

INTERNALS OF APPLICATION SERVER

TEAM REPORT

Communication Module / Data Binding

Submitted By :

Pratik Tiwari

Tirth Pandit

Contents

1	Introduction	2
2	Communication Model	3
3	Sensor Registration	4
	3.1 Interaction with other Modules	4
	3.2 Flow Details	5
	3.3 Overall Design Flow	5
4	Sensor Manager	7
	4.1 Flow Details	7
	4.2 Interaction with other modules	8

1 Introduction

- Communication module is responsible to dynamically bind sensor data with running algorithm/instances, also will help sensor registration via minimal effort of application developer.
- Global Kafka server will be launch during initial setup and same will be used for all communication within other modules.

1) Sensor Registration

- Sensor registration is responsible for initial setup of sensors and bind the sensors to a gateway for dumping data
- Steps:
 1. Get config file from Request Manager
 2. Validate config file if not in correct format send error msg.
 3. Store the Sensor information in Run Time Registry
 4. Send Sensor a topic to dump data
 5. Store that topic in RunTime Registry

2) Sensor Manager

- This module is responsible for binding correct sensor data stream with corresponding algorithm. Sensor Manager, Add sensor are the services provided by communication module. Will identify devices from the sensor database based upon the parameters.
 1. Sensor Manager will listen on a topic for request
 2. Sensor will validate the request and will check if sensor is available in Registry or not
 3. A temporary topic will be created and will be send to Deployment manager
 4. Sensor Manager will release a Thread to serve request will read data from sensor topic, Process the data as per user requirement ,dump the data in temporary topic created by sensor manager

3) Processing Sensor Data

- This module is responsible for analyzing sensor data and converting the same in format specify by application developer.

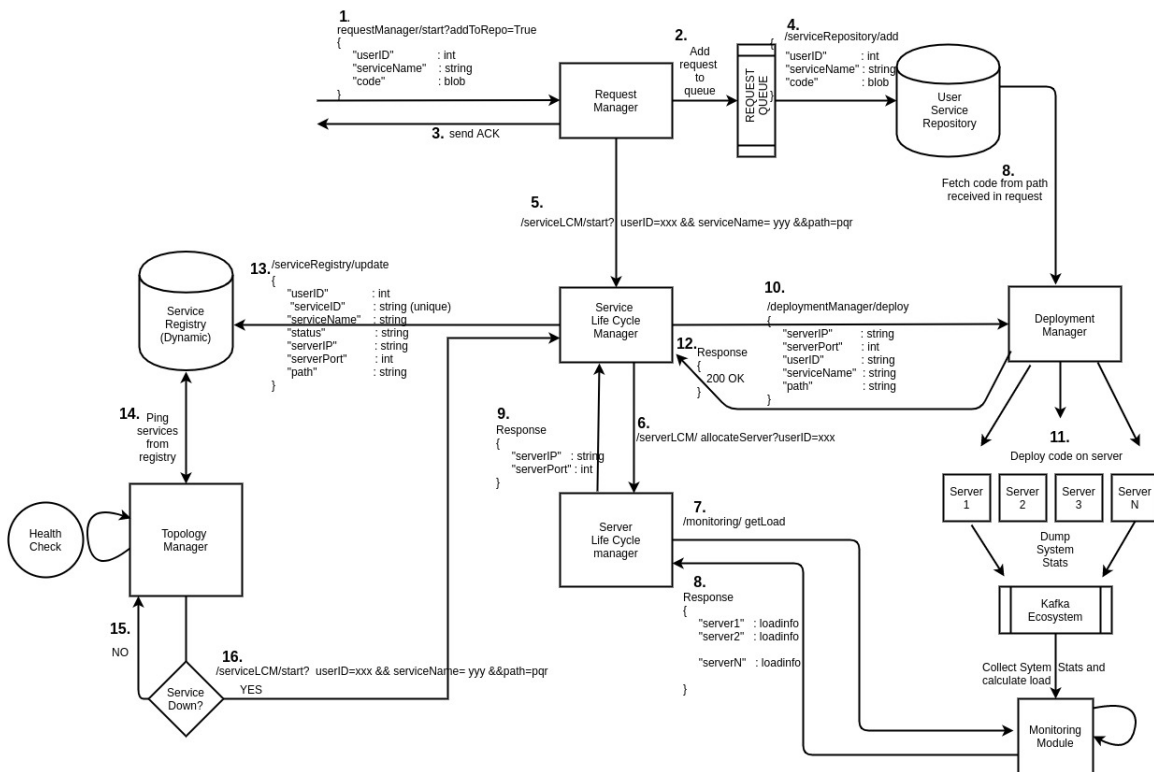
4)Control Sensor Data Rate

- This module is responsible for analyzing sensor data and controlling the flow of sending data to Nodes. Data rate will be controlled as specified by application developer.

