IAS Requirement Document

AI-on-Edge App Platform (Services Component)

Darshan Kansagara 2018201033
Vatsal Soni 2018201005
Dhawal Jain 2018201065

Services Requirements Scope Document

1. Intro to the Subsystem

Definition:

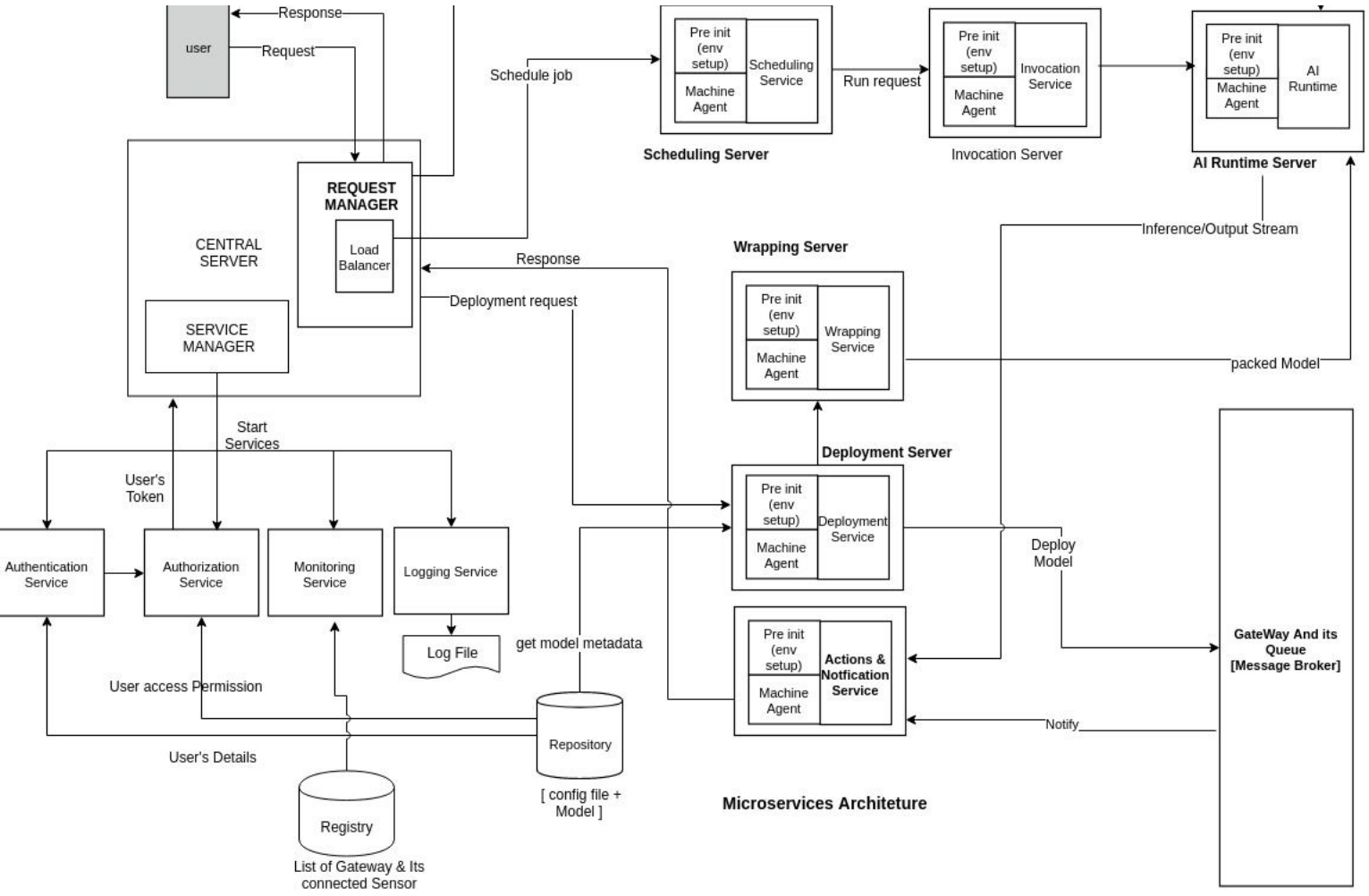
Service component is microservice based architecture of platform that coordinate with all other subsystem and different services. It includes user authentication, model packaging, deployment, and its relevant monitoring logging. It provide runtime for model deployment and invoke user's action and notification. It will also manages scheduling of model on gateway or server with different scheduling policy.

