

The manager then creates a kafka topic for the sensor/controller and establishes a connection with sensor/controller. Then using APIs with sensor/controller manager, sensor and controllers are abstracted with the application developer.

2 Block Diagram

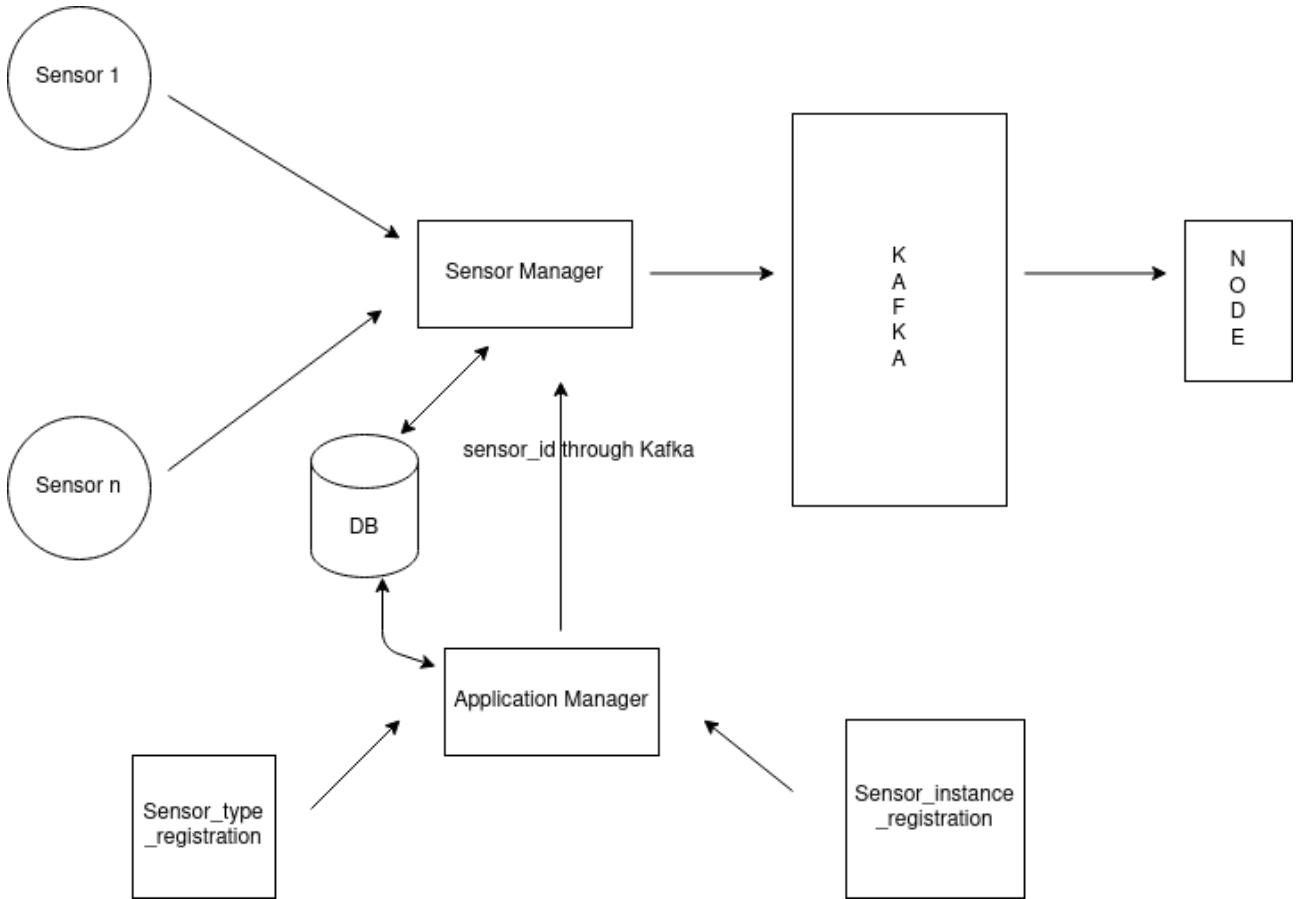


Figure 1: Block diagram of Sensor Subsystem

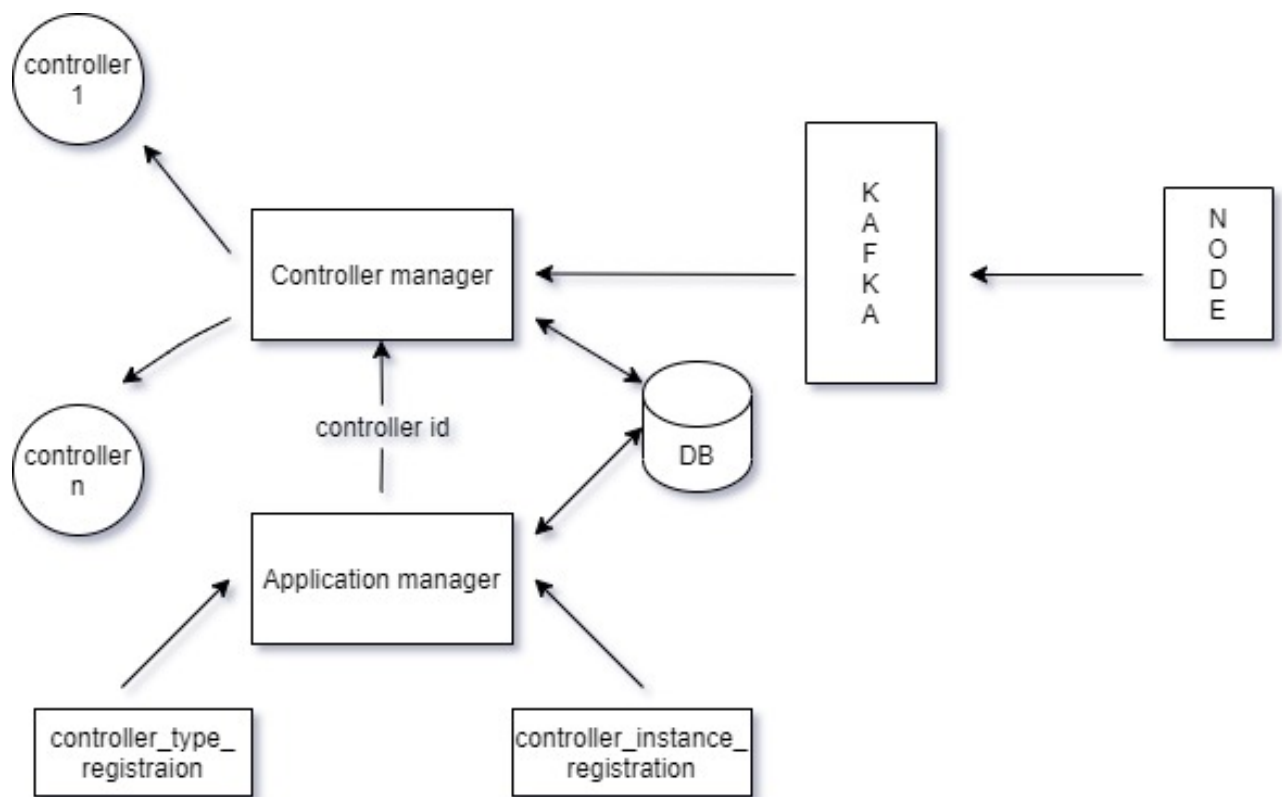


Figure 2: Block diagram of Controller Subsystem

3 Technologies Used

- Python Framework - overall development
 - Extensive Libraries support
 - Inbuilt data structures
 - Presence of Third Party Modules
- Apache Kafka - for communication
 - Producer-Consumer Model
 - Scalable
 - Fault Tolerant
- Azure
 - Scalability and Ductility
 - Interoperability
 - Cost effective
- Docker
 - Mobility - Ability to run anywhere
 - Rapid deployment
 - Isolation - It ensures your applications and resources are isolated and segregated.
- MySQL - for database
 - Data Security
 - High Performance
 - On demand scalability

4 Environment used

OS - Linux
Docker container

5 Components/ Subsystems

5.1 Sensor and controller

- Sensor/controller connect to sensor manager/controller manager respectively using unique ip and port.
- Each sensor/controller is uniquely identified using sensor id/ controller id respectively.