5)Sending Sensor Data

- Sensor data will be read by a process launched by sensor manager. It will read data from topic and send the data in {sensor-id, data} format to required instance.

2 Communication Model



3 Sensor Registration

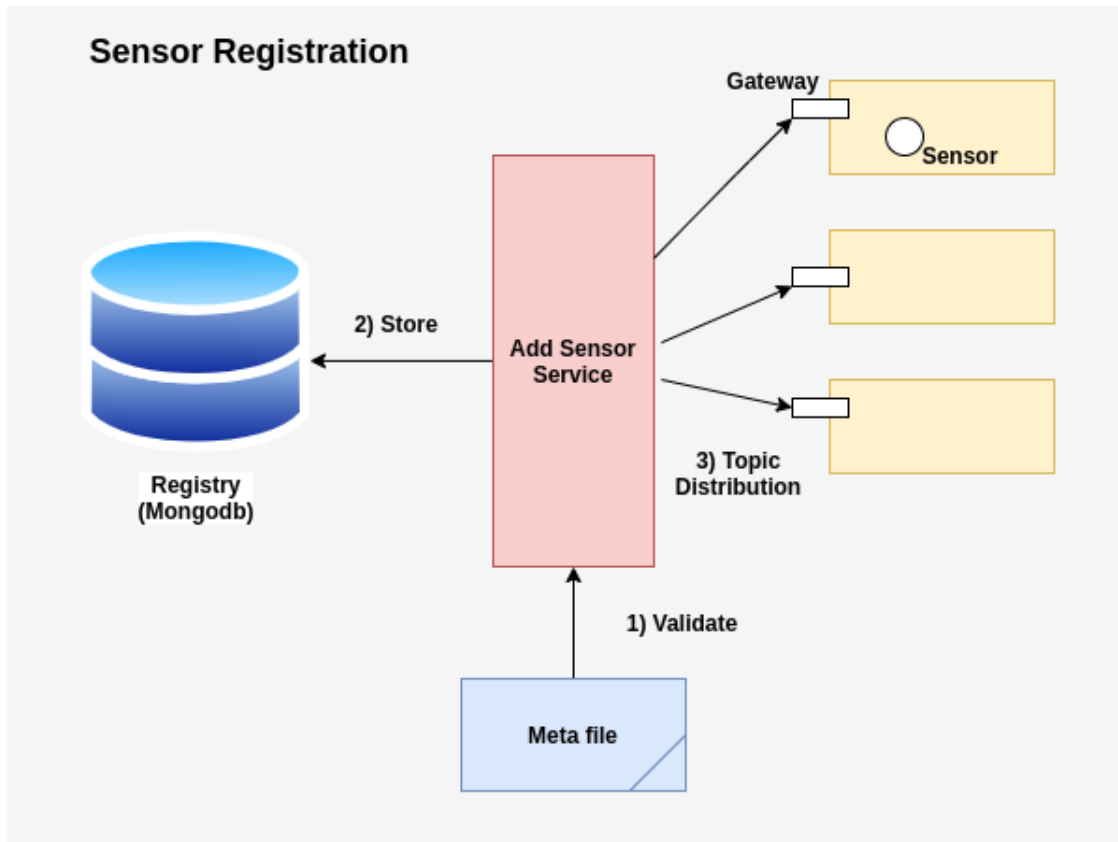
- Aim of this module is to create initial environment for the deployed application with respect to serving sensor data.
- User Interface will be provided to the admin for installing sensors to the platform. Admin can register sensors by providing meta/config file consisting information about sensors like ; Sensor Type ,Sensor Geo Location , Sensor output format ,Sensor Unique ID ;
- Sensor registration module will validate the config file and stores the appropriate information about sensor in platform repository.
- Sensor will also be responsible for creating unique data line(topic) .

```
Sensor-1 {  
  sensor_name: Fan  
  
  sensor_gateway : 'IP:Port'  
  
  sensor_data_type: output format Int/Float/Vector  
  
  sensor_geolocation: 'lat:long'  
  
  sensor_address: 'area:building:room_no'  
  
  data_dump: {  
    'kafka': 'broker_ip:topic'  
    'mongo_db': 'ip:port:passwd:document_name'  
  }  
  
  sensor_host: {  
    'socket': 'ip:port'  
    'kafka': 'kafka_broker_ip:kafka_topic'  
  }  
}
```