Scope:

Our platform provides a set of independent services that the app developer can use for app development with his own custom code updations. The platform provides various independent services like Security service (Authentication and Authorization), Build and Deployment capabilities , Logging and monitoring, Notification and Actions Service, Auto Scaling, Prediction and inference of AI model, Resource management, Scheduling service and repository to store data. All these services together handle all task from packaging model, scheduling, deployment and their monitoring.

2. Functional overview

2.1 Block Diagram



2.2 Brief description of each component

Our platform provides the following independent/autonomous services to the app developer who will be using our platform:

- **Authentication service :** Authentication service checks for the valid user using the authentication mechanism used in platform and send the authentication token back to the user for future use.
- **Authorization service :** Authorization service checks whether the authenticated user is authorised for a requested service or not.If the user is authorised then only the request is carried forward.
- **Scheduling service :** It will provide variety of scheduling. It will schedule model for particular time interval, between start and end time, repeat and many more .
- **Deployment Service :** It will package model and auto gen script to deploy model in any gateway or server instance.
- **Notification & Action Services** : It will notify user when user's response is ready. Also device sensor will notify server regarding specific event and server will take appropriate action.
- **Monitoring :** It will continuously monitored all service to check health of each service and create new instance of any failed/downed service immediate as action of fault tolerance.
- **Logging** : Provide an easy to use, general purpose logging system that keeps track of all events associated with a particular service. It all provides logs for the whole platform as well.
- **NFS :** It contains user's saved model, It will common interface between all services for storage purpose.

3. Use cases

List what the users can do with the service subsystem

1. User is authenticated by authentication module.
2. User will get notification via notification subsystem.
3. User will do inference using our subsystem
4. User will get monitoring health status in its dashboard about its model status.
5. User will schedule model deployment request to request manager which will passed to scheduler service.

Who are the types of users (name, domain/vertical, role, what they are trying to do when they need the system)

1. User1:
   Name : Normal User
   Domain/Vertical : Machine Learning Model Development
   Role : ML Developer
   Action : Develop and Deploy Machine Learning Models on edge/cloud server.

2. User2:
   Name : Admin User
   Domain/Vertical : Service Management
   Role : Service Admin
   Action : Adding/Removing of edge devices, Monitoring the subsystem and its modules ,also responsible for adding new services and thus enhancing the capabilities of server.

Atleast 5 usage scenarios.

It can play central role in all this application:-

1. Face matching by uploading an image and check if the person is visible in the camera feed and take appropriate actions.
2. Detect temperature & pressure anomalies in industrial IoT system and take actions based on the predictions.
3. Violation detection from a continuous video (like smoking / drinking offence)
4. Automatic vehicle number detection and send e-challan to the person who violates traffic rules
5. Crowd management - Keep track of head count of the people in large crowd gathering area like rallies so as to plan on deploying resources accordingly.

3.1 Primary test case for the project (that you will use to test)

Use-case Name: Security

- Domain/Company/Environment to use: Safety System
- Description:It will authenticate and authorised user before performing any task on platform.
- Information Model: Information of user that is stored in the database
- Sensory Data Used: Username and password
- Processing logic: It will verify user whether he/she is authorized or not and also handle access control of each user. This task is done by having information that is stored on database and take appropriate action.
- User's view : Will just see an alert message whether he has the required rights to access the functionalities of server or not.

Use-case Name: Logging and Monitoring

- Domain/Company/Environment to use: Platform Monitoring
- Description: It help to easily pinpoint the root cause of our application platform by monitoring every events of server.
- Information Model: Log file that contain all the information of all the events was performed on the server.
- Processing logic: It will access the log file and pinpoint the last event that was triggered and take appropriate action.
- User's view : Whole process will be abstract to user as it is done internally by the server for enhancing the performance and avoid replication of previously triggered independent event.