5.2 Sensor Manager

Sensor manager create topics in kafka for sensor instances and forward the data from sensors to respective partition of topics from where applicaion can consume it accordingly.

5.3 Controller Manager

Controller manager sends message/command to controller whenever application wants to send it.

5.4 Sensor/Controller type registration

For sensor/controller type registration, the zip file is uploaded to the repository and the json file is extracted and database table is created for it.

5.5 Sensor/Controller instance registration

- For sensor/controller instance registration, the zip file is uploaded to the repository and the json file is extracted and database table is created for it.
- For each sensor/controller instance a unique id is generated which is also sent to sensor/controller manager respectively using which Kafka topics can be created.

5.6 Config files

Config files are zipped inside sensor type registration/ sensor instance registration/ controller type registration/ controller instance configuration based upon the type of config file.

6 Actors involved

6.1 Application configurer

Application configurer will give sensor type config file, sensor instance type config file, controller type config file, controller instance type config file to the platform and it varifies the file and code format of the file. This configuration helps sensor/ controller to bind with application.

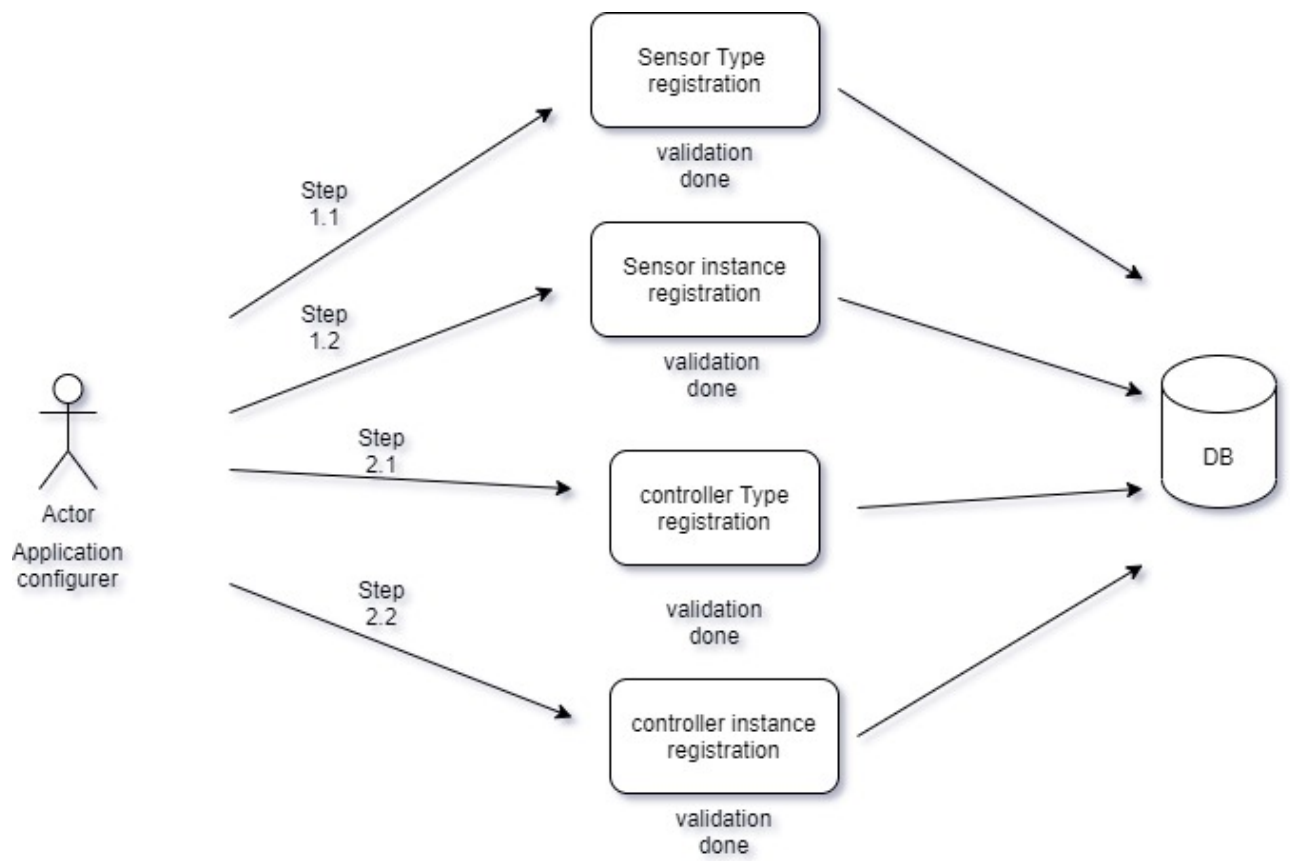


Figure 3: Application configurer

7 Interaction between this and other component

7.1 Sensor/Controller manager and Application manager

Sensor manager and Controller manager will get sensorid and controllerid respectively from the application manager which was generated upon registration of sensor and controller respectively on the platform.

The application manager produces the ids in kafka topic and sensor/controller manager consumes it.

7.2 Sensor/Controller manager and Node manager

An application instance running on a particular node may wish to communicate with the sensor manager and controller manager to send/ receive data stream to/from a particular/multiple sensors and controllers.