3.1 Interaction with other Modules

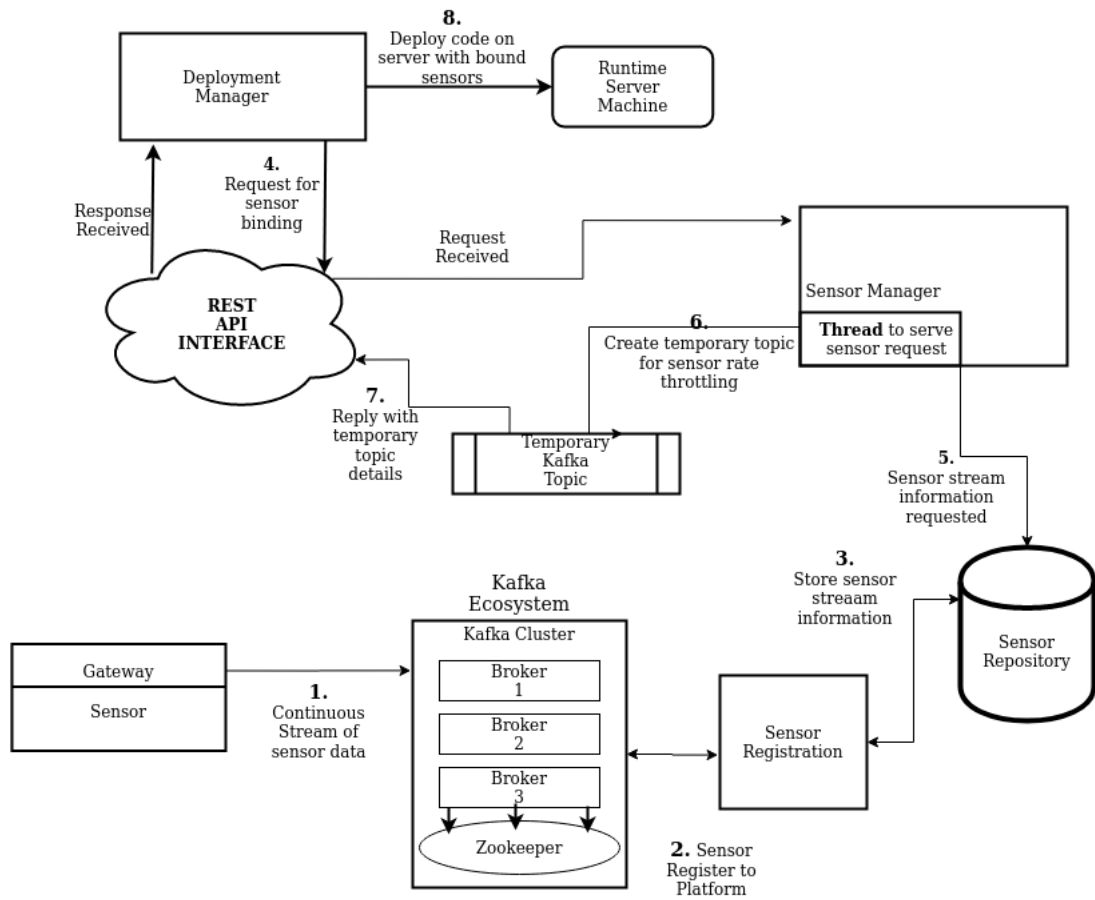
- Infrastructure admin can use user interface for sensor registration.
- User will be provided with UI for uploading config file
- Config file structure would consist of all important details of sensors.