Use-case Name : Notification Services

- Domain/Company/Environment to use: Alert/Notification system
- Description:It will trigger notification to the user about status of any event that is requested by the user.
- Information Model:Response provided by the service manager which it will receive from the model or subsystem that is responsible for servicing the user requested task or if already processed that information will be accessed from the repository.
- Processing logic: It will process the response message and notify the service manager about the response status and its message and which transfer it to user who have requested the service
- Edge devices capture the photographs and give it to process it to trained Model. Once we get positive response(object gets detected), It notifies to the user.
- User's view :User Dashboard.

Use-case Name : Deployment Services

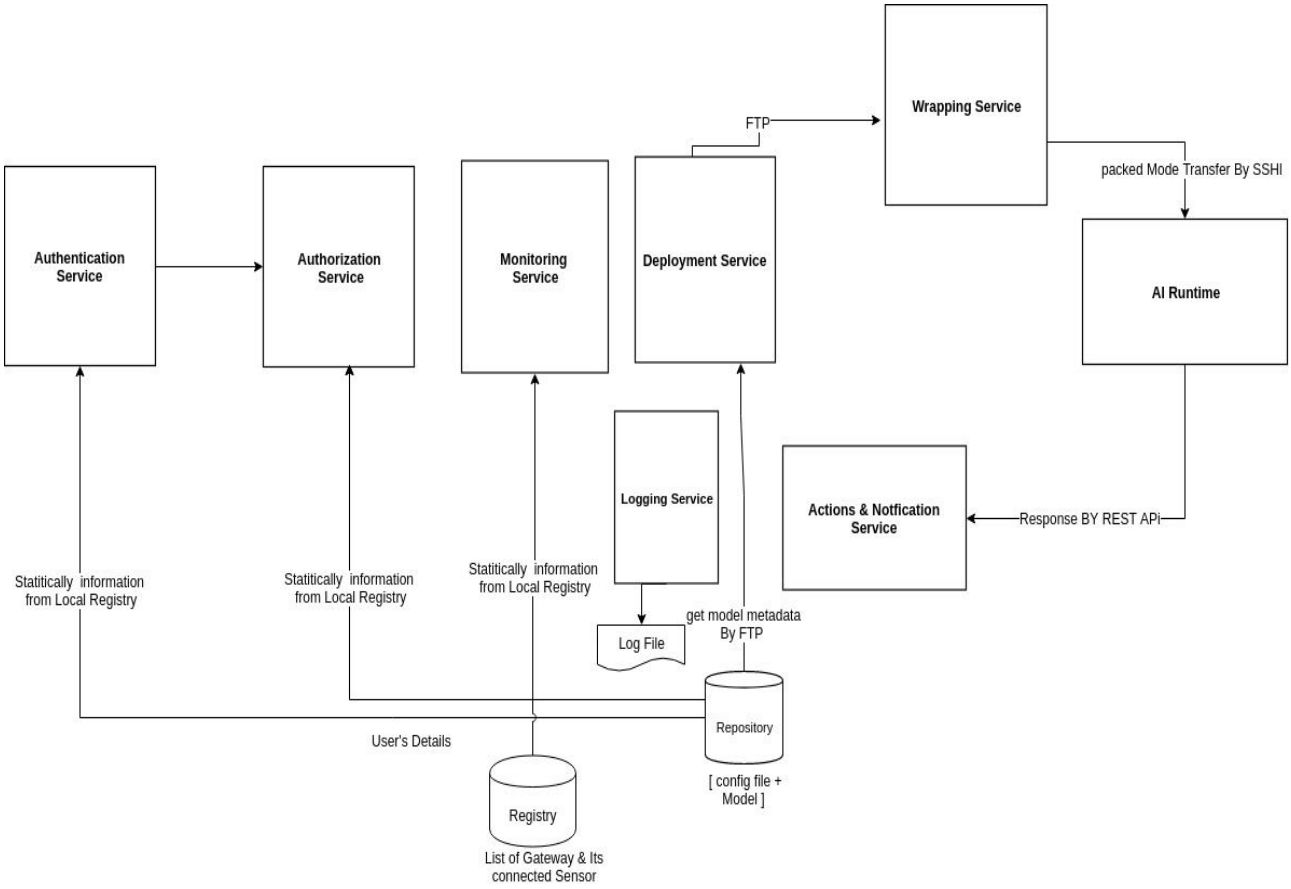
- Domain/Company/Environment to use: Packaging, Script Auto generation
- Description: It will package the model data and its related files and store it in NFS and generate script file to up/down the model.
- Information Model:Model data uploaded by user.
- Processing logic: It will receive the model from request manager and store the model in nfs. Also generate the script file and store in nfs and then call scheduling service to schedule the model to be deployed on Run machine.
- User's view :Deployment status on User’s Dashboard.