In case of sensor, Sensor manager produces data in kafka topic for node manager and in case of controller, controller manager consumes the message which was produce by node manager in kafka topic.

7.3 Sensor/Controller manager and Heart-beat manager

Sensor manager and controller manager will send continuous heartbeat message signal on regular interval of time.

8 Persistence

The name of the created topics are saved in a file. When the sensor/controller manager restarts, it tries to create topic with the names in the file. If it exists, it fetches the ip,port using the topic name from the database and restores the connection with sensor/controller.

9 Test Cases

9.1 Case 1:

Input : ip, port and unique sensor/controller id.

Output : Topic with name as unique sensor/controller id. Successful connection between sensors/controllers and sensor manager/ controller manager using given ip, port.

Discussion :The connection may be formed with different sensor at wrong ip, port.Thus, check the new connected ip, port values with the provided ones. If incorrect, terminate the connection and initiate a new bind request.

9.2 Case 2:

input : message/ signal from the application to controller manager. output : Display/notify message and change the controller values.

Discussion : The controller manager will send the value change command message to the specified controller. And the controller display message and changes its control values.

10 File Formats

- Sensor Type config file:

```
{
    "company" : "ABC",
    "model" : "X1",
    "data_rate" : "4 ms",
    "field_count" : 2,
    "field_1": "int",
    "field_2" : "string"
}
```

- Sensor Instance config file:

```
{
    "sensor_type": "sensor_type_1",
    "sensor_location" : {
        "ip" : "127.0.0.1",
        "port" : "5500"
    },
    "geo_location" : {
        "room_no" : "1",
        "house_no" : "123",
        "street" : "ABC",
        "city" : "Bhopal"
    }
}
```

- Controller Type config file:

```
{
    "company" : "QWE",
    "model" : "Y1",
    "input" : "string"
}
```

- Controller Instance config file:

```
{  
  "sensor_type": "controller_type_1",  
  "controller_location" : {  
    "ip" : "127.0.0.1",  
    "port" : "6500"  
  },  
  "geo_location" : {  
    "room_no" : "1",  
    "house_no" : "123",  
    "street" : "ABC",  
    "city" : "Bhopal"  
  }  
}
```