3.2 Flow Details



3.3 Overall Design Flow

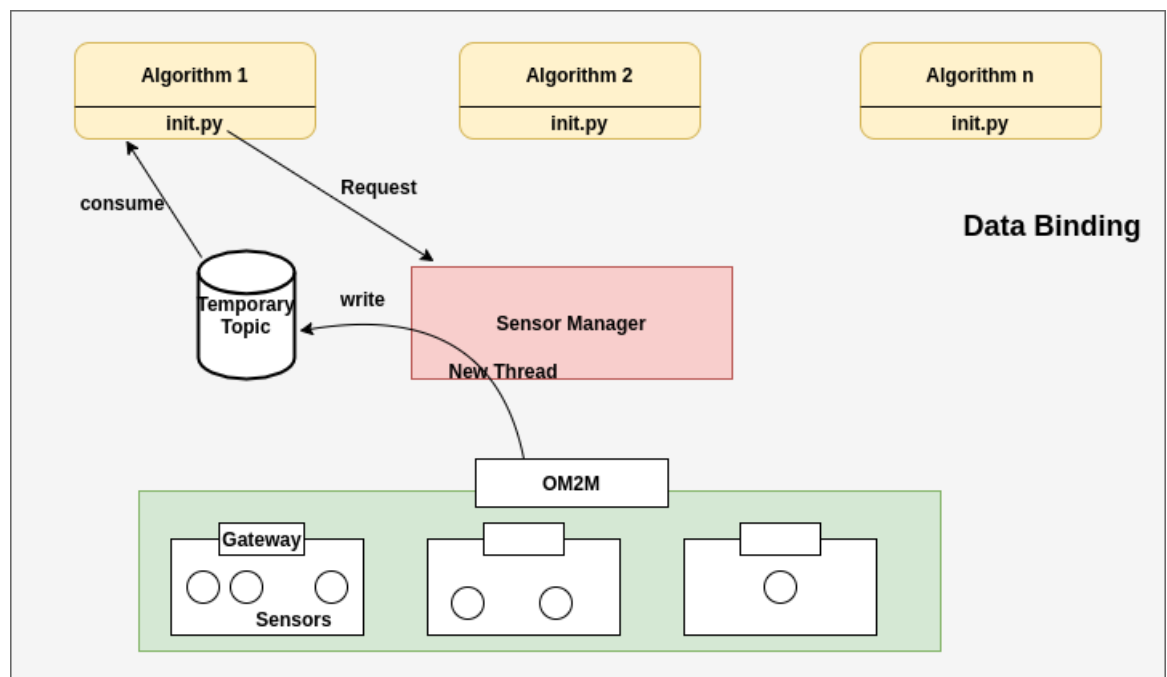
1. Each Sensor will have its unique property and its gateway point which will server data in or out to the sensor.
2. Admin will collect all information of sensors within it's infrastructure and will provide the same in form of JSON file to platform.
3. Registration sensor will validate the sensor config file and will perform appropriate action.



4 Sensor Manager

- Sensor manager module responsible for binding sensor data with deployed algorithm and sending the data as per the configuration provided by application developer
- sensor manager will overall take care of data binding of sensors data to algorithm , Preprocessing sensor data ,sending data at required rate
- Sensor manager will have access to all the details of sensors stored in platform registry,using this it will dynamically create temporarily process and Kafka topic for making the binding algorithm and sensor data.

4.1 Flow Details



4.2 Interaction with other modules

Sensor Manager \iff *platformRepository*

- on receiving sensor data request from running algorithm ,sensor manager will validate the request by checking existence of sensor from platform repository.
- After validation sensor will create temporary topic for algorithm and sensor communication.

Sensor Manager \iff *Node*

- Sensor manager will get request of sensor topic from init.py script running at some node.
- For each request sensor manager will detach a thread to serve the request.
- This thread will be responsible for reading sensor data from sensor topic ,Pre-processing the data as per config file , and write the data to temporary topic at given rate provide by sensor manager
- This temporary topic will read by node on which algorithm is running .

Overall Design Flow

1. Sensor Manager will listen to a common topic aware to all within the system for request.
2. Sensor Manager will serve request by detaching thread and performing required action on each request
3. Thread will read the data from sensor topic and will process the data in user required format. After processing it will even control the sending rate of data.
4. Config file shared by init.py will also specify at what time to server data and when to stop.

application_id