Use-case Name : Scheduling Services

- Domain/Company/Environment to use: Scheduling the model
- Description: It will provide the variety of scheduling type to schedule model on AI run machine.
- Information Model: Model data and scheduling parameters such as start time,end time, interval, repeat etc.
- Processing logic: It will receive details of model and scheduling info from deployment service and schedule it on AI run machine.
- User's view :Whole process will be abstract to user as it is done internally by the server.

4. Subsystems

4.1 Key submodules in the service-subsystem



4.2 Interactions involved across these subsystems

4.2.1 Protocols. Mechanisms.

- **HTTP** - Interaction between client and server
- **GRPC / REST** - Server and tensorflow serving
- **AMQP** : Server and Gateway communication channel
- **Web Services :** Some API calls between subsystem
- **NFS** : Network file system

4.3 External interfaces with the system

- **User Interface :**
  - **Web Application :** It may be flask web app for user to provide UI to user for inferencing task.

    - **Mobile Application :** It may be mobile app to interact with our platform through some API.
- **Server Component :** It will continuously interact with server manager for providing response of each task which subsequently goes to user. Our subsystem also interact with load balancer for picking up free Ip when load exceeds threshold on any services.

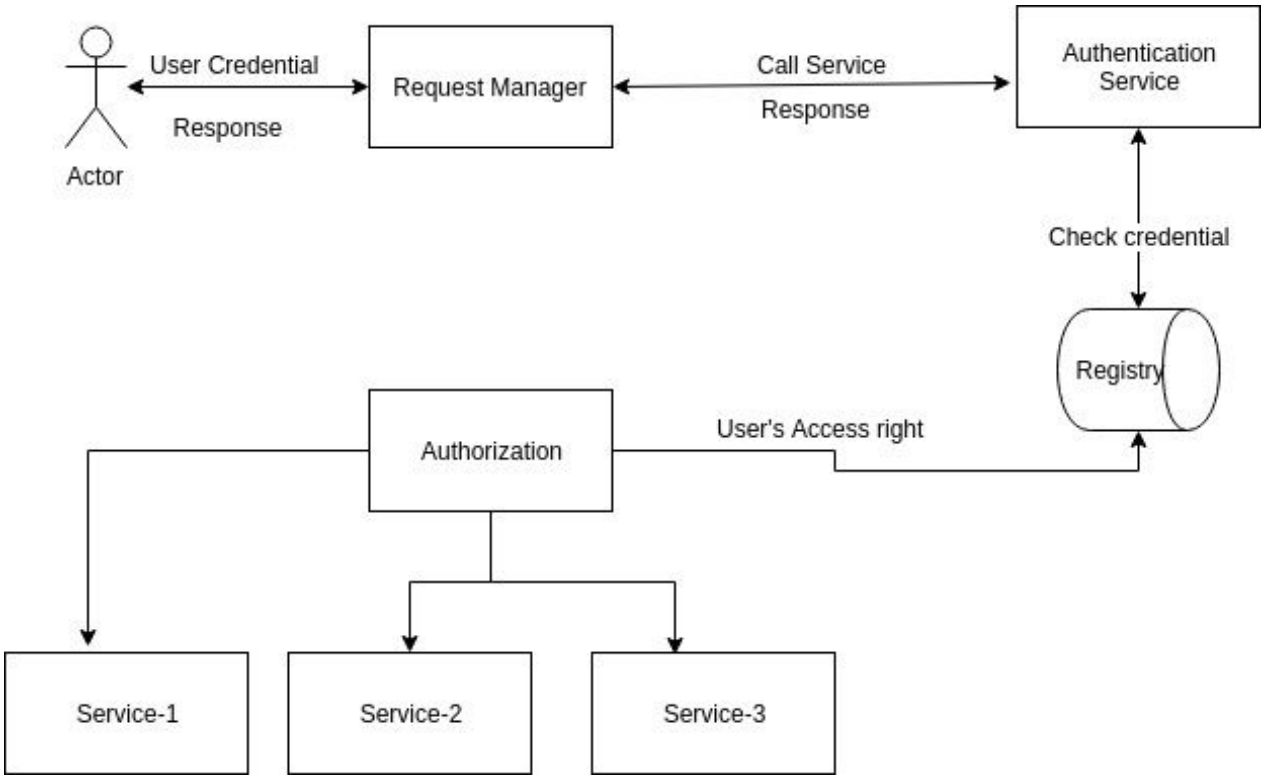
- **Message Broker**: Server interact with the gateway through message broker which further internally communicate with the edge device deployed on it and return the response through message broker.

4.4 Registry and repository

- Registry will contain the stats of every service (what instances are running on what machines, binary files location of each service in the common file system etc.)
- Repository will contain one directory per user which will in-turn contain one directory per service and will contain all the version-revisions of that service.

5. Brief overview of each of the sub module

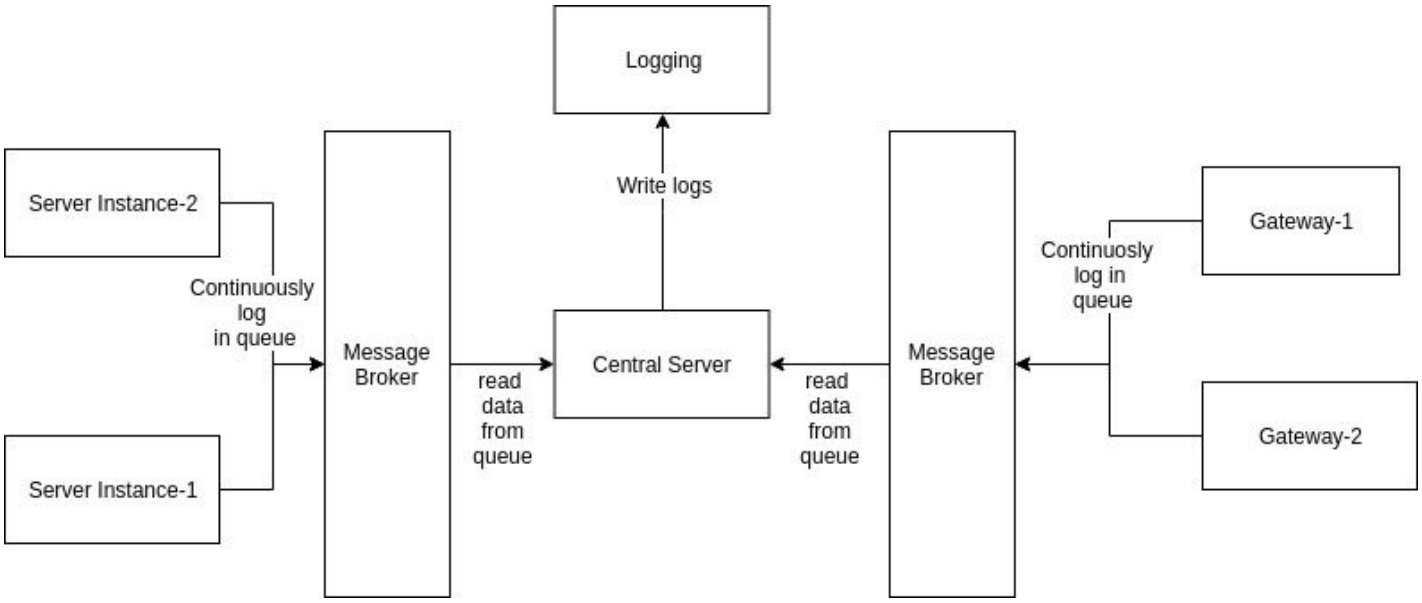
1. Security component (Authentication, Authorization and Encryption)



Main functionalities of this module are as follows:

- This service takes care of authentication of user and the service it is authorized to use.
- For maintaining data security and preventing system from external attacker it is used.
- It also maintains the integrity of data.
- Data is transferred in encrypted manner to prevent middleman attack.

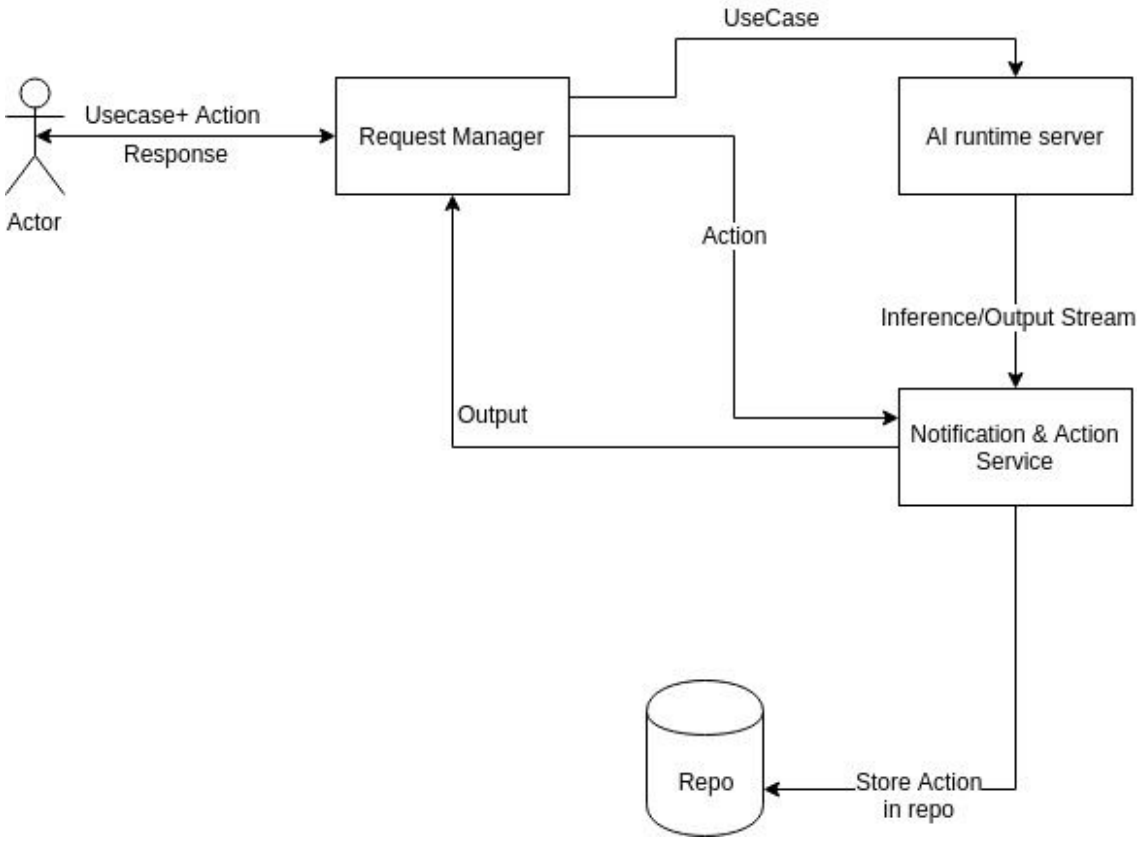
2.Logging (Service logging of every event)



Main functionalities of this module are as follows:

- Logging all the activities to manage and resources and monitor all the resource to keep them running and healthy.
- To find root cause of crash of our application platform by writing error and debugging logs.
- Search and download log.
- It will access the log file and pinpoint the last event that was triggered and take appropriate action.

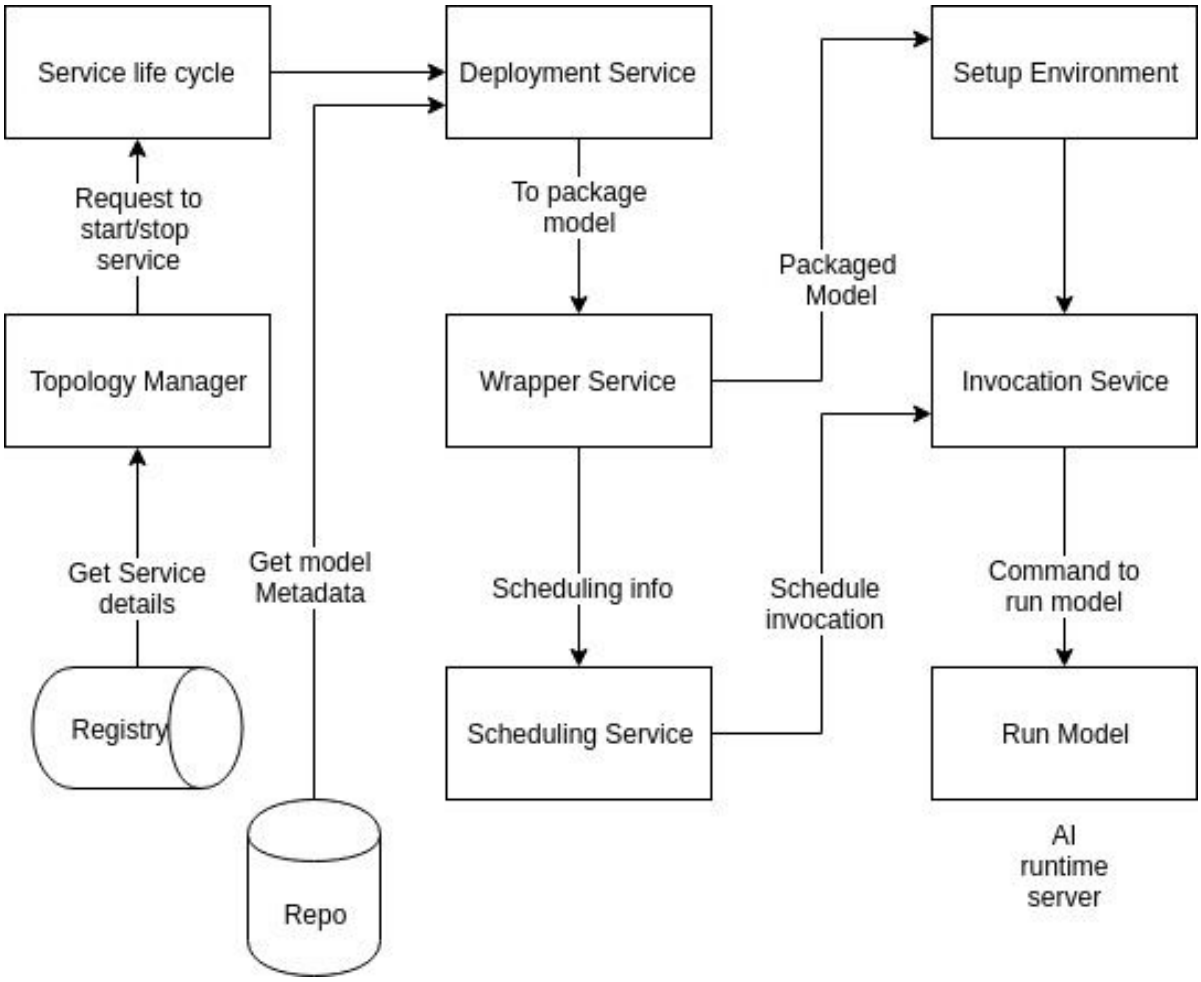
3.Notification Service



Main functionalities of this module are as follows:

- It notify user about specific event.
- Notification service send response along with status code to users.
- It gives alert or notify user when response of user's request is ready.

4. Deployment Service



Main functionalities of deployment services are as follows:

- It helps in deploying AI models in gateways and server instances.
- Inferencing can be done using deployment service only.
- Deployment service is responsible for scheduling of AI models.
- It is responsible for setting up runtime environment in server instances/gateways.